Study on Interoperability of Public Institution Chatbots, Interoperability with Consumer Assistants, and Question Answering Implementation

PART 1: INTEROPERABILITY OF PUBLIC INSTITUTION CHATBOTS

Completion tracker

Task 01 - Version history

Version	Main adjustments	Owner	Date
V0.1	First draft of 1.1 What is interoperability & interoperability standards	Deloitte	17/04/2024
V0.1	OP comments received for section 1.1	OP	24/04/2024
V0.2	First draft of 1.3 Current state of consumer assistant	Deloitte	24/04/2024
V0.3	First draft of 1.3 Key considerations for interoperability Partial addressing of OP comment on 1.1	Deloitte	08/05/2024
V0.4	First draft of 1.4 Viable approaches	Deloitte	15/05/2024
V0.5	First draft of 1.5 Regulatory outlook	Deloitte	22/05/2024
V0.6	First draft of 1.6 Implementation framework	Deloitte	07/06/2024
V0.7	Shortened version	Deloitte	21/06/2024
V0.8	Response to comments	Deloitte	26/07/2024
V0.9	Response to comments	Deloitte	09/08/2024
V0.10	Final comments	OP	12/09/2024
V0.11	Final comments addressed	Deloitte	27/09/2024
V0.12	Participants comments & testing phase insights	Deloitte	20/12/2024
V0.13	First draft 1.7.5 PoC subchapter included	Deloitte	17/01/2025
V0.14	All comments closed and PoC technical document added in Annex. Overall study enhanced by PoC findings.	Deloitte	10/03/2025

Contents

List of figures

List of tables

Executive Summary

Problem statement

The Publications Office (OP) has launched a project under the Digital Europe Program (DEP) with the goal of achieving interoperability between chatbots used by public institutions. The primary objective is to centralize information sharing, allowing citizens to access a wealth of information without needing to navigate multiple public institution webpages. This streamlined approach aims to enhance the user experience by providing a single point of access to comprehensive data. Currently, the concept of interoperability between chatbots is relatively unexplored, presenting a unique challenge and opportunity.

Key findings

The project has identified two potential approaches for interconnecting chatbots:

1. Primary Knowledge Directory (PKD): A global repository that catalogues available chatbots and acts as a centralized hub.
2. Interoperability Layer (IL): An intermediate layer that facilitates communication between the host bot (the chatbot initially receiving the query) and contributor bots (other chatbots that can provide additional information).

Each of these approaches can be implemented in the following ways:

- Centralized: This model relies on a single repository (PKD/IL) where all chatbots are stored. User queries are directly routed to this central PKD/IL, which identifies and engages the most appropriate chatbot to respond.
- Semi-centralized: In this approach, the host bot first attempts to answer the user's query. If it cannot provide a satisfactory answer, the query is then passed to the central PKD/IL to find the most suitable chatbot.
- Decentralized: Each chatbot maintains its own PKD/IL. User queries are automatically routed to the PKD/IL, which then follows a centralized approach to select the best chatbot for the response.

To assess the feasibility of these solutions, the project will implement a Proof of Concept (PoC). This PoC will apply one of the identified approaches to two chatbots to evaluate their performance and integration capabilities.

Recommendations

Creating interoperability between public institutions' chatbots will leverage existing chatbot functionalities to ensure wider coverage of information. This integration will streamline the information search process for citizens, allowing them to receive accurate and comprehensive responses from a single interface. Interoperability is crucial in making information retrieval more efficient and reliable. Citizens will benefit from a system that intelligently routes their queries to the most knowledgeable chatbot, minimizing the time and effort required to find relevant information. Public institutions will benefit from increased engagement and the ability to provide timely and precise information to their users.

Short conclusion

In conclusion, the interoperability project spearheaded by the Publications Office under the Digital Europe Program addresses a critical need for a unified information dissemination system among public institutions. By developing and implementing the Primary Knowledge Directory (PKD) or Interoperability Layer (IL) approaches, the project aims to simplify access to public information for citizens. The anticipated outcome is a seamless, efficient, and user-friendly system that enhances the public's ability to obtain necessary information quickly and effectively. The success of this initiative will mark a significant advancement in public information services, setting a new standard for chatbot interoperability and information accessibility. As well as a significant milestone for interoperability with potential of introducing a first preliminary standards that public institutions could follow to make future interoperability possible.

Abstract

This study examines multiple facets of chatbot interoperability within public institutions spanning various domains. First, it delineates the concept of interoperability, incorporating diverse programmes, initiatives, and standards pertinent to the topic. Next, it looks at the existing landscape of interoperability, highlighting study participants and detailing three case studies of interoperable projects within Europe. The discussion extends to the benefits, challenges, and risks inherent in establishing an interoperable network. Key considerations for interoperability are explored, such as the various types of chatbots, necessary setup requirements, and the influence of UX/UI principles. Critical prerequisites include dataset and conversational flow considerations, along with licensing and security aspects. Special attention is given to language considerations as the study focuses on the inclusion of low-resource languages in the interoperable network. Additionally, the study outlines viable approaches, including the centralized knowledge repository and interoperability layer, deployable in both centralized and decentralized formats. Important regulations, such as the AI Act and GDPR, impacting interoperability are also reviewed to advise on compliance. Lastly, a comprehensive implementation framework is presented, to serve as a guide for users in constructing an interoperable chatbot network.

Glossary of terms

Term	Description
A2B	Administration to Businesses
A2C	Administrations to Citizens
ABB	Architecture Building Blocks
AI	Artificial Intelligence
API	Application Programming Interface
ASR	Automatic Speech Recognition
BLEU	BiLingual Evaluation Understudy
BOSA	Belgian Digital Transformation Office
CCN	Content-Centric Networking
CDM	Common Data Model
CDNs	Content Delivery Networks
CEF	Connecting Europe Facility
Chatbot	Software application that leverages either pre-defined responses or artificial intelligence (AI) techniques to interact and respond to inquiries autonomously, eliminating the need for human intervention
CINT	Chatbot Interoperability Network Taxation
CINT-B	Chatbot Interoperability Network Taxation - Business
CINT-I	Chatbot Interoperability Network Taxation - IT
Contributing bot	Bot transferring content to host bot
CPOV	Core Public Organization Vocabulary
CPSV	Core Public Service Vocabulary
DEP	Digital Europe Program
DGA	Data Governance Act
DNN	Deep neural networks
DNS	Domain Name System
DoD	Definition of Done
DSI	Digital service infrastructures
EBP	European Blockchain Partnership
EBSI	European Blockchain Services Infrastructure
EIC	European Interoperability Cartography
EIF	European Interoperability Framework
EIRA	European Interoperability Architecture
EIS	European Interoperability Strategy
E-SENS	Electronic Simple European Networked Services
EU	European Union
FPS	Federal Public Service
GDPR	General Data Protection Regulation
GenAI	Generative AI
GPAI	General-purpose artificial intelligence
GPT	Generative Pre-trained Transformer
Host bot	Initial bot interface replying to user
HPC	High-performance computing
HTTPS	Hypertext transfer protocol secure

Term	Description
ICT	Information Communication Technology
IDA	Interchange of Data across Administrations
IDABC	Interoperable Delivery of European eGovernment Services to public Administrations, Businesses, and Citizens
IEEE	Institute for Electrical and Electronics Engineers
IL	Interoperability Layer
IMAPS	Interoperability Maturity Assessment of Public Services
IoT	Internet of Things
IP	Intellectual Property
IPRs	Intellectual Property Rights
IQAT	Interoperability Quick Assessment Tool
ISA	Interoperability Solutions for European Public Administrations
IT	Information Technology
KPIs	Key Performance Indicators
LLMs	Large Language Models
LRLs	Low Resource Languages
MeitY	Ministry of Electronics and Information Technology
ML	Machine Learning
MLP	MultiLingual Preprocessor
MT	Machine Translation
NA	Not Applicable / Not Available
NDA	Non-Disclosure Agreements
NeGD	National e-Governance Division
NIF	National Interoperability Frameworks
NLP	Natural Language Processing
NLU	Natural Language Understanding
NMT	Neural Machine Translation
OP	Publications Office of the EU
OVON	Open Voice Network
PKD	Primary Knowledge Directory
PoC	Proof of Concept
PRH	Finnish Patent and Registration Office
REST	Representational State Transfer
SLA	Service Level Agreement
SMT	Statistical Machine Translation
SOA	Service-oriented architecture
SOAP	Simple Object Access Protocol
STORK	Secure idenTity acrOss boRders linKed
TESTA	Trans European Services for Telematics between Administrations
UAT	User Acceptance Testing
UI	User Interface
UMANG	Unified Mobile Application for New-age Governance
UX	User Experience
VCS	Version Control System

Term	Description
VICA	Virtual Intelligent Chat Assistant

1 Interoperability of public institution chatbots

1.1 Introduction

In an era characterized by rapid technological advancement, the capacity for systems to work together seamlessly—termed interoperability—has emerged as a cornerstone of digital transformation. With the rise of generative AI, chatbots have jumped back at the forefront of user experience with generative AI assistants enhancing current intent-based chatbots. Interoperability to allow the exchange between different systems and a larger knowledge sharing is an essential capability for enabling chatbots, often developed by diverse entities, to communicate, exchange data, and use the exchanged information effectively. Interoperability holds particular importance for public institutions where the promise of streamlined communication, resource optimization, and enhanced public service delivery hinge on the robust exchange of information across various platforms.

This study delves into interoperability between public administration chatbots, including those operating across different knowledge domains. Public administration chatbots are virtual assistants designed to manage inquiries, disseminate information, and provide services to the public. These chatbots operate in various knowledge areas such as healthcare, education, transportation, and social services. As the drive towards digital governance intensifies, the necessity for these AI-driven tools to interact and share information seamlessly has become increasingly apparent. Effective interoperability between these chatbots would not only enhance service delivery but also ensure a unified citizen experience across different service domains which leads to increased user adoption.

The study is organized into several comprehensive sections to explore the subject in depth. First, "What is Interoperability" establishes a foundational understanding of interoperability, its types, and its significance in digital public services. Next, "Current State of Interoperability" analyses existing practices within public administration chatbots, including a review of existing chatbots working towards interoperability, as well as benefits, challenges, and risks associated with it. "Key Considerations for Interoperability" then addresses critical factors for success, encompassing technical, and organizational dimensions like key requirements for interoperability and UX/UI considerations. The section "Viable Approaches" explores strategic and technical solutions, offering insights from successful implementations. "Regulatory Outlook" examines the role of legislation and policy, providing an overview of current regulations and potential legislative changes. Finally, "Implementation Framework" presents a structured roadmap for public institutions to develop interoperable chatbot systems, detailing steps from planning and development to deployment and continuous improvement. By exploring these areas, the study aims to present a holistic view of interoperability in the context of public administration chatbots, offering valuable insights and actionable recommendations to foster a more integrated, efficient, and citizen-centric digital public service ecosystem.

The importance of chatbot interoperability extends beyond technical considerations; it embodies a shift towards a more collaborative and interconnected public sector. This collaboration can eliminate redundant information silos, thus promoting more accurate and timely responses to public inquiries. Moreover, interoperable chatbots have the potential to improve decision-making processes by aggregating data from multiple sources, thereby providing a more comprehensive view of various public administration functions. By establishing a standardized approach to chatbot interoperability, public institutions can ensure that they remain agile and responsive to evolving citizen needs. Furthermore, the study will highlight how successful chatbot interoperability can enhance citizen trust and satisfaction by providing consistent and coherent service delivery across different public sectors. Ultimately, this study seeks to underscore that achieving chatbot interoperability is not merely a technical goal but a critical component in the evolution towards a more efficient, transparent, and user-centric digital government. In the future, this progression can foster broader advancements, potentially paving the way for enhanced interoperability in various sectors.

1.2 What is interoperability

Interoperability is the ability of different systems, devices, or applications to connect and communicate in a coordinated way, without effort from the end user. The Institute for Electrical and Electronics Engineers (IEEE) further declared interoperability a crucial aspect for the connected future, especially for Internet of Things (IoT) systems (IEEE Standards Association, 2016).

According to the latest version of the European Interoperability Framework (European Commission, 2017), there are four levels of interoperability:

1. **Technical interoperability** – Ensures functional communication of data between systems, servers, and applications (e.g., hardware, software, networks, protocols, interfaces, and data formats).
2. **Semantic interoperability** – Ability of systems to exchange and comprehend data (e.g., metadata, data schemas).
3. **Organisational interoperability** – Seamless collaboration between organizations through alignment of business goals, governance, processes for information exchange and workflows.
4. **Legal interoperability** – Ensures organization operating under different legal frameworks can collaborate. Addresses differences in laws across borders through agreements or new legislation.

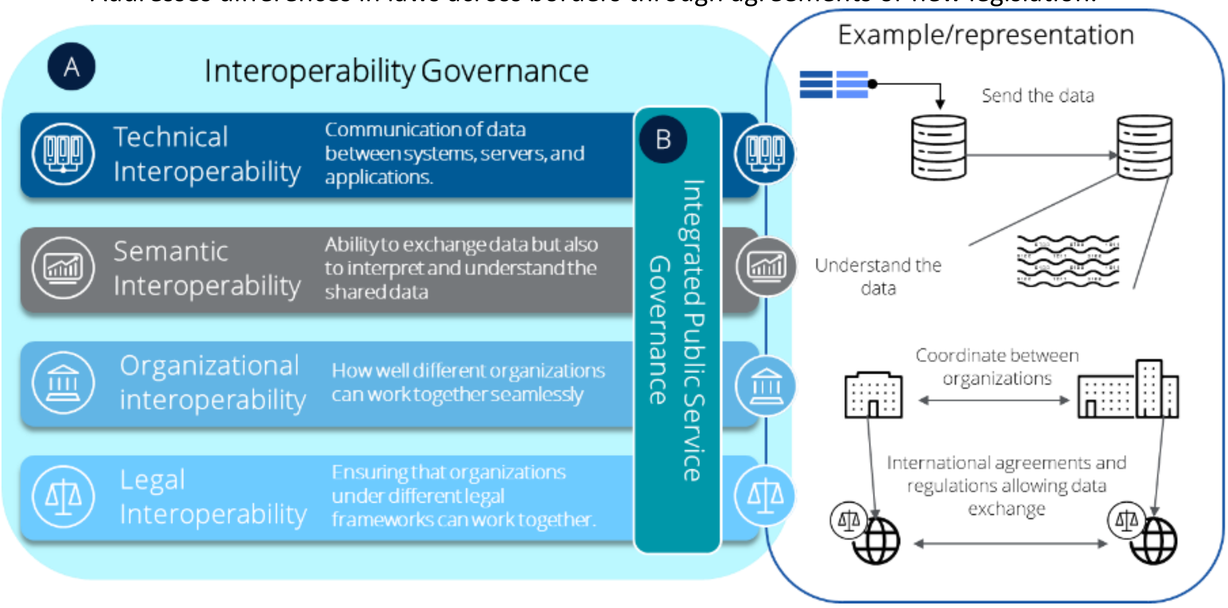


Figure 1. Levels of Interoperability

This plays a crucial role in facilitating effective interoperability. For chatbots, semantic and technical interoperability will influence efficiency of their interactions. The next section will focus on the EU programmes and initiatives and additional interoperability standards from there.

1.2.1 Interoperability standards

The European Union (EU) prioritizes interoperability for several reasons, one being the growing recognition that efficient public services, effective communication, and smooth functioning of the digital single market are heavily dependent on interoperable systems. This section will cover the current active EU interoperability programmes[1] and initiatives[2]. As an outcome from these programmes & initiatives, some overarching interoperability standards have been established, such as the EIF recommendations that frame the way forward.

[1] EU programmes are one of the main tools for implementing EU policies and can cover a wide variety of sectors (e.g., education, health, innovation). They usually span several years, are strategy-based and often come with specific funding.
[2] EU Initiatives are usually proposed by the European Commission and then approved by the European Parliament and the EU's Council of Ministers. They are usually in response to emerging societal, economic, or environmental and may aim to promote cooperation, dialogue and understanding.

1.2.1.1 EU Interoperability programmes

Past EU programmes (1995 -2021)

The EU's first programme, Interchange of Data across Administrations (IDA), began in 1995, focusing on administrative efficiency, a unified Europe, telecommunication networks, and data protection. Its successor, IDA II (1999-2005), expanded benefits to businesses and citizens and ensured safe data exchange. The Interoperable Delivery of European eGovernment Services (IDABC, 2005-2009) aimed at efficient, transparent eGovernment services, emphasizing administrative efficiency and cross-border services. Interoperability Solutions for European Public Administrations (ISA, 2010-2015) centred on cross-border interaction and interoperable solutions. ISA II (2016-2020) prioritized seamless interaction in a Digital Single Market and efficient administration via digital solutions. For more in-depth information regarding these EU programmes, refer to Appendix B.1.1.

Curent EU programmes: Digital Europe Programme (DEP) – (2021-2027)

The Digital Europe Programme aims to shape and support the digital transformation of Europe's society and economy. This initiative aims to invest & ensure accessibility across the economy and society by all public administrations, business, and citizens in key digital areas:

- High-performance computing (HPC),
- Artificial Intelligence (AI),
- Cybersecurity and trust,
- Advanced digital skills.

In conclusion we have seen that the previous EU programmes foster technological advancement and facilitate interoperability across administrations, organizations, and citizens. These programmes have evolved over time, to keep up with the pace of technology. The following section will showcase some of the initiatives and the link to chatbot interoperability.

1.2.1.2 EU interoperability initiatives

The first initiative related to interoperability was the Trans European Services for Telematics between Administrations (TESTA) which began in 1996 and was updated in 2002 in the context of IDA, IDABC, ISA, and ISA II. It facilitates secure and efficient communication between European public administrations, evolving over successive phases to adjust to technological advancements. The European Interoperability Strategy (EIS) was then introduced in 2010 in the context of ISA and ISA II to provide an overarching strategic context for EU public service actions, promoting interoperability. Core Public Service Vocabulary (CPSV) and Core Public Organization Vocabulary (CPOV) were both developed for ISA and ISA II respectively to facilitate semantic interoperability among EU public administrations. The European Interoperability Cartography (EIC) is a repository of interoperability solutions for EU public administrations, viewed as an ongoing initiative which was initially created in 2017 for ISA II. Finally, the European Interoperability Architecture (EIRA) was also created in 2017 in the context of ISA II. It is an essential blueprint for enhancing interoperability across all European public services, continuously contributing to European digital single market efforts. Additional information on these initiatives can be found in Appendix B.1.2B.1.2.

European Interoperability Framework (EIF) (2004, updated in 2010 & 2017)

One of the current EU initiatives which is most relevant for chatbots interoperability is the European Interoperability framework. It delivers a set of recommendations to augment the interoperability of digital public services all over the EU and offers specific suggestions to streamline public administrations at every level. It aims to enhance interoperability among public administrations, as well as between administrations and the public. The EIF is a broad initiative that impacts all layers of interoperability: technical, semantic, organizational, and legal (offering recommendations about standards, principles, and guidelines across these layers). For more in-depth information about the EIF, consult Appendix B.1.1B.1.2.

Figure 2. EIF sections[3]

Section 1 Underlying principles covers four categories: context for EU actions on interoperability, core interoperability principles, generic user needs and expectations and foundational principles for public administration cooperation. Section 2 introduces the interoperability model underpins all digital public services. It comprises four layers (legal, organizational, semantic, technical), an integrated public service governance, and interoperability governance for effective interoperability and seamless digital service delivery across Europe. Section 3 discusses the conceptual model designed to guide the planning, development, operation, and maintenance of integrated public services, applicable across all levels of government. This promotes 'interoperability by design' and consists of loosely coupled components interconnected via shared infrastructure.

1.2.1.3 Implications of EU programmes and initiatives

EU programmes have considerably advanced interoperability standards, particularly the 47 recommendations outlined in the European Interoperability Framework (EIF). This universal, adaptable framework guides public administrations in using shared IT solutions like APIs and software components. Though the field evolves rapidly, the EIF keeps updated to remain relevant.

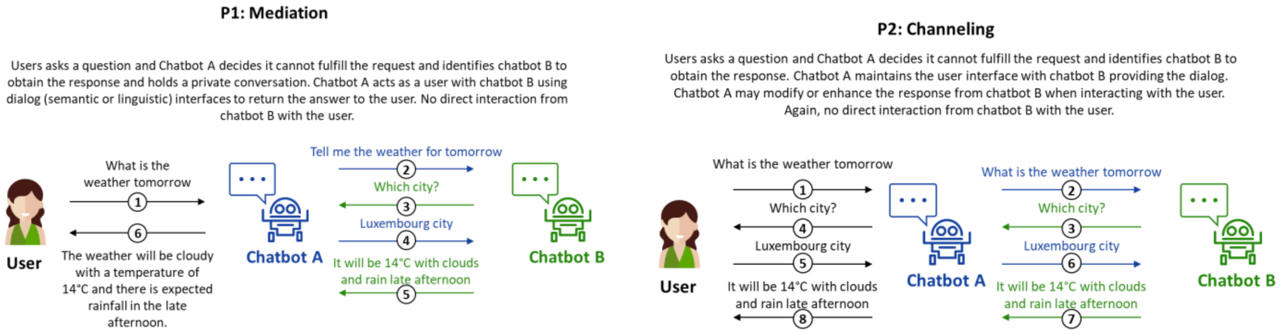
Integrating Generative AI (GenAI) into conversational AI systems necessitates industry-wide standards, focusing on conversational fluency, interoperability, and chatbots' continuous learning. As this domain advances, industry leaders should introduce standards to guide responsible, user-friendly chatbot development. Existing and future EU programs should also encompass these developments, with the current Digital Europe Programme (DEP) including AI as a key pillar. Legislative actions, such as the AI Act's inclusion of 'General-purpose AI models', prompt cross-organizational discussions, fostering best practices and market standards in chatbot interoperability.

1.2.1.4 Industry standards

The Open Voice Network (OVON) is a non-profit association dedicated to advancing the value, credibility, and usefulness of voice assistance in daily life. It is focused on two initiatives:

- The **Open Voice TrustMark Initiative** is promoting and creating ethical solutions to make conversational AI worthy of trust — promoting ethical principles across focus areas of privacy and security, media and entertainment, health and wellness, and consumer education.
- The **Open Voice Interoperability Initiative**[4] focuses on creating specifications and software for AI voice and chat assistants to collaborate using a standard messaging API. It involves developing an open, standard application programming interface (API) widely adoptable to enhance integration for voice and conversational AI. As part of this pursuit, an ecosystem of standardized, cross-communicating conversational assistants is envisioned, allowing users to switch seamlessly between different assistants and language models as needed. This is facilitated through the identification of three architectural patterns of interoperability conversational interaction:

[3] To see the full list of recommendations, please refer to the EIF at the following link: eif_brochure_final.pdf (europa.eu)
[4] Please find the link to the initiative: Interoperability Initiative - Open Voice Network and its github Open Voice Interoperability Initiative · GitHub



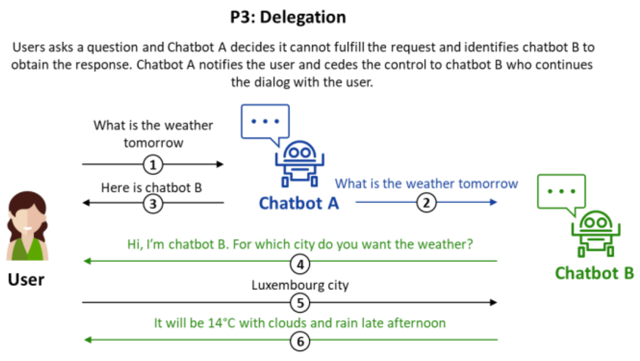


Figure 3. Architectural patterns outlined by the OVON

Appendix B.1.3 will cover additional information about this initiative including standard development process and future directions.

The difference between Mediation and Channelling lies in the interaction dynamics: For Mediation, Assistant A and Assistant B hold a private behind-the-scenes conversation, where Assistant A often already has most or all of the information needed from the user before reaching out to Assistant B for additional input, translating user requests and responses could be achieved. For Channelling, Assistant A acts more as an intermediary, continuously relaying and translating user requests and responses between the user and Assistant B, enabling ongoing dialogue and interaction.

Having expanded the notion of interoperability within various programmes and initiatives and the relating standards from the EIF recommendations, it is now important to discuss its relevance in the current context.

1.3 Current state of interoperability

Figure 4. Involved Member States participants

1.3.1 Overview of study participants and interoperable chatbots

We surveyed European public institutions about their chatbot implementations and interoperability efforts, by inviting EU institutions and public institutions of member states with chatbots to participate in the study, share their experiences to contribute to the advancement of chatbot interoperability. The **Error! Reference source not found.** showcases the spread of participants in the study. A more detailed overview of the participants can be found in Table 1 below. There is a rise of virtual assistants in public institutions, indicating a progressive shift towards more efficient and responsive services. It is important to note that the absence of certain countries in this study does not imply that they do not possess chatbots; rather, it indicates that these countries did not participate in the study.

The chatbots utilized by public institutions play a significant role in promoting efficient and effective service delivery and communication to the citizens. This section will provide an overview of chatbot interoperability projects ongoing in EU member states, for more detailed information on the use cases refer to A.B.1.1.1. For information on interoperability projects outside of Europe, see Appendix A.B.1.1.2.

Furthermore, we examine the four participant use cases which are interoperable to inspire potential approaches for connecting different chatbots. We supplemented the case studies through collaborations and workshops with multiple institutions. Starting up smoothly ran live from 2019 to 2024, providing available public information, whereas Bürokratt and CINT are under development. Starting up Smoothly is a collaboration between three public institutions to pilot a common chatbot service. The goal is to provide a common chatbot service to entrepreneurs who would like to start their business in Finland. Bürokratt is an interoperable network of chatbots on the websites of public authorities that allows people to obtain information from these authorities through a chat window. The goal is to interconnect public authorities' chatbots to provide comprehensive information to users. Chatbot Interoperability Network Taxation (CINT) aims to interconnect tax chatbots to provide accurate tax obligations to users across Europe in their native language. The Orchestrator Project is an internal interoperability initiative connecting the chatbots of two Belgian public institutions via a master bot.

Table 1. Overview of study participants

ID	Country	Institutions	Anonymized	Topic	Languages	Technology	Status	Interoperable
1	Austria, Belgium, Croatia, Greece, Latvia, Lithuania, North Macedonia, Portugal, Poland, Romania, Slovakia, Sweden	Pilot project for EU Fiscalis	Tax institutions across Europe	Tax	Official languages of each participating country	API, centralized database, translation service	Interoperable PoC coming soon	Yes
2	Belgium	Belgian Digital Transformation Office (BOSA)	Digital institution		Dutch, French	NLP intent-based chatbots (Presteria and IBM Watson), LLMs in the future	PoC in development	No
3	Estonia	Joint Agency of enterprise Estonia and KredEx	Business and Innovation organization	Living, working, and doing business in Estonia	English, Estonian, Russian	LLM & human-defined intents	Production	Yes
4	Finland	Finnish Immigration Service (Migri), Finnish Patent and Registration Office (PRH), Development and Administration centre for ELY Centres and TE	Three administrative institutions	Starting a business, immigration, looking for a job	English, Finnish	Intent based, Boost.ai	Pilot until 2024, now discontinued	Yes

ID	Country	Institutions	Anonymized	Topic	Languages	Technology	Status	Interoperable
		Offices (KEHA Centre)						
5	Luxembourg	Service central de Législation Luxembourg	Legislative institution	Law	French	Mistral	PoC	No
6	Luxembourg	List Luxembourg	Research organization	Framework to automatically develop chatbots			Development	No
7	Portugal	Agency of administration modernization	Administrative institution	Immigration	16 including English, French, Spanish, Italian, Russian, Ukrainian, Southeast Asian languages such as Hindi, Bengali, Urdu	ChatGPT 3.5	Development	No
8	Portugal	Independent (Cabinet of justice)	Legislative institution	Justice	Understands Portuguese but only responds in English	ChatGPT 3.5	Production	
9	Estonia	Bürokratt		Estonian Police and Border Guard Control, Statistics, municipalities, environmental board, National Library,	Estonian		Production	No

ID	Country	Institutions	Anonymized	Topic	Languages	Technology	Status	Interoperable
				Consumer Protection and Technical Regulatory Authority, Tax and Customs Board				

1.3.1.1 Starting up Smoothly

Project description

The project piloted a triad of chatbots to assist foreign entrepreneurs in Finland regarding starting a company (PRH), taxation (Vero), and immigration (Migri) (Miessner et al., 2019). Users interact with a single interface, and the AI determines which chatbot best answers their query based on the topic. Each organization's chatbot can direct users to other relevant organizations. The chatbot serves as automated customer service, responding to user inquiries.

Target audience

The project was specifically designed to assist **foreign entrepreneurs** in initiating their businesses Finland.

Technology

All three chatbots have the same technology (boost.ai environment) to make it simpler to implement interoperability.

Timeline

The development timeline of the project lasted 12 months, from March 2018 to March 2019, and was divided in two phases.

Phase 1: Chatbot network demo (March 2018 – June 2018)

Phase 2: Chatbot network pilot (June 2018 – March 2019)

Interoperability setup and flow

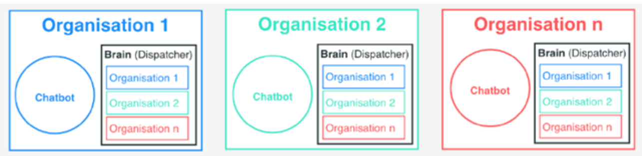
Chatbot included in the interoperable network

Institution	**Finnish Immigration Service (Migri)**	**Finnish Tax Administration (Vero)**	**Finnish Patent and Registration Office (PRH)**
Chatbot name	Kamu	VeroBot	PatRek
Languages	Finnish & English	Finnish & English	Finnish & English
Description	Answers migration related questions and was made to be friendly and answer questions to the point	Answers tax related questions and was made to be friendly and answer questions to the point	PatRek chatbot is able to answer questions related to setting up a company
Feature	Can use emojis selectively to make discussions more humane (never more than one in a row, positive emojis and country flags)	It can use emojis selectively to make discussions more humane (maximum one in a row)	NA

Three institutions and chatbots were part of the project.
Vero later on left the project and was replaced by Aino (owned by Development and Administration centre for ELY Centres and TE Offices (also known as KEHA Centre)).

How the interoperable network is managed

The project used the concept of a brain as a dispatcher to facilitate interoperability. The dispatcher knows all other chatbots in the network and forwards the question it received to all of them.



The chatbot with the highest probability threshold answers the question of the user on the same interface as where the conversation started.

Testing

Several rounds of user testing were conducted to gain valuable insights into the perception and usage of the three chatbots by immigrants. This testing aimed to understand how users interacted with the services in real-life scenarios to improve both the content and technical capabilities of the chatbots.

Results

During the period from November 2018 to June 2019, a significant amount of user engagement was observed with the three chatbots. Some numbers:

- Over **53,000** conversations with the chatbots, which averaged around 240 daily chats.
- More than **3,100** transfers made through the chatbot service

Interoperable network after the pilot

The original network kept working after the pilot, until VeroBot left the network in 2021. In 2022, Aino chatbot (owned by Development and Administration centre for ELY Centres and TE Offices (KEHA Centre)) joined the network. At the time, the target audience became wider: it contained foreigners willing to work in Finland, too. Interoperability was found very useful tool, but requirement to use the same technology didn't make it simple to keep in use. In 2024, the project met its final end.

1.3.1.2 Bürokratt

Project description

The goal is to create an Estonian state virtual assistant. Bürokratt is an interoperable network of chatbots on the website of public authorities that allows people to obtain information from these authorities through a chat window. It provides individuals, or users, with the opportunity to access direct public and information services using virtual assistants. The chatbot's source code is open source.

Target audience

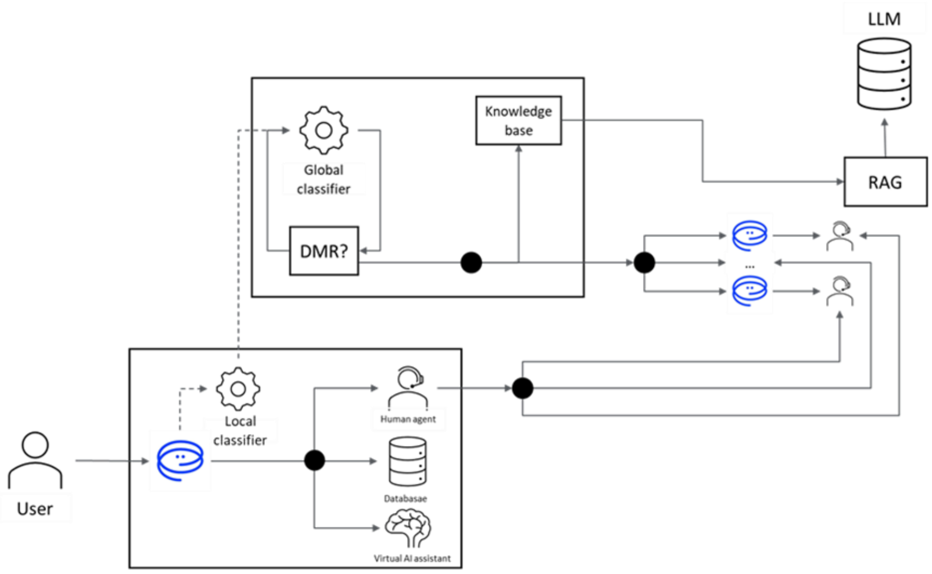
Aims to provide **Estonian citizens** the ability to easily access public services using virtual assistants. It is currently live in 10 services:

- Estonian Police and border Guard Board
- Statistics Estonia
- Viimsi municipality
- Integration Foundation
- National Library or Estonia
- Environmental board
- Register of Economic activities
- Consumer Protection and Technical Regulatory Authority
- State information system authority
- Estonian Tax and Customs Board

Timeline

2020: Vision and concept paper drafted, and the first pilot project conducted.
2021: Development focused on speech-enabled projects, machine translation, speech recognition, and 80 public sector case studies.
2022: Chat function implemented, enabling citizens to access select public services.
2024: Chatbot currently deployed in 10 different services across Estonian public institutions.

Interoperability setup and flow



Testing

Currently not available

Results

- By end of 2020: 41 AI solutions deployed in government and private sector.
- By end of 2021: Alpha version of Bürokratt implemented in three agencies.
- 2022: Progress towards Bürokratt's beta version, with AI processing citizen messages.
- 2024: Currently deployed in 10 different services across Estonia.

1.3.1.3 Chatbot Interoperability Network Taxation (CINT)

Project description

The Chatbot Interoperability Network on Taxation (CINT) is an EU collaborative initiative. The objective is to provide comprehensive information and support in native language for EU individuals regarding tax obligations. To reach this objective, CINT has the following goals:

1. Instruct national tax administrations to ensure chatbots address topics relevant to citizens from various countries.
2. Establish technical linkage among tax administrations' chatbots via a centralized referral service/API.
3. Implement a central API to provide content in the required language using automated translation.

Target audience

The project seeks to provide assistance and support to **EU taxpayers**.

Technology

Key components:

- Centralized hub
- Centralized API
- Automated translation (EU eTranslation)

Timeline

Key milestones:

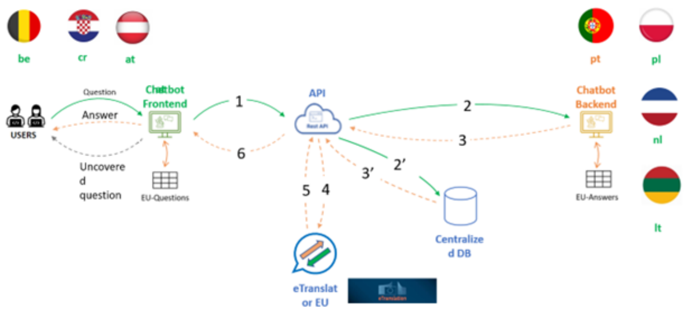
- 2023: Kick-off of the project
- 2024: A PoC is expected to be completed in Q4 2024
- 2025: The project is anticipated to conclude in May 2025

Interoperability setup and flow

Chatbot included in the interoperable network

Twelve countries are currently involved for the development of PoC: Austria, Croatia, Lithuania, Latvia (with chatbots), Romania, Portugal, Poland, North Macedonia, Slovakia, Belgium, Sweden (with databases) and Greece (both with chatbot and database).

How the interoperable network is managed



Two group perspectives

1. CINT-B: Focusing on information for chatbots (e.g., Q&A needed by taxpayers). Organize and catalogue data for universal use, ensuring systematization.
2. CINT-I: Focusing on technical challenges of interconnecting the chatbots (e.g., language, platform differences and mitigating risks related to these challenges).

The solution considered is to have a centralized API requiring minimal changes to internal chatbot systems.

Testing

Currently not available

Results

At its current phase, the project remains in its early stage with a Proof of Concept (PoC) that is nearing completion. With the completion of the PoC, the project will make significant progress, and it will lay the foundation for future development and implementation involving numerous tax interoperability chatbots across Europe.

1.3.1.4 Orchestrator implementation

Project description

The Orchestrator implementation aimed to introduce a master bot that links two chatbots for responding to Belgian users on particular topics via their website. This pilot seeked to test the feasibility of interoperability between the chatbots.

Target audience

The project's primary purpose was to assist **Belgian citizens** with online administrative tasks (e.g., eID and conducting federal job searches).

Timeline

Currently not available

Technology

The two chatbots are intent based and use the same technology (IBM Watson) to simplify the interoperability implementation.

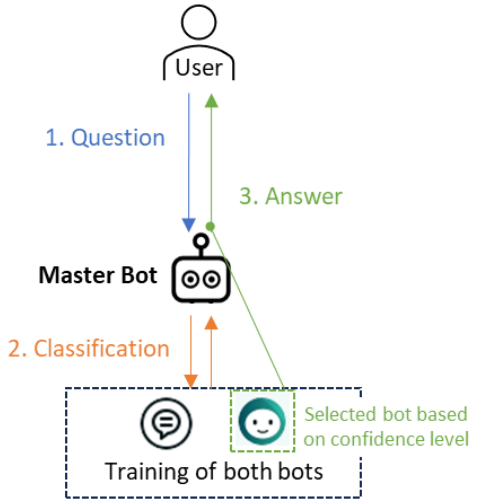
Interoperability setup and flow

Chatbot included in the interoperable network

Two chatbots were part of the project.

Service Desk	Helping people to log in with their eID
Bea	Helping people with federal job searches question

How the interoperable network is managed



The master bot was trained on both chatbots' datasets and, upon receiving a user query, performs a classification between the two domains based on an NLP analysis and confidence level. The possible outcomes were either the Service Desk bot, Bea bot, out of scope, or disambiguation. If Bea Bot had the highest confidence level, an additional classification was carried out using only the training data of the chosen bot to provide an answer to the user.

Testing

Two challenges emerged:
1. Disambiguation when training data overlaps or similar questions arise— needing to determine which bot to use.
Potential mitigation: Allow the user to choose the chatbot.
2. Scenario interruption when a question in the middle of a scenario is directed to another bot.
Potential mitigation: Ask the user if they want to switch the conversation, then switch, and, after responding to the question, return to the previous conversation.

Results

The master chatbot was developed three years ago but has not been deployed on the website due to governance aspects. The future objective is to enhance the chatbots using Large Language Models (LLMs).

1.3.2 Benefits, Challenges & Risks

Formulating effective interoperability demands understanding the current status of your chatbots and services. The following section details the potential benefits, challenges, and risks associated with chatbot interoperability.

1.3.2.1 Benefits

Benefits are defined as the advantages or positive outcomes that results from interoperability. By identifying and maximizing benefits, individuals, organizations, or societies can create value, improve performance, and achieve desired objectives. Some key benefits to organizations implementing chatbot interoperability:

Table 2. Overview of potential benefits of interoperability between chatbots

Benefit	Description
Improved service delivery	Interoperability facilitates seamless collaboration between chatbots of public services, improving efficiency and accessibility for citizens and businesses.
Larger knowledge base	Interoperable chatbots broaden organizations' reach by pooling information from various sources, offering an expansive knowledge base for both employees and customers.
Better Insights and Analytics	Enables platform providers to gather insights and analytics from multiple chatbot interactions, supporting data-driven decision-making and platform enhancements
Collaborative network	Interoperable chatbots promote a collaborative network among organizations with similar challenges, facilitating shared insights and strategies for collective advancement.
Increased User Adoption	Interoperability encourages more businesses to adopt the platform, knowing that their chatbots can work together with other systems, boosting each platform's user base
Cost-effectiveness	Having interoperable chatbots can reduce costs as public institutions can share technologies and leverage shared services.

Interoperable chatbots provide benefits like improved collaboration, efficiency and access to vast knowledge bases. Embracing these tools is critical for organizations to deliver seamless, valuable services in our interconnected world.

1.3.2.2 Challenges

Challenges are defined as obstacles or difficulties that requires effort or skill to overcome to achieve interoperability. They often arise when encountering constraints or uncertainties and can be overcome by strategic planning, innovation or problem-solving skills. Some of the potential challenges for organisation:

Table 3. Overview of potential challenges of chatbot interoperability and possible mitigation actions

Phase	Challenges	Mitigation actions
Planning	**Infrastructural investment –** Requires significant technical resources and infrastructural investment to establish	• Form collaborations to share expertise and infrastructure • Use cloud-based platforms and open-source frameworks
	Governance Models – Complexity in coordinating chatbot interoperability initiatives among different organizations	• Establish governance models and agreements to address data sharing, liability or dispute resolution • Facilitate communication through regular meetings, calls, workshops and having dedicated platforms/channels
	Compatibility and information asymmetry – Incompatibilities and development challenges in diverse systems can cause incorrect responses	• Encourage user feedback • Understand and investigate technical infrastructure

Phase	Challenges	Mitigation actions
	due to varying platforms, protocols, and data formats (Elkhodr, Shahrestani, & Cheung, 2016).	• Choose chatbots with flexible or similar architecture for easier integration • Establish common protocols for seamless communication
	Lack of centralized list of existing chatbots – Leads to missed opportunities for interoperability. Currently only static studies exist which mainly list existing interoperable initiatives[5] [6].	• Encourage the development of a generalized dynamic database
	Maturity difference – Some institutions will have a fully developed chatbot while others are still working on theirs.	• Think of different ways of collaboration, i.e. maybe chatbot cannot interoperate yet access to database can be shared
Development	**Language barriers –** Dialogue translation of chatbots, especially low-resource languages and the lack of publicly available multilingual chat corpus (Gain, et al., 2022).	• Invest in AI translation technology or onboard translation company • Involve native experts in translation, particularly for culture-sensitive or subject-specific matters
	Resource intensive – Building interoperability is time and energy consuming. The processing of data and network are resource intensive too, leading to potential budget issues.	• Determine optimal resource allocation • Align on budget within interoperable network
	Question Redirection – A development challenge is effectively managing question redirection and corresponding answer display.	• Design clear decision-making algorithms for redirection rules of different questions • Interfaces should be transparent if user is redirected
	Conversation Flow and Handoff – Managing flow, context retention and user experience consistently during cross-bot interactions has technical and design challenges.	• Define clear triggers for bot handoff • Keep users informed about the handoff process • Carefully design chatbot dialogues for coherent conversations, even when changing bots
	Consent & session management – Obtaining user consent for data sharing across multiple chatbots, while managing session lifecycles for personalized cross-bot interactions.	• Continually test and monitor session management • Perform user testing to validate session management in chatbot interactions and gather improvement feedback. • Implement consent management mechanisms to ensure user control over information sharing
	Error Handling and Recovery – Handling errors during cross-bot interactions is challenging due to diverse chatbot error strategies and recovery capabilities.	• Implement a set of standardized error codes across all chatbots. This will ensure uniform error handling across different chatbots. • Align on recovery strategy

[5] AI watch, artificial intelligence in public services
[6] Mapping innovation in the EU public services : a collective effort in exploring the applications of artificial intelligence and blockchain in the public sector

Phase	Challenges	Mitigation actions
	Multilingualism – Lack of publicly available multilingual chat corpus (Gain, et al., 2022). This presents a challenge for chatbot interoperability in dialogue translation of chatbots with different languages, especially low-resource languages.	• Use of EC eTranslation tool can overcome language limitations (will become open source in the future) • Consider native experts to be involved in translation
Testing & Maintenance	**Lack of data standards –** Standard protocols are necessary to ensure data is up to date and maintained well across the interoperable network	• Align on unified data standards, protocols & centralized repositories within interoperable network
	Security and privacy – Chatbots might handle sensitive user data. Ensuring secure data sharing and regulation compliance can be complex when multiple chatbots interoperate.	• Make sure to use secure communication protocols, robust authentication and authorizations mechanisms, and anonymise user data if present

Interoperable chatbots present both benefits and challenges, including integration complexities, data sharing issues, and information asymmetry. Addressing these challenges necessitates collaboration and standardization, thereby unlocking the potential for improved user satisfaction and organizational efficiency.

1.3.2.3 Risks

Interoperability may pose risks, highlighting potential negative outcomes linked with uncertainty. Effective risk management includes identifying, analysing, assessing, and mitigating these risks to safeguard assets, reputation, and stakeholders. Currently, no universal framework exists for measuring interoperability (Gerontas, 2020). The risks identified below are drawn from industry insights and existing research on interoperability.

Table 4. Potential risks that can be encountered with chatbot interoperability and possible mitigation actions

Phase	Risks	Mitigation actions
Planning	**Inadequate interoperability standards** – Absence of universally accepted interoperability standards and guidelines creates complexity in uniform data handling and communication among chatbots.	• Follow existing standards such as the EIF where available • Establish industry standards and guidelines for chatbot interoperability
	Regulatory – Compliance is necessary, e.g., take into consideration legislative framework at European, national level. Aspects such as transparency also need to be taken into account.	• Conduct legal assessments to understand all relevant laws and regulations • Schedule regular audits and compliance checks
Development	**Misalignment of Responses –** Uncoordinated chatbots can produce contradictory responses. Lack of context may lead to repetitive or incomplete information, affecting the user experience.	• Define clear protocols and standards for consistent chatbot responses • Use context transfer mechanisms to keep user context during switches between chatbots
	Page related information – If one chatbot includes information that requires to be on the specific website of the chatbot it can lead to confusion for users. Users might believe they are interacting with the chatbot from the website mentioned, causing mistrust in the interoperability network.	• Provide users with clear disclaimers at the beginning of their interaction, informing them that they may receive responses from multiple chatbots in the network and advising them to verify the sources of information
	User experience inconsistencies – Poor interoperability may cause inconsistent branding	• Establish banding & design guidelines

Phase	Risks	Mitigation actions
	and user experiences across platforms, leading to trust issues and limited chatbot capabilities.	• Clearly communicate the limitations to users and provide alternative options or fallback mechanisms to overcome platform-specific restrictions.
Testing & Maintaining	**Data breaches –** Cross-system data exchanges may compromise sensitive data security due to weak integration, leading to potential unauthorized access or malicious activities (Hasal, et al., 2021).	• Utilize strong encryption methods and implement access controls and secure transfer protocols • Conduct security audits to minimize vulnerabilities
	Quality differences – Disparities in commitment and interest levels among different participating institutions.	• Align upfront on maturity levels for different bots when setting up interoperable network (interoperability mechanism should accommodate different maturity levels) • Establish contracts such as SLA to outline quality expectations and responsibilities
	Governance – Think about how interoperability will be maintained in the future.	• Form a governance body comprising all participating institutions • Clearly define RACI matrix with roles and responsibilities listed per party (setup and post-live maintenance)
	Responsibility for corrections – In case of mistakes in the reply given by a chatbot, the institution owning the chatbot is the only one able to update the chatbot to ensure accurate communication within the interoperability network.	• Develop clear protocols for inter-institutional collaboration, detailing how errors are communicated and resolved within the network
	Dependence on vendors / service providers – Chatbot interoperability could increase vendor dependence, limiting the ability to switch vendors without putting network at risk.	• Contractual provisions can be used to limit/ prevent strong independence on vendors • Make use of interoperable API which is agnostic of technology where possible

The benefits, challenges, and risks of interoperable chatbots reflect the technology's complexity. A common aim is enhancing user experience and service delivery. Achieving this involves overcoming security, compatibility, and standardization challenges and managing risks. In turn, organizations can leverage chatbot benefits, deliver superior user experiences, and pioneer technological innovation. Now, considering these aspects, we will move to the considerations of the prerequisites that make it feasible. In the next section, we will delve into the key considerations to make interoperability possible looking at chatbot types, requirements, multilingualism, UX/UI, and security aspects to consider for ensuring effective interoperability.

1.4 Key considerations for interoperability

1.4.1 Types of chatbots

Chatbots, with their varying forms and capabilities, serve several uses and can be selected based on the specific case at hand. The subsequent section details the most prevalent chatbot categories. Their respective advantages and drawbacks can be found in Appendix B.3.1.

A. **Rule-based Chatbots:** These simple chatbots function on predefined rules to answer queries using heuristics to generate the best response (Gupta & Hathwar, 2020; Akkineni, Lakshmi, & Sarada, 2021). They can be multilingual but are not optimised for ambiguous interactions (Adamopoulou & Moussiades, 2020; Ramesh, Ravishankaran, Joshi, & Chandrasekaran, 2017; Jia, 2009).
   Context-Aware Chatbots: They store past interactions for more relevant responses. They understand complex queries and follow-up questions (Gupta & Hathwar, 2020).
B. **AI/ML powered Chatbots:** Advanced by using Machine Learning (ML) and Artificial Intelligence (AI) for accurate experiences, learning from past conversations (Suta, Lan, WU, Mongkolnam, & Chan, 2020).
   Generative Chatbots (NLP): These chatbots leverage Large Language Models (LLM's) in this category predict the next word in a sentence for human-like responses from scratch (Kim, Chua, Rickard, & Lorenzo, 2023; Scotti, Sbatella, & Tedesco, 2023).
   Voice Enabled Chatbots: This is another sub-type of chatbots separate from the above-mentioned options as it can be added to any other type of chatbot as a functionality. They utilize voice recognition technology for voice-based responses. Examples include Siri, Alexa, and Google Assistant (Avandegraund, 2024).

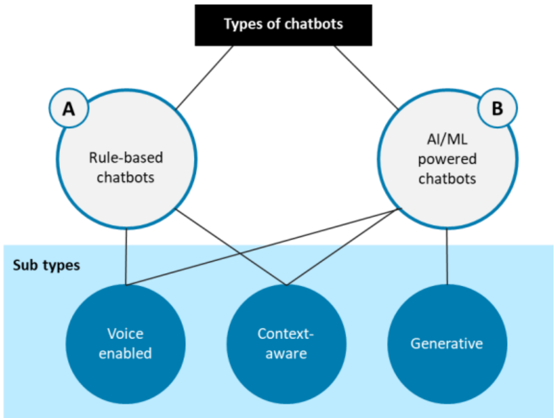


Figure 5. Types of chatbots

In summary, chatbots come in various types with diverse functionalities, from rule-based bots with restricted responses to AI-chatbots simulating human interaction. Each offers advantages and drawbacks, allowing organizations to select chatbots that best cater to their needs (see Table 22. Chabot types : Pros vs. Cons to have a comparison overview between both types). Further details on how different chatbots can interconnect will follow in Section 1.5.

1.4.2 Basic interoperability requirements

To achieve integrated communication and collaboration among chatbot systems, certain basic requirements should be met. This section explores the key aspects necessary to create the foundation of chatbot interoperability.

1.4.2.1 Dataset considerations

The dataset, key for chatbot responses, requires considerations like data sharing, format, and quality to deploy successful interoperable chatbots.

- Integration Compatibility: Assess if chatbots can integrate with other technologies or platforms, including API and software compatibility. Establish data-sharing agreements among chatbot-developing organizations to facilitate interoperability (Valencia, et al., 2023). Ensure chatbots' model adaptability for learning from new data inputs.
- Data Quality & Variety: Use accurate, relevant, and unbiased training data to equip the chatbot for a wide range of queries. This data should cover various topics, languages & dialects. Regular data quality checks and updates maintain the relevance and accuracy over time (Babu & Boddu, 2024). Use data augmentation techniques to enhance data scarcity and improve chatbot models.

Dataset considerations, such as data sharing agreements, adhering to data quality standards, and diverse training data, are essential for effective and efficient interoperable chatbots. Correct dataset considerations ensure accurate responses and seamless integration with other chatbot platforms. Upfront alignment and standards for the interoperable network needs to beset and all participating organisations should comply.

1.4.2.2 Conversational flow considerations

When designing interoperable chatbots, it is essential to consider the various conversational functionalities that enable seamless and effective interactions. Ferman (2018) emphasizes the importance of conversation flow within chatbots and outlines best practices for chatbots:

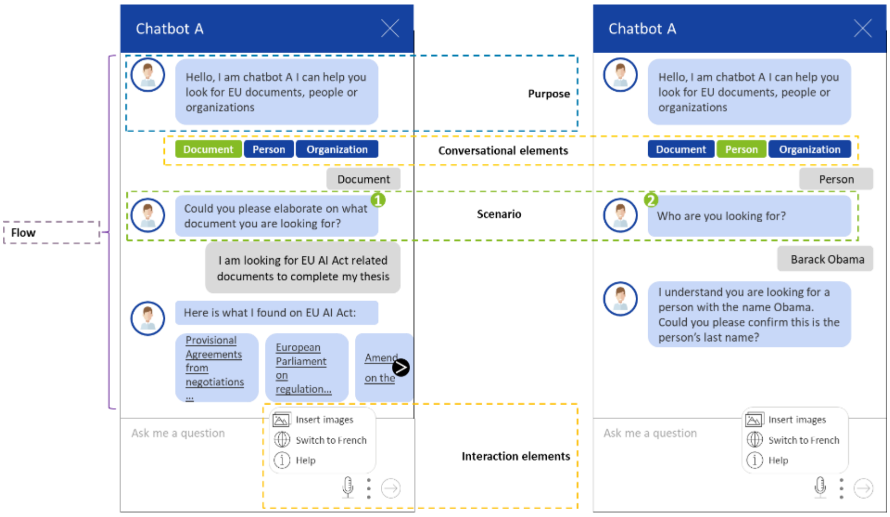


Figure 6. Chatbot conversational flow & best practice elements

<u>Chatbot purpose</u>: Defining a clear purpose promotes better communication design, scenario handling, and interaction elements. This definition aids in shaping the chatbot's conversational flow and elements.

<u>Conversation flow:</u> Craft a seamless conversational flow that lets users comprehend topic switches. This strategy is particularly vital for interoperable chatbots.

<u>Conversational scenario</u>: Creating conversational scripts outlines chatbot-supported actions and situations. This proactive preparation controls user expectations and clearly establishes the chatbot's boundaries.

<u>Conversational elements:</u> Choose elements that support the chatbot's purpose and improve conversational flow. These elements might be conversational (e.g., assertions or questions) or interactive (e.g., text integration, links, images, emojis, buttons).

The conversational direction must be considered together with the chatbot's transparency about its limitations. A robust knowledge base can guide user conversations effectively as the more knowledge a chatbot has, the better it can direct conversations (Ferman, 2018). Interoperable chatbots' conversation flow can be either one-directional or multi-directional. In a *one-directional flow*, conversations move from the host to the contributing chatbot, which only provides answers without questioning the host. Useful scenarios for one-directional flow include:

- <u>Pilot testing:</u> When evaluating interoperability performance, beginning with one-directional flow can be effective. Once successfully tested, consider implementing a multi-directional flow.
- <u>Data privacy</u>: For sensitive information, such as medical or judicial data, one-directional flow can safeguard privacy.
- <u>Unique entry-Point:</u> A one-directional flow allows a widely known chatbot to utilize smaller, specialized bot data for users, maintaining a consistent user interface.

Contrary to the one-directional flow, *multi-directional flow* allows a flexible, two-way data and conversation exchange. This approach offers a comprehensive information exchange that promotes more in-depth conversation, improving user experience as well as a cooperative experience where every bot in the network exchanges information, thus contributing to a robust knowledge base akin to a network of expert advisors. A multi-directional flow network not only enhances the usage of all the bots but also directs users to other services they might find intriguing, increasing overall network awareness and usage.

Voice considerations in the context of interoperable chatbots is important, even if text-based communication is the more prevalent method (ChatBot, 2023 (Updated in 2024)). This is primarily due to the need for precise information. Voice options, however, are prevalent in smart home devices or smartphones as they simplify communication and add a conversational dimension. These voice-interactive systems simplify communication giving them a more conversational aspect. Nonetheless, this doesn't particularly translate well in the realm of web chatbot usage where users seek detailed information, although the use of voice is expected to increase in the next years (ChatBot, 2023 (Updated in 2024)).

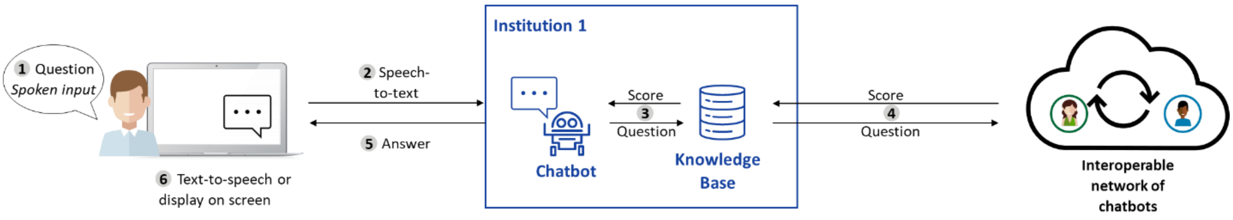


Figure 7. Visual representation of using voice in interoperable chatbots

In the interoperable chatbots scenario, current standards are lacking but this doesn't hinder interoperability. Interactions are processed by converting speech into text, hence chatbots with or without speech support can interoperate seamlessly, with an extra speech-to-text step for voice-supporting bots (see Figure 7). For voice chatbots, speech recognition and Natural Language Processing (NLP) are key for accurate user input understanding and response. Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) are used for speech input translation into processable text (Verma, 2024). Crucial features also include multi-turn conversations and context preservation, enhancing complex dialogues and relevant responses (Rybakova, 2022). Addressing these can help organizations develop interoperable chatbot systems for effective conversational experiences.

1.4.2.3 Licensing and contract considerations

In this section, we will delve into some of the contractual aspects of interoperable chatbots, looking at different documentation recommended through the different stages (more details on each in the appendix B.3.2).

A. Pre-contractual
   - **Non-Disclosure Agreements (NDA):** NDAs are used to ensure that sensitive and confidential information that may be shared between the parties -which will be engaged later on in a contractual relationship- during formal or unformal discussions/negotiations, are identified as such and should not be disclosed to any third parties, unless clearly authorized by law or duly circumscribed in the NDA.

B. Onboarding
   - **Project Agreement:** This clearly defines the scope of the services, roles and responsibilities of each party (including the payment of the related fee by the customer), and assumptions as well as expectations of all parties involved.
   - **Service Level Agreement (SLA):** This defines the level of service expected by the customer from a supplier, laying out the metrics by which that service is measured, and the remedies or penalties, if any, should the agreed levels not be achieved.

C. Data Privacy
   - **Data Processing Agreement:** Establishes rights and obligations of contractual parties for handling shared personal data during the interoperability process and determines details regarding processing of personal data as required by General Data Protection Regulation (GDPR).
   - **Data Protection & Privacy:** Ensures GDPR compliance in data collection, processing, use, and sharing; addresses GDPR principles and security measures.

- **Privacy Notice:** A privacy notice is a statement including details about processing of individuals' personal data conducted by a business. It is a requirement under GDPR to fulfil transparency principle.

D. Other terms
   - **Intellectual Property Rights (IPRs):** The EIF tends to favour the use of open standards and specifications regarding IPRs and has been updated in 2017 in hopes to further facilitate interoperability.
   - **Copyright:** The EU Software Directive provides some allowances with copyright law, stating reverse engineering to achieve interoperability is not an infringement, given specific conditions are met (European Parliament & Council, 2009).

Having a clearer understanding of the contractual requirements to consider for interoperability between chatbots, we now turn our attention to another important point for interoperability, namely the impact of multilingualism.

1.4.2.4 Language considerations

Multilingualism has a significant impact on interoperability between chatbots, as it introduces challenges related to language coverage, translation accuracy, etc. These factors are crucial to consider in chatbots as they can hinder the capability of chatbot systems especially when considering a low-resource language. In this next section we will investigate multilingual aspects to consider in interoperability.

Impact of language on interoperability

Interoperability will require different bots across different organisations and countries to communicate, bringing multilingualism under the magnifying glass. The EU has a rich linguistic diversity with over 200 languages spoken of which 24 languages are recognized official languages (Tirosh, 2024).

Low resources languages (LRLs), also known as under resourced languages, low-density languages, or resource-poor languages, are characterized by their limited digital presence, scarcity of linguistic experts, and inadequate electronic resources for speech and language processing (Ranathunga, et al., 2023). In these languages, there could be a lack of essential tools such as pronunciation dictionaries, vocabulary lists, and other necessary resources for language analysis and development (Besacier, Barnard, Karpov, & Schultz, 2014). Various metrics are employed across different research papers, thus no universally agreed-upon list of low-resources languages, and more resource could become available for a particular language, transitioning the language from low-resource to high-resource.

The table below presents the 24 official languages of EU (European Union, n.d.) and the ones considered as Low resources languages (LRLs)[7] by two studies:

Table 5. Europe's 24 official languages and low resource languages: Study A (Del Gratta, Frontini, Khan, Mariani, & Soria, 2014)[8] and Study B (Alves, Thakkar, & Tadić, 2020)[9]

		Considered low resources				Considered low resources	
	Language	Study A[8]	Study B[9]		Language	Study A[8]	Study B[9]
1	Bulgarian	X		13	**Irish**	**X**	**X**
2	**Croatian**	**X[10]**	**X**	14	Italian		
3	Czech		X	15	**Latvian**	**X**	**X**

[7] For the scope of the PoC on chatbot interoperability, we consider a language as low-resource if one of the two studies mentioned in this section consider it as a low-resource language.

[8] Del Gratta, R., Frontini, F., Khan, A. F., Mariani, J., & Soria, C. (2014, May). The LRE Map for under-resourced languages. In Workshop Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era, Satellite Workshop of LREC (Vol. 14).

[9] Alves, D., Thakkar, G., & Tadić, M. (2020). Evaluating language tools for fifteen EU-official under-resourced languages. arXiv preprint arXiv:2010.12428.

[10] For Study A, drafted in 2014, Croatia was considered, but not as a European Language as they joined the European Union in 2013.

		Considered low resources	
	Language	Study A[8]	Study B[9]
4	**Danish**	**X**	**X**
5	Dutch		
6	English		
7	**Estonian**	**X**	**X**
8	**Finnish**	**X**	**X**
9	French		
10	German		
11	Greek		X
12	**Hungarian**	**X**	**X**

		Considered low resources	
	Language	Study A[8]	Study B[9]
16	Lithuanian	X	
17	**Maltese**	**X**	**X**
18	Polish		X
19	Portuguese[11]		X
20	Romanian		X
21	**Slovak**	**X**	**X**
22	**Slovenian**	**X**	**X**
23	Spanish		
24	Swedish		X

Chatbots integrating low-resource languages face issues mainly stemming from limited (training) data:

- Limited NLP Model Coverage: Low-resource languages can have insufficient Natural Language Processing (NLP) method coverage, affecting chatbot communication, understanding, response-generation, and thus interoperability (Paul, Latif, Adnan, & Rahman, 2019).
- Inaccurate Language Nuances Understanding: Expert models struggle to capture intricacies, and complexities, especially with scarce training data, leading to potential misunderstanding of user queries, inaccurate responses, and reduced interoperability.
- Translation Challenges: Accurate machine translation is difficult for less-documented languages and can affect response accuracy and interoperability between chatbots.
- Thorough Testing: Limited language resources necessitate meticulous responses scrutiny with intensive testing and evaluation processes, potentially constraining the chatbot's value and impact.

Approaches and techniques in language processing and translation

There are several potential translation approaches to overcome some of these low-resource limitations to apply to chatbots. The main two categories of approaches are:

A. Indirect approaches - These techniques offer methods of enabling communication across languages using an intermediary language or steps. Indirect approaches include pivot languages and human translation.

Pivot language translation: These helps circumvent limited bilingual resources through a third language (pivot language) which is used to translate between the source and target languages (Paul, Finch, & Sumita, 2013). This process involves two steps: First, translating the source language to the pivot language using source-pivot trained statistical translation models. Second, Translating the pivot language translation into the target language using a second translation engine trained on pivot-target resources. This technique aids in translating between languages lacking bilingual resources, however it may diminish translation quality due to potential errors in the two-step process (Zaiets, 2021).

Figure 8. Pivot languages

Human translation: This involves interpreting text or speech between languages while preserving cultural nuances and the original form. It captures complex sentence structures and meanings often missed by machine translation

[11] As the study categorized Portuguese in European languages, European Portuguese was considered.

and can handle dialects or idioms that automated systems might stumble over, thereby boosting interoperability. However, it can be time-consuming and inconsistent with multiple translators, may introduce biases, and involve considerable costs. For low-resource languages, collaborations with native speakers or engaging professional translation firms can be beneficial. They can evaluate translation quality, understand cultural contexts, and identify terms that may be meaningless in certain languages. Despite the complexities, multilingualism enhances chatbot interoperability.

B. Direct approaches - Apply translation techniques (e.g., machine learning techniques, or machine translation services to process natural languages) directly from source language to the target.

Natural Language Processing (NLP): Driven by machine learning, NLP approaches comprehends and responds to user input contextually, without predefined replies (Adamopoulou & Moussiades, 2020). Techniques include *Cross-lingual transfer learning* and *multilingual models (see more detail in appendix A.B.2.3.2).* Though translation accuracy might be restricted due to limited training data, continuous NLP and machine learning advancements offer prospects of improvement.

Machine Translation (MT): This involves using software to translate text between languages. Two prominent trends in MT include Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). While SMT demands fewer resources, NMT offers more accuracy by modelling entire sentences in a single integrated model, with BLEU scores (Tsang S. , 2022) used to evaluate machine translations (*see more detail in appendix A.B.2.3.2*).

Large Language Models (LLMs): LLMs mark a turning point in chatbot evolution by generating human-like text. They are trained on extensive text corpus to be able to translate and comprehend numerous languages, especially high-resource languages (Botpress, 2023). Performance with low-resource languages is inconsistent due to limited data, but continuous advancements and sufficient training hold promise for better translations (*see more detail in appendix A.B.2.3.2*).

Translation APIs: Google Translate or Microsoft Translate provide Translation APIs which can enhance a chatbot's multilingual capability by offering broad language support and credible translation quality usually bi-directionally (Church, 2018).

Additional in-depth information on the different translation approaches can be found in Appendix B.3.3B.3.3.

1.4.2.5 Security considerations

Interoperable chatbots' security can be fortified via specific measures. Key aspects include exclusive communication with desired chatbots and abuse protection by employing data encryption, authentication, and protocols. Aspects such as privacy will be covered more extensively in section 1.6.

Ensuring communication with only desired chatbots

Exclusive communication with desired chatbots can be ensured by contract requirement, request routing, and metadata access. The interoperability contract determines privacy data protection (more details in Section 1.4.2.3 Licensing and contract considerations). Also, designing chatbots to interact only with predefined options limits communication to trusted and authorized chatbots. Furthermore, this "List of Contacts" of the chatbot can have access to metadata of other chatbots, allowing it to verify whether they can answer questions, this adds an additional layer of verification and ensures that requests are directed to the most suitable chatbot.

Protecting chatbots from abuse

Preventing resource abuse: Rate limiting prevents resource abuse by restricting request rates from each chatbot, ensuring system performance. Additionally, authentication and authorization mechanisms limit interaction to authorized chatbots. Protecting chatbots from abuse can be done using HTTPS encryption and tokens. HTTPS encryption, a widely used protocol, secures chatbot communication by encrypting transmitted data, preventing unauthorized access. For interoperable chatbots, this safeguards confidentiality and integrity of shared information. A Token or Key system enhances security by assigning unique identifiers to authorized chatbots, controlling interactions and reducing unauthorized access risk.

Protecting chatbot components: Requires identifying each bot's components, addressing potential security concerns, and implementing guardrails in Language Models (LLMs) to avoid divulging sensitive data. For LLMs susceptible to prompt injection attacks, defences include input validation, prepared statements, and intrusion detection.

1.4.2.6 API considerations

APIs allow different software entities to communicate with each other. Adhering to standardized APIs ensures that software components work together seamlessly, providing a reliable and consistent experience. Focusing on best practices, including clear documentation, version control, error handling, and security, helps ensure that the API remains robust and adaptable to future needs.

API Standards considerations

In interoperability between chatbot APIs would be used in two ways:

1. **Communicate User Query to Chatbot Network:** Route user queries to the appropriate chatbot or interoperability transfer components (see 1.5 Viable approaches for more information on PKD and IL that can facilitate such interoperability) using an API.
2. **Direct Communication with Selected Chatbot:** After selection, facilitate communication directly between the user and the specific chatbot.

Transitioning to stakeholders involved in APIs, it's important to understand the roles of API providers and API consumers. API providers build, expose, operate and maintain APIs. They provide APIs to others such as API consumers that develop apps that use APIs (Bouza, 2018).

API specifications define a standard way API of structuring and handling requests and responses. Key elements include:

- Authentication and Authorization: Secure methods such as OAuth 2.0.
- Message Formats: Standardized formats like JSON or XML.
- Endpoint Definitions:
  Query Endpoint: To initiate the user query and identify the appropriate chatbot (e.g., /API/query with parameters query_text, user_id).
  Communication Endpoint: To facilitate direct communication with the selected chatbot (e.g., /API/chat/{bot_id} with parameters message, session_id).
- Error Handling: Define standard error codes and messages for user-friendly error responses.
- Rate Limiting and Throttling: To manage API requests and prevent abuse.

A mandate setting general standards for chatbot interoperability could have significant impacts. Although such a mandate does not currently exist, examples of similar initiatives have emerged, such as PSD2 in banking, which mandates banks to open their payment services and customer account information to third-party providers through APIs, fostering competition and innovation. Following the PSD2 example, the following illustrates the potential opportunities such a mandate would bring for chatbot interoperability:

- Open Access: Chatbot platforms must expose APIs that allow third-party developers to access their services.
- Standardization: Adoption of standardized API specifications to ensure seamless communication between different chatbots.
- Enhanced Interoperability: Different chatbot platforms can interact with each other, enabling diverse chatbot functionalities within a single ecosystem.
- Innovation and Competition: Encouraging innovation by allowing developers to build on top of existing chatbot platforms, promoting competition and better services for end-users.

For additional information on API types and protocols refer to B.3.4. Having reviewed the key considerations for interoperability in connecting chatbots, the focus will now shift to exploring the impact of interoperability on user experience (UX) and user interface (UI) design, and the considerations that need to be addressed.

1.4.3 The impact of UX/UI principles on interoperability

In terms of UX/UI considerations for interoperable chatbots, there are several significant factors to keep in mind. It is important to note that throughout the UX/UI considerations, it is crucial to involve users as much as possible and thoroughly test different approaches to determine what works best for them. For the purpose of this section, we refer to the **host bot** and the **contributing bot** as different chatbots in the interoperable network. The host bot is the bot where the conversation started and where the conversation is ongoing, and the contributor bot is any bot where the host requests additional information from to share with the user.

1.4.3.1 Conversational flow elements

Ensure design elements align with platform capabilities, flows, and languages, favouring existing front ends with minimal changes. Factoring interaction facilitation with other chatbots is crucial as users might be unaccustomed to bot interoperability. Consistency in language and flows makes for a seamless experience. Vocabulary adjustments should be considered on a case-by-case basis.

Feasible options for redirection

Regarding interoperable chatbots, a bot lacking specific knowledge can redirect the user to another bot with expertise. Redirecting within an interoperable network can occur in three ways, each requiring user consent and specific techniques for a seamless experience.

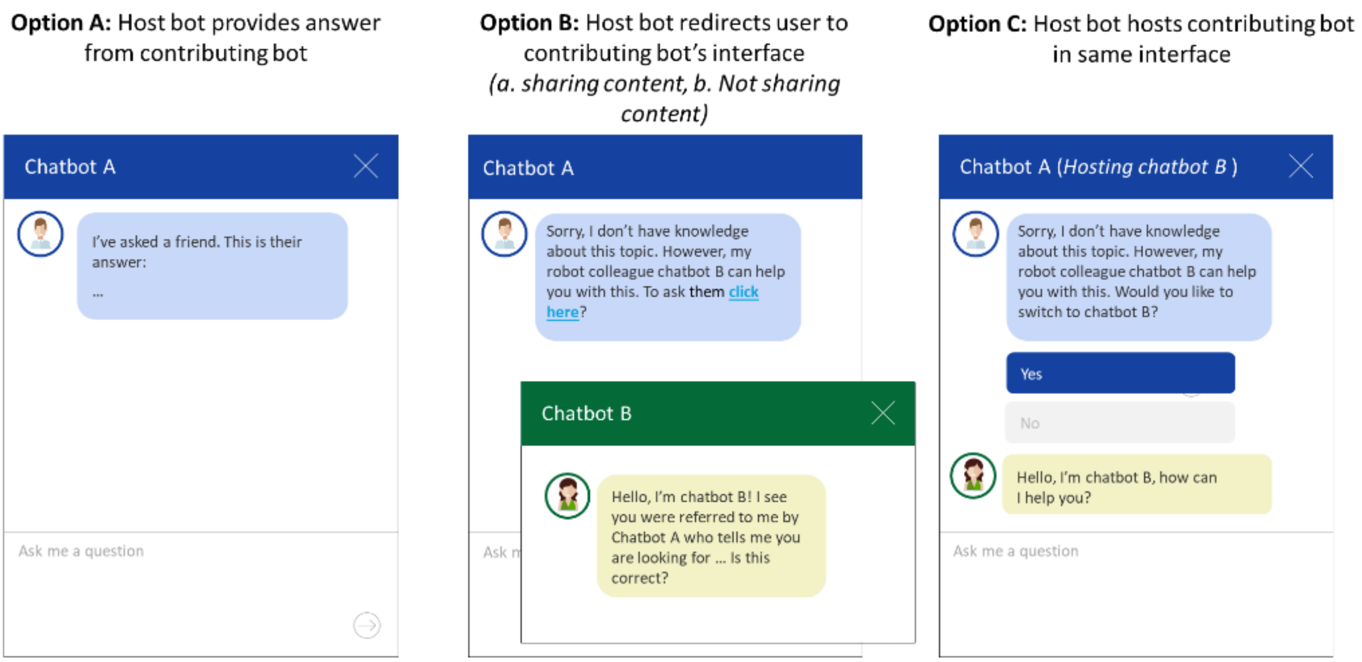
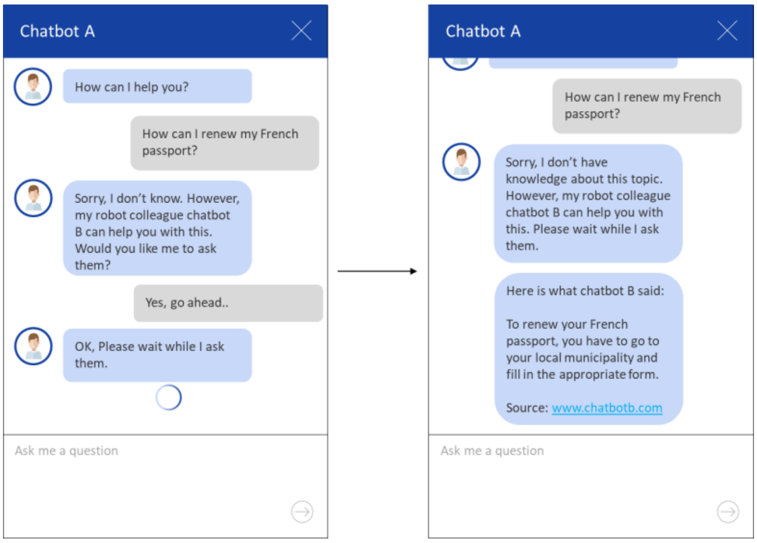


Figure 9. Options for redirection

<u>Option A:</u> Host bot provides answer from contributing bot

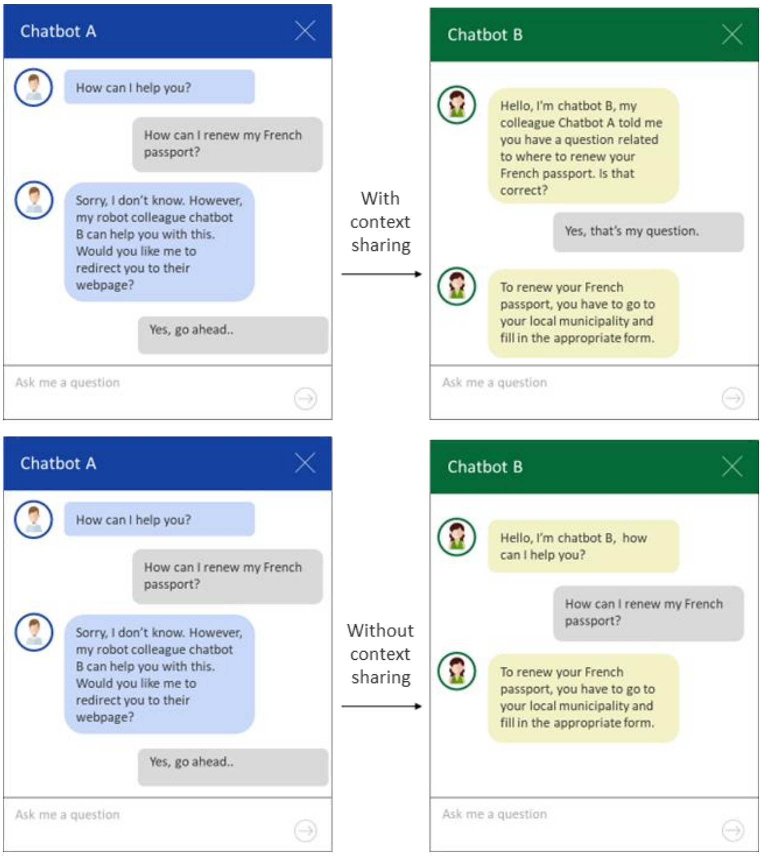


In this scenario, the host chatbot may not immediately switch to the other bot but instead forwards the user's request and waits for a response. In the case of sensitive information, the chatbot can also prompt the user about the option to ask another bot the question to gain consent to share the content.

Additionally, the website or application of the contributing chatbot can be provided, if desired.

Figure 10. Answer in same interface

<u>Option B:</u> Host bot redirects user to contributing bot's interface (a. sharing content, b. not sharing content)



The host bot proposes the user to switch to the knowledgeable bot, and upon user consent, they are redirected to the webpage of the new bot. Users have two options: (a) the original host bot can share the details of the conversation with the new bot to provide context, ensuring that the user does not need to repeat their questions. This enhances the user experience as it is more convenient for the user but raises privacy concerns and comes with legal considerations. Alternatively, for simplified applications or personal information, a scenario (b) without context sharing is also possible, meaning the user will need to provide the necessary details again to the knowledgeable bot.

For option B, the user will have a new tab opened and will therefore know of the existence of both bots, but can in the future directly enter either, depending on the questions they have.

Figure 11. Redirected to different bot in separate interface

Option C: Host bot hosts contributing bot in same interface

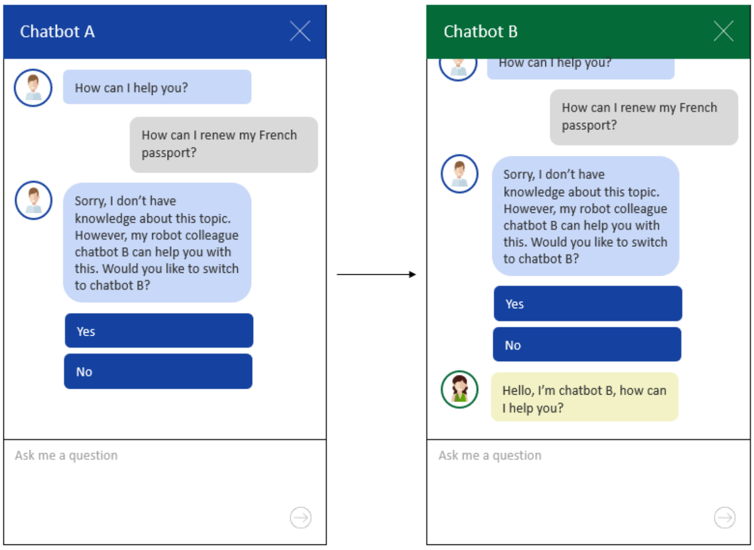


Figure 12. Redirected to different bot in same interface

In this approach, the main chatbot informs the user that it does not have the answer but knows another bot that is more likely to. The user can switch to the other bot without leaving the current interface.
In this example, certain features can make the interoperability clearer. For instance the top bar, text bubble colour and bot identity icon for the different chatbots are unique and showcase that the user is discussing with another bot. The users will have the seamless experience, without having to switch interfaces and all previous responses and context is automatically stored and shared.

All examples demonstrate how the host bot could effectively handle redirection by leveraging the capabilities of other chatbots. The visual representation of the chatbot might differ greatly between chatbots, size of interface, and more, see Figure 53 in Appendix B.4.3.

Overall, the effectiveness of redirection options between chatbots greatly depends on the type of question asked by the user and identified intent by the chatbot. Chatbot answer types can be classified in two categories: direct answers and pre-defined flows. For direct cases, the question asked by the user was identified by the chatbot as a certain intent with one direct answer (e.g., User: What is the population of France, Chatbot: 65 million). In pre-defined flows, the question asked by the user was identified by the chatbot to require more information before a final answer can be provided and will therefore trigger a pre-defined series of questions in order to provide an appropriate answer (e.g., User: I am looking for a government job, Chatbot: Are you a Luxembourgish or international citizen? User: Luxembourgish, Chatbot: Click on the following link to see offers).

For scenarios where direct answers can be provided, option A can be more effective, which involves a straightforward handoff where the host bot asks a question and the contributing bot can provide a single, precise reply. This method ensures a seamless and efficient user experience with minimal transition disruption. However, in cases where the question triggers a predefined flow because additional information is needed, option B or option C can be more suitable, as option A is simplified and might not facilitate the best user experience. Option B employs a more structured approach asking a series of questions to collect the necessary details to provide an answer whereas option C hosts the contributing bot to follow their own flow. Option B would be best if only two chatbots decide to interoperate while option C might be more suitable for a larger interoperable network. Option C is especially well-suited when a lot of switching between the available chatbots is expected as this would preserve a good user experience compared to switching between multiple websites. Below is a more detailed comparison of the different redirection options. More information is provided on redirection returns in the next section, Table 7Table 7.

Table 6. Chatbot redirection options comparison

Criteria	Option A: Direct Answer from Contributing Bot	Option B: Redirect to Contributing Bot Interface	Option C: Host Contributing Bot in Same Interface
Technical Complexity	• Low	• Low	• High
Technical Requirements	• API integration for querying contributing bot • Data parsing and formatting	• API integration for redirect functionality	• API integration for multiple bots • Interface adaptation for multiple bots
Constraints	• Host bot is responsible for content from contributing bot • Limited UI changes needed	• User might not return to host bot • Possible broken user experience due to context switching	• Requires robust front-end design to differentiate between bots • Potentially higher resource consumption
User Experience (UX)	• Seamless and quick responses • Less control over the quality of the response	• Context switching might be confusing for users • User initiative required for redirection	• Smooth UX if properly implemented • Unified interface with clear differentiation of bots
Maintenance & Updates	• Requires regular testing to ensure API changes are managed	• Less direct control over the contributing bot's changes • Updates needed if redirect paths or contributing bot interfaces change (very unlikely)	• Higher maintenance due to complexity • Constant synchronization between bots and session handling
Scalability	• More scalable as it requires minimal changes for each new contributing bot	• More scalable as it requires minimal changes for each new contributing bot	• Less scalable due to increased complexity with each additional bot
Security	• Potential security concerns with regards to data exchange & sharing	• Less security concerns as it redirects to other chatbot website and interface	• Potential security concerns with regards to data exchange & sharing
Development time	• Moderate development time as it involved API integration with minor changes	• Moderate development time as it involved API integration with minor changes	• Longest time due to potential complexity in front-end development and integration
Use cases	• Best for direct answers	• Best for pre-defined flows	• Best when a lot of switching is necessary and there are multiple bots in the network

There are some key considerations to take into account when redirecting. When redirecting a query to another intent-based chatbot, it's crucial not to alter the responses from the contributing bot, as this would necessitate reworking its intent replies. Instead, a better approach would be to prepend the response with a statement like, "Here is the information provided by chatbot B:" before the contributing bot's reply as shown in Figure 10. It is also important to avoid interrupting the transcript of the intent-based chatbot for multi-part questions. For instance, if the chatbot needs to ask three additional questions as part of the conversation flow to provide a complete answer based on the initially identified intent, stopping the conversation before it concludes can impact both the contributing and host chatbots (e.g., chatbot statistics). Lastly, when dealing with chatbots for public institutions, it is crucial to consider which competent authority's chatbot is best suited to respond to users. The chosen chatbot should be the most reliable and have the most trustworthy and authentic sources of information. If the authentic source chatbot is unable to provide a reply, it is worth considering whether another chatbot should be permitted to respond to the user.

Transfer mechanisms

Various methods can be utilized to establish a connection between different chatbots. Three transfer methods will be described

Reactive transfers – Reactive transfers are triggered when the user queries the host bot something that is outside the host bot's knowledge base, but the host bot is aware of a contributing bot in the network with information that is suitable. The host bot can the trigger the transfer in a predefined way (trigger options described in 1.4.3.1 In terms of UX/UI considerations for interoperable chatbots, there are several significant factors to keep in mind. It is important to note that throughout the UX/UI considerations, it is crucial to involve users as much as possible and thoroughly test different approaches to determine what works best for them. For the purpose of this section, we refer to the **host bot** and the **contributing bot** as different chatbots in the interoperable network. The host bot is the bot where the conversation started and where the conversation is ongoing, and the contributor bot is any bot where the host requests additional information from to share with the user.

1. Conversational flow elements).

2. **Proactive transfers** – Proactive transfers happens in similar situations as Reactive transfers. Thus, the context of the question is not suitable for the host bot to answer, however the host bot is aware of a contributing bot in the interoperable network with the needed domain of knowledge. Additionally, to the reactive transfer, in the proactive transfer the bot will also provide a small context before redirecting the user to the relevant bot. In these types of handovers, the users are proactively given more information and not only the option to transfer.

3. **Manual transfers** – Manual handovers required the users to explicitly ask to talk to a certain chatbot or organisation. The user needs to be aware of the existence of the bot or bot network and can be suitable for users that have been referred by the host bot before, but for instance does not know how to directly reach the referred bot. Manual transfers triggers can differ and should be defined. Some examples can include:
   “Can you transfer me to [Bot B]?”
   “I want to discuss with the bot from [Organization B]?”
   “I need advice from the [subject: Tax] bot”.

A combined approach, enabling all three methods to provide a user-friendly experience and a fluid interoperability between their bots is advisable (Miessner, et al., 2019).

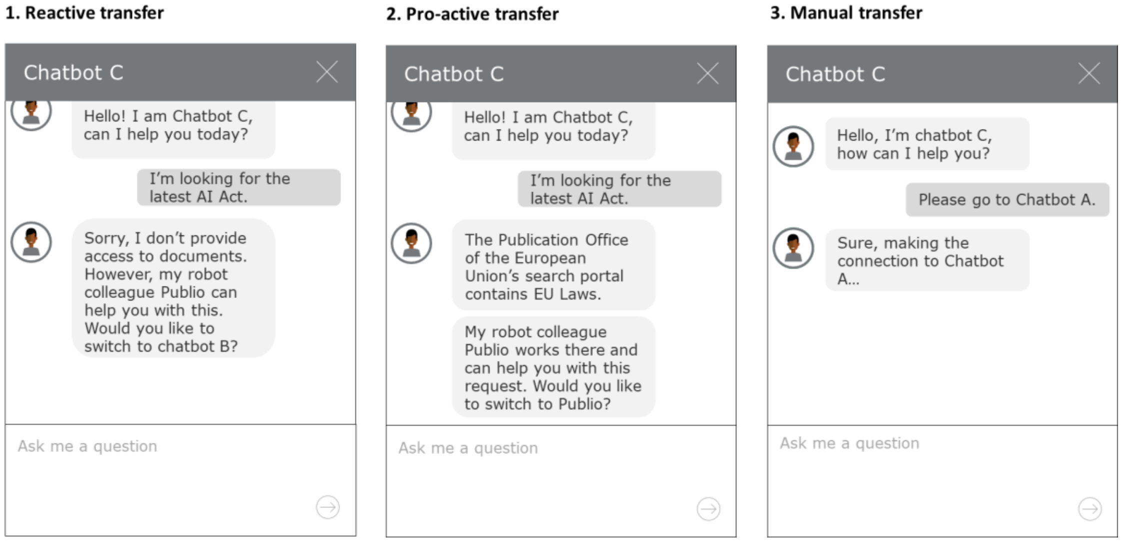
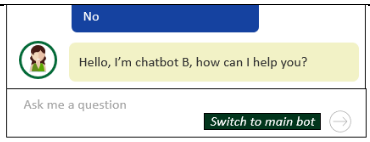


Figure 13. Example of the three methods to transfer the conversation

Redirection return options

Regarding switching back to the host bot, several possibilities can be considered to help the user.

Table 7. Redirection return options and examples

Return options descriptions	Return examples
User prompt – If the user does not know the specific intent to trigger the return, the bot can simply ask to go back to the host bot after a certain period of time. This can also be actively asked by the user.	E.g., *Chatbot B*: "Would you like to switch back to chatbot A?" E.g., *User*: "I want to go back to the previous chatbot"
Add a button – Provide a clearly labelled button, such as "Return to Bot A" to easily navigate back. This could also be added to an existing menu for example.	E.g., 
Usage notice – Inform users from the start about the presence of multiple bots.	E.g., *Chatbot A*: "This chatbot is connected to two bots, one dedicated to tax, and one to immigration. Do not hesitate to ask about these topics."
Command alias – Create a simple command like "/home" that users can use to instantly return to the host bot. Users should be informed about this functionality	E.g., Chatbot A informed the user at the beginning of the switch to chatbot B that they could switch back with "/home". *Chatbot B*: … *User*: "/home" *Chatbot A*: Hello,…

Besides the redirection options, other aspects need to be considered. For example, there are some cases where it makes sense to disable the option of switching to another bot. This can be relevant of a predefined flow has been triggered and the user is already deep inside this flow. This helps to maintain focus and prevent context loss. For a comparison of the three redirection return options, refer to Appendix B.4.1.

Context handling

When discussing interoperable chatbots between public institutions, the ability to redirect users between chatbots is crucial. There are two main approaches: transferring context and not transferring context. Transferring context involves passing conversation details like previous messages and user intent between chatbots. This ensures a seamless user experience, improved accuracy, and better problem-solving but presents technical complexity, privacy issues, and increased processing loads. Additionally, the redirection option impacts context transfer difficulty; for instance, when shifting between websites keeping the context can be more challenging than staying within the same interface.

Conversely, not transferring context means each new conversation starts fresh, simplifying implementation and preserving privacy but potentially leading to redundant user input and limited problem-solving. The choice depends on integration levels, service complexity, privacy regulations, and technical resources. A hybrid approach, where basic info is transferred but detailed info is kept separate, might be most effective. The goal is to create chatbots that provide efficient, accurate, and user-friendly public service access while maintaining privacy and security standards.

1.4.3.2 Design elements

One key consideration is how to align visual interface components between chatbots, such as the number of characters in text, button sizes, feedback buttons, emojis, and picture sizes. On several design elements we recommend the different chatbots interconnecting to align to facilitate interoperability and avoid user inconsistencies which are risks previously seen. It should be noted that the button size and other visual interface aspects can be handled by the API, reducing the need for manual adjustments in this regard (in the case that the conversation remains within the original bot and not redirected to the other bots in the network).

Number of characters in text: A bot’s quick reply element might have a character limit for its response buttons. To establish consistency and avoid errors when bots forward messages to each other, we recommend taking the lowest word count option in the available chatbots and make it the standard for all in the interoperability network (unless doing so would negatively impact the user experience, for instance specific use cases where extensive explanations are required by the user such as a description of an accident for insurance purposes). There is also the option to request the sending bot to shorten the text to fit into the limit of the receiving bot, but context can be lost and therefore alignment of character limitations is preferrable. Prior to implementing any changes, it is advisable to track the average response length and assess the feasibility of the proposed adjustments.

Feedback buttons: Feedback design element plays a crucial role in determining the user experience while interacting with a bot. In **Error! Reference source not found.**, different feedback options are showcased:

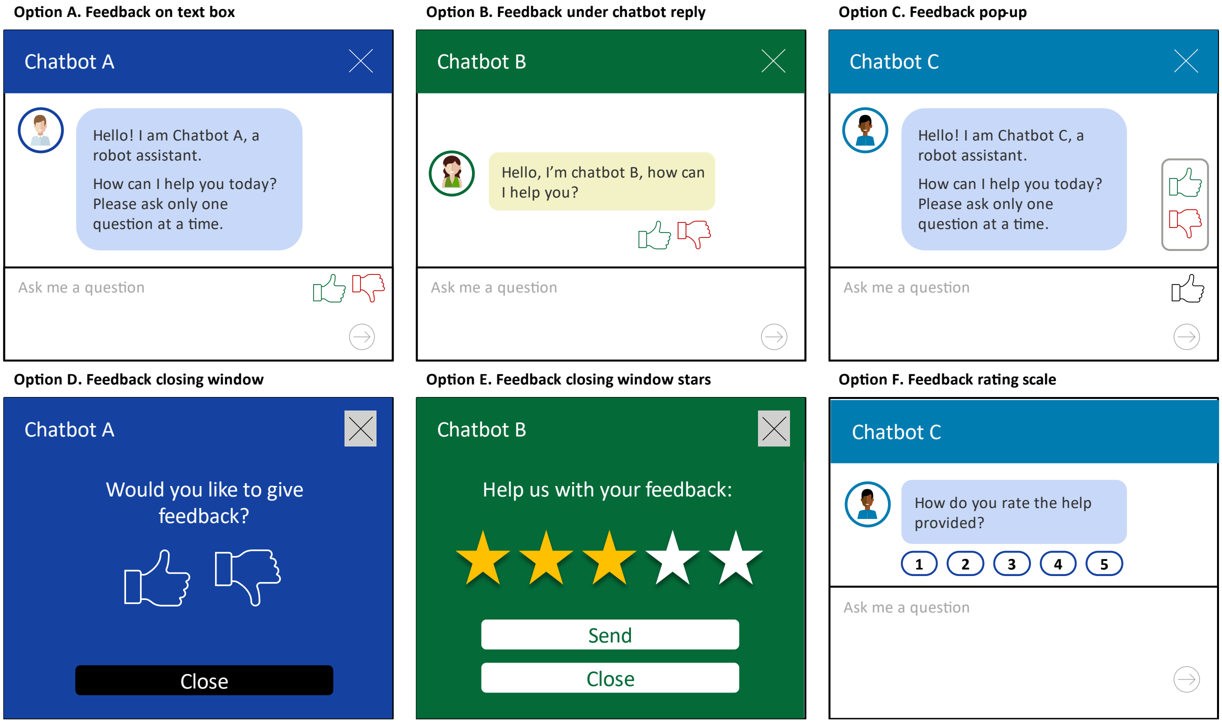


Figure 14. Potential feedback options

When considering interoperability between different chatbots, several factors related to feedback mechanisms need to be taken into account such as the origin of the conversation, the impact on cost, and the feedback trigger mechanism. When feedback is given, not only the evaluation or comment left by the user is considered, but also the conversation transcript. This helps in understanding the context of the evaluation/comment. Feedback is then analysed and noted for improvements before being deleted after a certain period (in compliance with GDPR rules). Furthermore, the origin of the conversation needs to be identified to determine which bot handled the conversation and whom the feedback is addressing. To effectively track conversations and identify which chatbot responded to each query, each chatbot must have a unique ID. This ID is essential for accurately viewing logs and determining the source chatbot. Additionally, the impact on cost is essential, as some chatbots incur a cost per conversation or an incremental fixed cost based on the number of conversations. Therefore, it is critical to link the feedback to the bot that handled the conversation so it can have the relevant statistics. This requires a seamless feedback trigger mechanism, where every time a bot is activated by a user query, the conversation transcript and user feedback are sent to the respective bot. Both chatbots need a dedicated API to receive the combination of feedback results and the conversation transcript.

Media support: Some chatbots might support rich media inputs like images and voice messages while others might be limited to text-based only interactions as seen in the example of the Figure below. This difference could limit the information exchange. For instance, a chatbot designed to send photo responses will face issues on a platform that only supports text. A lack of standardized media support across platforms could result in inconsistencies in user experience. For more information on this, refer to Appendix B.4.2.

In the case of interoperability facilitated through switching interfaces between different bots (or hosting a contributing bot within the host bot's environment), there are additional considerations, as shown in Figure 15.

Moving forward with chatbot design elements, the focus will shift towards conversational style delving in creating engaging interoperability interfaces, and exploring language and tonality used by chatbots.

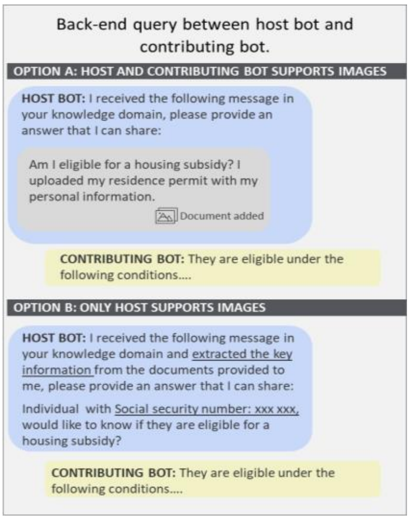


Figure 15. Switching interface options

1.4.3.3 Conversational Style

Chatbot personality

Chatbot personality is the combination of unique characteristics or attributes that form the distinctive identity of the chatbot, enhancing user engagement through mimicking human-like dialogues and behaviours (Ferman, 2018).

Use of emojis: In terms of the conversation style of the chatbots, there are several important factors to take into account. First, it is important to determine which types of emojis can or cannot be displayed, particularly in a text interface. As seen in the Finnish case in 1.3.1.1, chatbots may be limited to positive emojis and flags, used no more than one consecutively. While no specific emoji type is prohibited, problems can arise if a browser is outdated or emojis are viewed on different browsers/platforms/devices, which can disrupt user experience.

Restricted words and topics: Chatbots must be programmed to adhere to cultural, national, and other norms, avoiding taboo words. Each bot maintains its own proscribed word list and should eschew offensive, discriminatory language, explicit content, and discussions about illegal activities (Jackson, 2023). They should avoid politically sensitive matters, derogatory or unsettling subjects, and refrain from seeking personal details like social security numbers or passwords, and cultural norms in different countries could necessitate further restrictions (Mirza, 2023; Fromet, 2021).

Conversation style: Another consideration is the choice between a legal style or a casual style of communication. While it may be challenging to make significant changes to legal texts or terminology, it is important to recognize that legal nomenclature may not be universally understood in all areas. The adaptation from a legal style to a more casual one may depend on the chatbot's connection to other specific chatbots, as the context of the conversation will influence the appropriate communication approach. We propose that the conversational style remains true to the current bot interacting with the user.

This next subsection will look into the user interaction and looking into some of their perspective to consider having a seamless experience.

User interaction

Several aspects have to be considered in the user interaction with the chatbots such as the transparency in the transfers between the bots and the conversation pace.

Transfer transparency: Transparency is a key element in the interaction between user and chatbots. When a chatbot is transparent about seeking an answer from another bot, it gives users a clear understanding of the conversation flow. This transparency is also important to maintain a high-level user **satisfaction**. This is why it is strongly recommended to indicate the source of the information if the source changes during the conversation. Transfer transparency can be facilitated in the following ways:

Ask for consent to switch to another bot and share the conversation history

Adding the source link to enable the user to also interact with the other chatbot directly or read up on additional information on the source website

See the options in the figure below, a comparison of the different options and their pros and cons can be found in Appendix B.4.4B.4.4B.4.4.

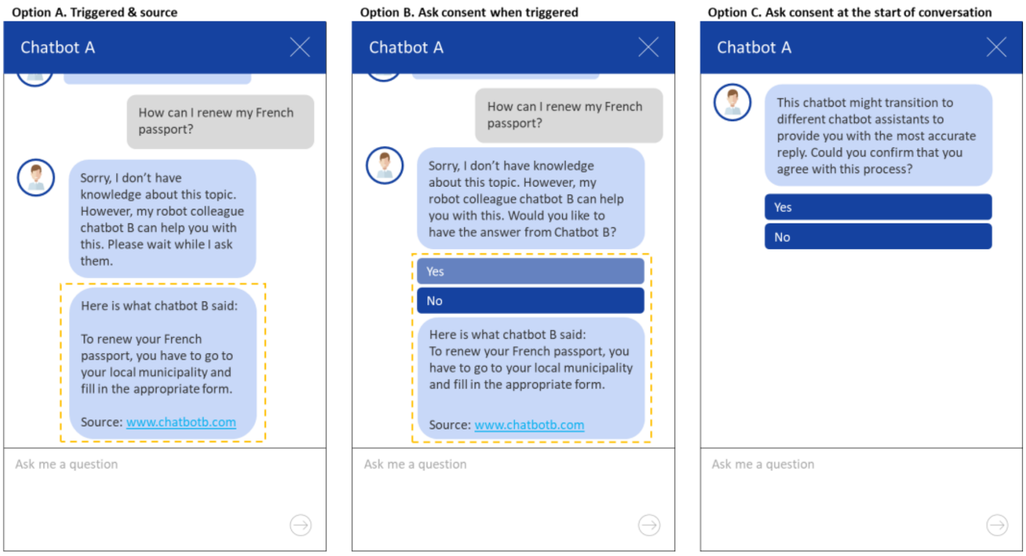


Figure 16. Use transfer disclaimers

Expected interaction speed: Additional UX/UI considerations are the chatbot's conversation pace and language speed. Preventing the bot from rapid-fire messages can avoid overwhelming users. With interoperable chatbots, system slowness could happen hence a progress indicator can be useful to show ongoing actions. While allowing speed of language modification isn't generally necessary, preserving the scroll position is recommended when new messages arrive as users view previous ones.

Response time & latency: Response time is crucial for UX/UI. A high-performing API ideally responds within 0.1-1 second. For interoperable chatbots using an external API, response times hinge on API performance, network latency, and API request handling time. They can vary with API complexity and network traffic. Possible response time optimization strategies include caching mechanisms and asynchronous programming techniques.
Geographical server distance impacts latency, or communication delay between the chatbot and external source. Implementing content delivery networks (CDNs) or edge computing can minimize latency by serving data from closer servers or positioning servers nearer to the user. However, elements such as network infrastructure and internet service providers can also influence latency and are often out of control.

1.5 Viable approaches

1.5.1 Feasible options to facilitate interoperability

This section explores viables methodologies and angles to establish interoperability between chatbots. First, we will investigate the different ways in which chatbot systems can share their domains of knowledge looking at different approaches and options. Then a comparison will be established for the different approaches and their options to compare the feasibility of interoperability.

1.5.1.1 Components to facilitate interoperability

To make interoperability between multiple chatbots possible, several approaches and options can be studied. To apply them, different elements need to be incorporated. Presented below are some of these specific components:

Primary knowledge directory (PKD): A primary knowledge directory can be seen as a global repository that catalogues the available chatbots, their domains of expertise, and the types of queries they can handle (Dhanani, 2023). It allows other chatbots in the network to be aware of the domain of knowledge of the other bots within the network. Here, the PKD would hold the metadata from various chatbots which would encompass details on their capacity, intent, functionality allowing them to interact/ transfer to other chatbots. Key considerations for setup:

- **Intent & entity mapping:** This involves noting the shared intents and entities from the different chatbot systems and creating a mapping between them. Consequently, this map aids in translating user inputs processed by one system to the other. An example map can be found in Appendix 00 with more information about intents and entities within the context of interoperability.
- **Dialog management**: This involves understanding how each chatbot generates responses based on identified intents and entities, manages conversation flows, and tracks context. Each organization should share its conversational flow to facilitate the interoperability mechanism. The aspect will be less relevant for LLM based systems that might not follow a structured conversation flow as an intent-based chatbot. Dialog management can be done by the PKD in two ways:
    - Refers the host bot to the most relevant bot in the network to answer and these bots then connect to answer user in the selected manner
    - Collects the answer directly from the most relevant bot in the network to answer and provides this information to the host bot (taking care of the interaction as a middle dialogue management layer).

EXAMPLE: Chatbot A which is a tax chatbot is asked on how to file an online complaint for theft. Based on this user input ***the PKD*** *will see which other bot is best to connect to – chatbot B an immigration chatbot or chatbot C a Police chatbot (all this information is stored in the PKD). The PKD transfers the user to chatbot C, the Police chatbot that can deal with topics on theft.*

Interoperability Layer (IL): This layer is an intermediate layer between the host bot and the contributor bots. Unlike the centralized directory it does not store the metadata of the all the chatbots, however it has information on all other bots in the interoperable network, which the host bot could transfer to. Key considerations for setup:

a) **Referral mechanism:** The way the referral will take place should be established as all bots in the network should share the information in a suitable format. Options include:
    - **Confidence threshold/ similar metric:** Understand the confidence threshold of each bot to answer the query and evaluate the best to respond based on this or other suitable metrics. This score functions as a threshold where the lowest matching score is considered acceptable to trigger a bot response and the highest score is the bot that can answer the user query best. The proposed default confidence threshold of 70% but can be seen as a benchmark (Kocur, 2023). This can be adapted depending on the needs of each chatbot.
    - **Direct answer:** Interoperability layer will need to be built with mechanism to evaluate the most appropriate response (this could be an LLM). Similarly to the PKD, this component can answer the host bot or refer to the host bot.

EXAMPLE: Chatbot A (host) forwards the question from the user to chatbot B, C and D through the IL. In return, they send the confidence score for this query (indicating how likely each can answer the question). The IL calculates the highest score (if higher than the minimum threshold set) and makes sure that Chatbot A is connected to that bot for getting the related information.

Transfer mechanism: This mechanism acts as a vehicle, effectively translating between chatbot systems, converting user inputs into respective formats and translating back responses to a universally understood format. Conceptually, it operates as a robust translation interface for varying LLM chatbots (Pohrebniyak, 2024). It manages data conversions, intent mapping, and communication protocols, smoothing interactions between diverse chatbot systems. Additionally, the routing could be done by an LLM to understand the query and know which bot would be best to answer the user query.

Another fundamental aspect is the different types of chatbots to interoperate. For example, establishing connection between several intent-based chatbots, only between LLMs, or having a mix of LLMs and intent-based chatbots.

1.5.1.2 Sharing domains of knowledge

This section will look into how chatbots could share their domains of knowledge in order to achieve interoperability, each having a centralized, de-centralized, or semi-centralized option. Refer to 1.7.1 Phase A: Initiation for information on governance.

Approach A: Use primary knowledge directory

The first approach would use a *primary knowledge directory* at the core of the system. As previously mentioned, the primary knowledge directory would have access to the metadata from all the bots in the network.

<u>Centralized interoperability</u>
The use of a primary knowledge directory with a centralized option, shown in Figure 17, works as follows:

Requesting phase:

1) The user sends a query to the host bot

Processing phase:

2) Each query is directly shared to the *primary knowledge directory* (note that this option redirects all decisions to the PKD)
3) The *primary knowledge directory*, with metadata access for all network bots, scores chatbots on their ability to answer user queries and selects the most suitable one.
4) Options:
   a) The *primary knowledge directory* informs the host bot of the most appropriate bot in the network to answer the question based on the highest score
   b) The *primary knowledge directory* serves as an exchange layer to get the answer from the selected bot

Answering phase:

5) Options:
   a) Host bot facilitates interoperability in a pre-defined manner. This is done by directly interacting with the chosen chatbot using the transfer mechanism to ensure interoperability between the bots. Please refer to 1.4.3.1 In terms of UX/UI considerations for interoperable chatbots, there are several significant factors to keep in mind. It is important to note that throughout the UX/UI considerations, it is crucial to involve users as much as possible and thoroughly test different approaches to determine what works best for them. For the purpose of this section, we refer to the **host bot** and the **contributing bot** as different

chatbots in the interoperable network. The host bot is the bot where the conversation started and where the conversation is ongoing, and the contributor bot is any bot where the host requests additional information from to share with the user.

b) Conversational flow elements regarding these interoperability display options:
   i. **Option A:** Host bot provides answer from contributing bot
   ii. **Option B:** Host bot redirects user to contributing bot's interface
   iii. **Option C**: Host bot hosts contributing bot in same interface

c) The exchange layer sends the answer to the host bot

6) User receives the answer to the query through the host bot

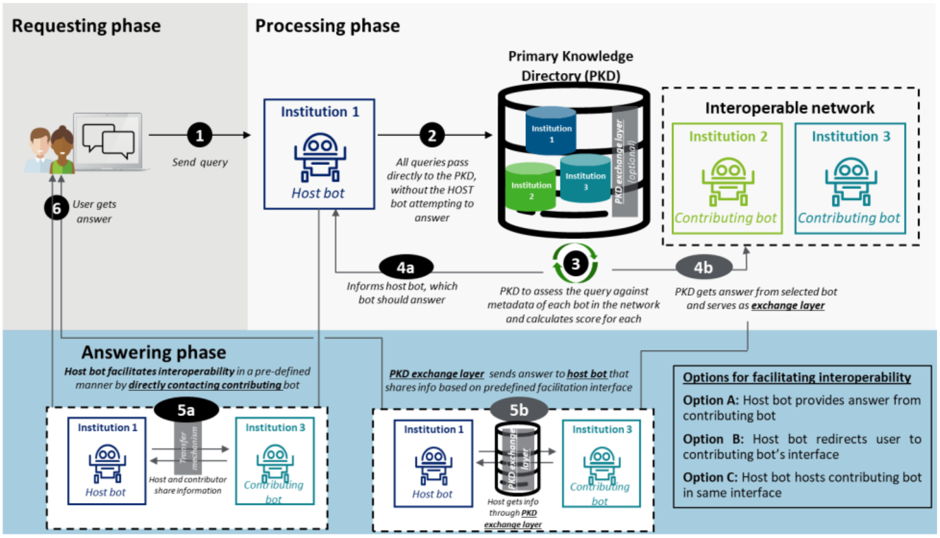


Figure 17. Overview of a centralized approach with a PKD

Semi-centralized interoperability

The use of a primary knowledge directory with a semi-centralized option. This could entail copies of the PKD or separately maintained PKDs but the host bot would try to answer the query before sending it to the network of other chatbots, see Figure 18**Error! Reference source not found.**. The process works as follows:

Requesting phase:

1) The host bot receives a user request

Processing phase:

2) The host bot tries to answer to query and only if it cannot do so (below threshold), the query will be passed to the *primary knowledge directory* (note that this option maintains the decision making within the host bot and does not delegate all decisions to the PKD. This can be facilitated by creating an 'interoperability intent that detect queries with a lower threshold to avoid that the host bot answers all queries.)
3) The *primary knowledge directory*, with metadata access for all network bots scores chatbots on their ability to answer user queries and selects the most suitable one.
4) Options:
   a) The *primary knowledge directory* informs the host bot of the most appropriate bot in the network to answer the question based on the highest score
   b) The *primary knowledge directory* serves as an exchange layer to get the answer from the selected bot

Answering phase:

5) Options:

a) Host bot facilitates interoperability in a pre-defined manner. This is done by directly interacting with the chosen chatbot using the transfer mechanism to ensure interoperability between the bots. Please refer to 1.4.3.1 Conversational flow elements regarding these interoperability display options:
   i. **Option A:** Host bot provides answer from contributing bot
   ii. **Option B:** Host bot redirects user to contributing bot's interface
   iii. **Option C:** Host bot hosts contributing bot in same interface

b) The exchange layer sends the answer to the host bot

6) User receives the answer to the query

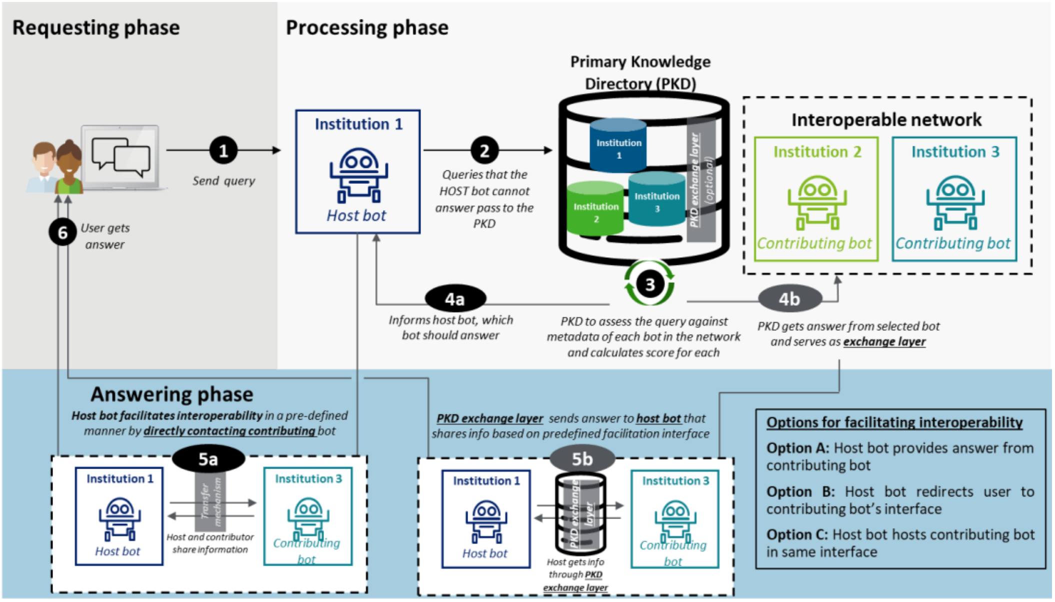


Figure 18. Overview of a semi-centralized approach with a PKD

<u>De-centralized interoperability</u>

The use of a primary knowledge directory with a de-centralized option. This could entail copies of the PKD or separately maintained PKDs. Process shown in Figure 19, works as follows:

Requesting phase:

1) The host bot receives a user request

Processing phase:

2) Each chatbot possesses its own *primary knowledge directory* which has access to the metadata of each chatbot in the network and knows which chatbot is able to answer which intents
3) The *primary knowledge directory* within the host bot returns a score for how well each chatbot is able to answer the user query and chooses the most appropriate bot
4) Options:
   a) The *primary knowledge directory* informs the host bot of the most appropriate bot in the network to answer the question based on the highest score
   b) The *primary knowledge directory* serves as an exchange layer to get the answer from the selected bot

Answering phase:

5) Options:

a) Host bot facilitates interoperability in a pre-defined manner. This is done by directly interacting with the chosen chatbot using the transfer mechanism to ensure interoperability between the bots. Refer to 1.4.3.1 Conversational flow elements regarding these interoperability display options:
   i. **Option A:** Host bot provides answer from contributing bot
   ii. **Option B:** Host bot redirects user to contributing bot's interface
   iii. **Option C:** Host bot hosts contributing bot in same interface

b) The exchange layer sends the answer to the host bot

6) User receives the answer to the query

This approach is similar to the method used for the Finnish chatbot interoperability seen in section 0.

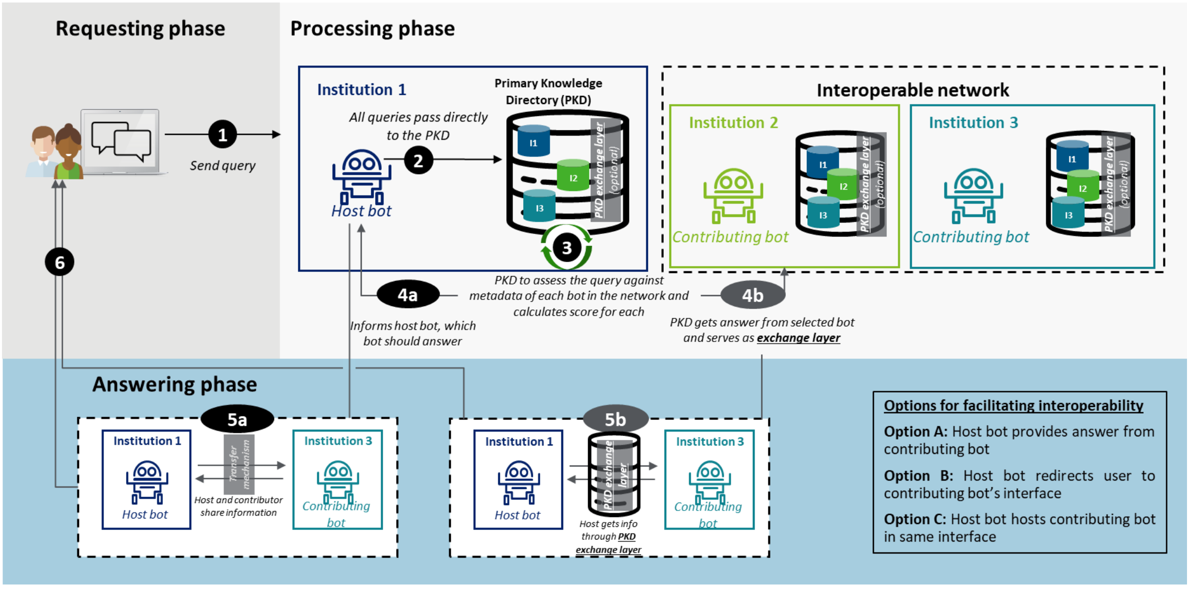


Figure 19. Overview of a decentralized approach with a PKD

Approach B: Use interoperability layer

The second approach would use an *interoperability layer*. As previously mentioned, the interoperability layer would not need any prerequisites regarding sharing metadata.

<u>Centralized interoperability</u>

The use of an interoperability layer with a centralized option, shown in Figure 20Figure 20, works as follows:

Requesting phase:

1) The host bot receives a user request

Processing phase:

2) The user query is shared to the *interoperability layer* (note that this option redirects all decisions to the IL, which in turns takes over the full role of deciding which bot will answer)
3) The *interoperability layer* sends the question to all bots in the network and evaluates each contributing bots response and a score how well they can each answer the question and chooses which bot is more appropriate to answer it
4) Options:
   a) The *interoperability layer* informs the host bot of the most appropriate bot in the network to answer the question based on the highest score

b) The *interoperability layer* serves as an exchange layer to get the answer from the selected bot

Answering phase:

5) Options:

Host bot facilitates interoperability in pre-defined manner. This is done by directly interacting with the chosen chatbot using the transfer mechanism to ensure interoperability between the bots. Please refer to 1.4.3.1 In terms of UX/UI considerations for interoperable chatbots, there are several significant factors to keep in mind. It is important to note that throughout the UX/UI considerations, it is crucial to involve users as much as possible and thoroughly test different approaches to determine what works best for them. For the purpose of this section, we refer to the **host bot** and the **contributing bot** as different chatbots in the interoperable network. The host bot is the bot where the conversation started and where the conversation is ongoing, and the contributor bot is any bot where the host requests additional information from to share with the user.

a) Conversational flow elements regarding these interoperability display options:
   i. **Option A:** Host bot provides answer from contributing bot
   ii. **Option B:** Host bot redirects user to contributing bot's interface
   iii. **Option C:** Host bot hosts contributing bot in same interface

b) The exchange layer sends the answer to the host bot

6) User receives the answer to the query

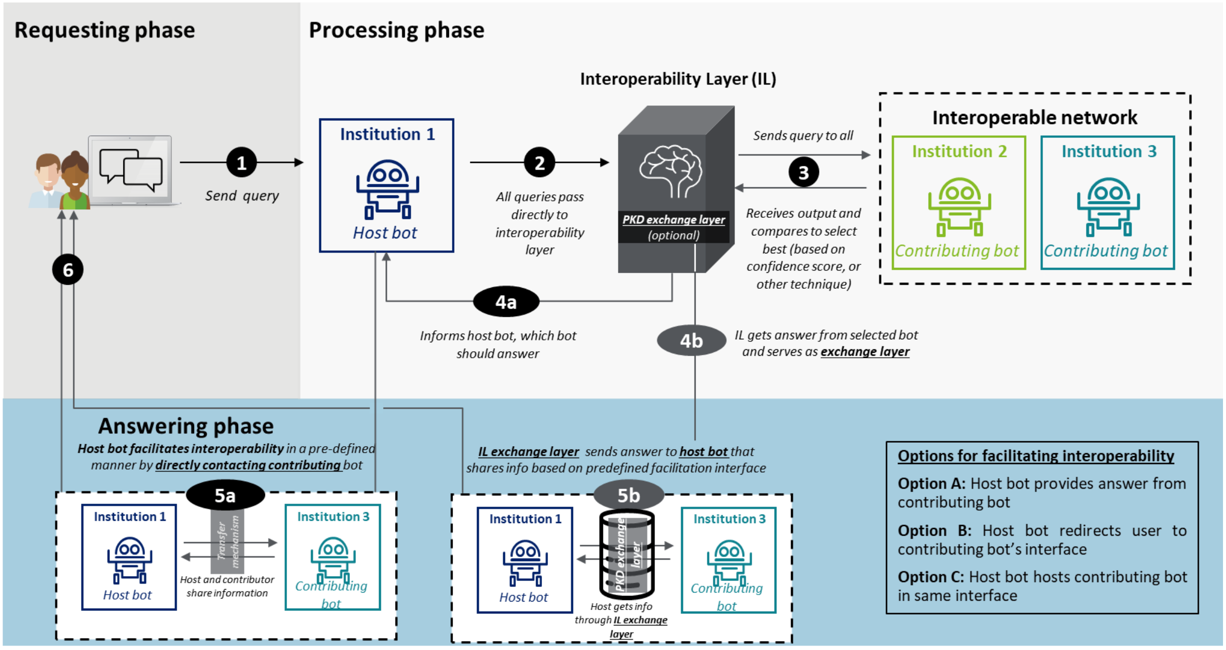


Figure 20. Overview of a centralized approach with an IL

<u>Semi-centralized interoperability</u>

The use of an interoperability layer with a semi-centralized option. This could entail copies of the IL or separately maintained ILs. This process would look as follows:

Requesting phase:

1) The host bot receives a user request

Processing phase:

2) The host bot tries to answer to query and only if it cannot do so, the query will be passed to the other bots in the network
3) Each chatbot possesses its own *interoperability layer* which sends the user request to all the contributing bots in the network. The *interoperability layer* is inside the host bot and will attribute a score to each response the contributing bot are sending back to determine the most appropriate one (note that here the decision making is maintained within the host bot and does not delegate all decisions to the IL. This can be achieved by setting a threshold where the IL is called or creating an interoperability intent that can be triggered when the host bot has a lower confidence.)
4) Options:
   a) The *interoperability layer* informs the host bot of the most appropriate bot in the network to answer the question based on the highest score
   b) The *interoperability layer* serves as an exchange layer to get the answer from the selected bot

Answering phase:

5) Options:
   a) Host bot facilitates interoperability in pre-defined manner. This is done by directly interacting with the chosen chatbot using the transfer mechanism to ensure interoperability between the bots. Please refer to section 1.4.3.1 Conversational flow elements regarding these interoperability display options:
      i. **Option A:** Host bot provides answer from contributing bot
      ii. **Option B:** Host bot redirects user to contributing bot's interface
      iii. **Option C:** Host bot hosts contributing bot in same interface
   b) The exchange layer sends the answer to the host bot
6) User receives the answer to the query

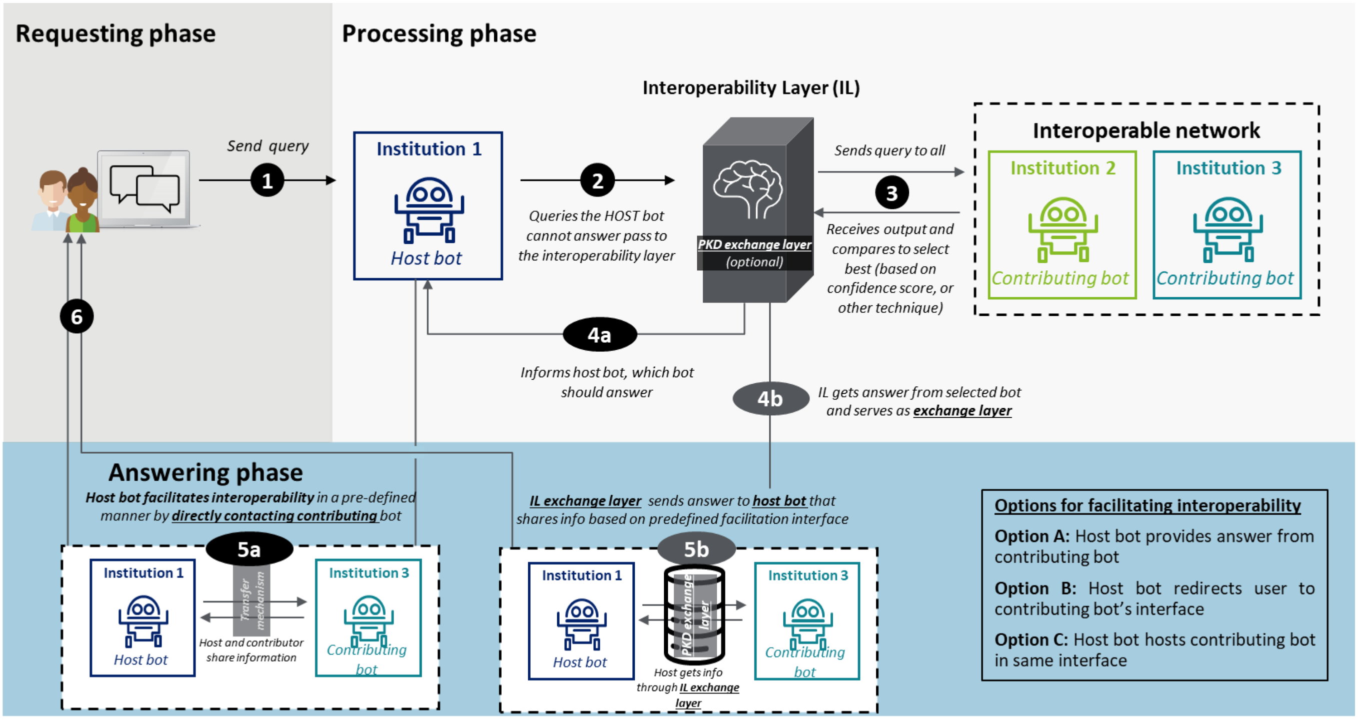


Figure 21. Overview of a semi-centralized approach with an IL

<u>De-centralized interoperability</u>

The use of an interoperability layer with a de-centralized option, shown in Figure 22. This could entail copies of the IL or separately maintained ILs.

Requesting phase:

1) The host bot receives a user request

Processing phase:

2) Each chatbot possesses its own *interoperability layer* which sends the user request to all the contributing bots in the network
3) The *interoperability layer* is inside the host bot and will attribute a score to each response the contributing bots are sending back to determine the most appropriate one
4) Options:
   a) The *interoperability layer* informs the host bot of the most appropriate bot in the network to answer the question based on the highest score
   b) The *interoperability layer* serves as an exchange layer to get the answer from the selected bot

Answering phase:

5) Options:
   a) Host bot facilitates interoperability in pre-defined manner. This is done by directly interacting with the chosen chatbot using the transfer mechanism to ensure interoperability between the bots. Please refer to 1.4.3.1 In terms of UX/UI considerations for interoperable chatbots, there are several significant factors to keep in mind. It is important to note that throughout the UX/UI considerations, it is crucial to involve users as much as possible and thoroughly test different approaches to determine what works best for them. For the purpose of this section, we refer to the **host bot** and the **contributing bot** as different chatbots in the interoperable network. The host bot is the bot where the conversation started and where the conversation is ongoing, and the contributor bot is any bot where the host requests additional information from to share with the user.
   b) Conversational flow elements regarding these interoperability display options:
      i. **Option A:** Host bot provides answer from contributing bot
      ii. **Option B:** Host bot redirects user to contributing bot's interface
      iii. **Option C:** Host bot hosts contributing bot in same interface
   c) The exchange layer sends the answer to the host bot
6) User receives the answer to the query

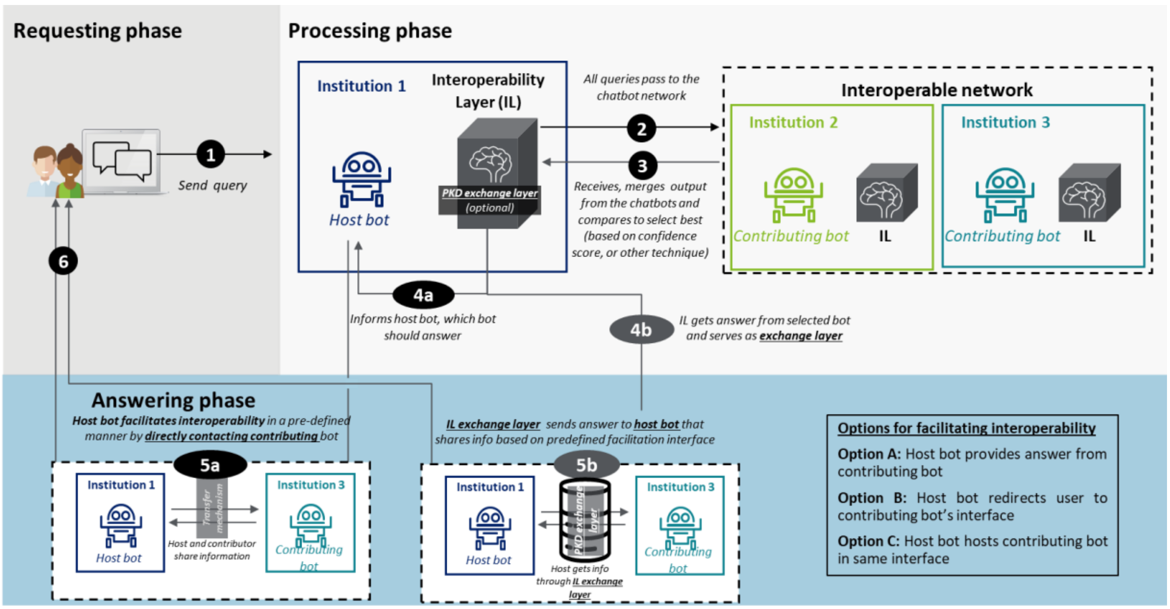


Figure 22. Overview of a decentralized approach with an IL

1.5.2 Comparison of the different approaches

This section will compare the above-mentioned options, looking at gaps and benefits for each approach. These points are in line with some of the risks and challenges outlined in section 1.3.2 Benefits, Challenges & Risks.

Table 8. Comparison of the different approaches

Key: Pro (normal font), *Con (italic font)*

		Approach A: Use Primary Knowledge Directory (PKD)	Approach B: Use Interoperability Layer (IL)
Storage optimisation	Cent.	With centrally stored metadata, every institution has access to the same information, which is only stored once.	No shared knowledge base needs to be built and maintained.
	Semi.	Same as centralized.	Same as centralized.
	Decent.	*Maintaining separate PKD for each institution could lead to data variations or duplication, escalating the likelihood of communication errors between chatbots. Furthermore, the absence of a reference PKD compounds the issue, as the complexity of synchronizing these PKDs potentially increases exponentially with the number of chatbots in the network.*	No shared knowledge base needs to be built and maintained.
Maintenance	Cent.	Central management of metadata permits to implement any changes once. *Storage & maintenance cost will be incurred (increasing with the network).*	All operations are kept in one place, so it is easier to understand, manage and maintain. No metadata storage cost involved. *Decision mechanism needs to be maintained continuously to output relevant information.* *If not properly managed, the central IL can affect the overall performance of the system.*
	Semi.	Same as centralized.	Same as centralized.
	Decent.	*Managing updates across multiple PKDs can be complex, time-consuming, and would require good communication.* *Storage & maintenance cost will incur individually (increase with network).*	No metadata storage cost. *Decision mechanism within the IL needs to be maintained continuously to output relevant information.* *Development and maintenance might be challenging due to all chatbots working independently.*
Scalability	Cent.	*Adding a new bot requires sharing predefined information into the interoperable network in established format (e.g., entities, intents, flow).* Adding and removing information from CKB, once collected, is simple as it is only in one central PKD.	Easy to scale as IL with function regardless of new bots added. Central IL might need to increase its capacity to handle more user questions in case of a large interoperable network.
	Semi.	Same as centralized.	Same as centralized.
	Decent.	*Adding a new bot requires sharing predefined information into the interoperable network is established format (e.g., entities, intents, flow).* *Adding or removing collected information from the CKB is challenging as each bot in the network maintains their own PKD.*	*Each bot in the network requires an IL to be setup (intercepting answers and facilitating the best response).*
Response	Cent.	Fast response time as all metadata is already available.	*As all chatbots need to respond before a decision is made, answer time will be longer.* *Processing in the IL can delay response (depending on tokens in query and decision mechanism deployed).*

		Approach A: Use Primary Knowledge Directory (PKD)	Approach B: Use Interoperability Layer (IL)
	Semi	Fastest response time when host bot can answer the user query directly as no PKD is used as intermediate step. For queries forwarded to PKD same speed as Centralized.	Fastest response time when host bot can answer the user query directly as no IL is used as intermediate step. *When host bot cannot answer the user query, answer time will be longer as all chatbots need to respond before a decision is made. Processing in the IL can delay response (depending on tokens in query and decision mechanism deployed).*
	Decent.	Fast response time as all metadata is already available.	*As all chatbots needs to respond before a decision can be made, the response time will be longer. Processing in the IL can delay response (depending on tokens in query and decision mechanism deployed).*
Disruption handling	Cent.	*If the PKD encounters an issue or becomes unavailable, it can disrupt all the chatbot services.*	*If the IL encounters an issue or becomes unavailable, it can disrupt the interoperability as access to all the chatbot services is interrupted.*
	Semi.	As most queries will be answered by host bot, disruptions in the PKD will have less impact than a centralized system.	As most queries will be answered by host bot, disruption in the IL will have a smaller impact than a centralized system as the host bot can mitigate some disruptions if the IL encounters an issue.
	Decent.	For de-centralized PKD's the disruption has a smaller impact as other bots have their own PKD.	For de-centralized IL the disruption has a smaller impact as other bots have their own IL.
Setup	Cent.	Institutions would have to share information to build the PKD. This includes alignment on the level of information shared, the structure for the PKD and will require workload for each institution. *All institutions would need to conform a to central setup and make decision on the maintenance and cost sharing.*	Institutions need to align on configuration of the central IL (how the decision mechanism will be triggered & maintained) Easier to setup and manage a single IL controlling connections.
	Semi.	Same as centralized.	Same as centralized.
	Decent.	*Institutions would have to share information to build the PKD. Moreover, they would need to conform to a central setup and make decisions on the maintenance and cost sharing. Coordinating the individual PKDs across multiple institution could make the setup more complex and time consuming.*	Institutions need to align on configuration of the central IL (how the decision mechanism will be triggered & maintained). Coordinating the individual ILs across multiple institution could make the setup more complex and time consuming.
Energy	Cent.	In a centralized system, there is only one instance of the PKD which reduces redundant processing and storage. It also means there is only a single lookup	In a centralized system, there is only one instance of the IL which reduces redundant processing. *The IL requires sending queries to multiple chatbots and processing their confidence score*

		Approach A: Use Primary Knowledge Directory (PKD)	Approach B: Use Interoperability Layer (IL)
		operation to determine which chatbot is most appropriate to respond.	*which consumes energy (depending on the size of the IL being stored, the energy levels for computation vs storage should be compared).*
	Semi.	There is only one instance of the PKD which reduces redundant processing and storage. Even less energy would be consumed if the host bot can answer most user queries as the PDK would not be called all the time.	There is only one instance of the IL which reduces redundant processing. *The IL requires sending queries to multiple chatbots and processing their confidence score which consumes energy.* However, this con is reduced as the host bot determines first if it can answer the query therefore avoiding going thought the IL process all the time.
	Decent.	*A de-centralized approach requires each chatbot to maintain its own PKD potentially increasing overall energy consumption and carbon footprint.*	*A decentralized Intelligent Layer (IL) would consume significantly more energy, as it distributes queries across multiple bots and maintains multiple instances of the IL.*

Interoperability among chatbots requires a well-defined approach and prior setup. It's a collective endeavour varying with the chatbot types employed. Intent-based chatbots operate on a clear rule-based framework while LLM-based ones leverage AI for human-like text responses. They can both be integrated into an interoperable network with the described techniques. Interoperable chatbots may need to share details such as purpose, conversation flow, and scenarios, connect with databases for expansive information, utilize external APIs or services for specific tasks, or even hand off complex queries to human experts for enriched conversation experiences elements (more information in 1.4.2 Basic interoperability requirements). This section looked at the possible approaches to chatbot interoperability and their respective comparison. The next section will focus on the regulatory landscape and how it can impact the interconnected chatbot systems.

1.6 Regulatory outlook

1.6.1 A view on relevant EU regulations

The feasibility of interoperability between chatbots also relies on legal and regulatory context in which they operate and produce output. These regulations will define under which conditions can the chatbots connect with data in order to maintain a safe and compliant online environment for everyone involved, be it the different institutions or users. Regulations relevant to interoperability can be categories in three categories: **AI**, **privacy** and **data**.

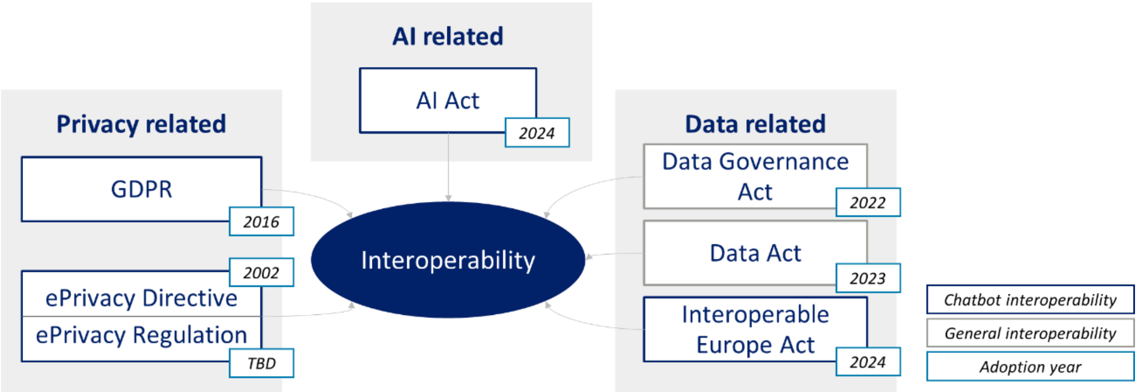


Figure 23. EU regulations impacting interoperability

The **AI Act**, initially proposed in 2021, has recently been adopted by the EU Parliament and the Council. The final version of the text[12] (dated May 2024) is expected to be enforced in 2024. With this assumption, most of the provisions will start to apply in June 2026, however, some requirements will be applicable earlier. The exceptions to this are that prohibited risk systems will need to comply within 6 months and GPAI (General Purpose AI) within 12 months. The AI Act is based on a risk-based approach of AI systems and considers the risks and pace of technological advancements of certain AI technologies.

For privacy related regulations[13], there are the **GDPR** (General Data Protection Regulation)[14] and the **e-privacy Directive**[15]. The GDPR is an EU regulation on information privacy to protect and empower EU citizens' privacy and how organizations approach data privacy (European Parliament & Council of European Union, 2016). Additionally, Regulation (EU) 2018/1725[16] outlines rules specifically for the processing of personal data by and between Union institutions and bodies. On the other hand, the e-Privacy Directive concerns the processing of personal data and regulates the sending of spam and cookies (European Parliament & Council of the European Union, 2002).

As for data regulations, there are the **Data Governance Act**[17] and **Data Act**[18]**,** designed to regulate the data sharing and ensure transparency for access and use of data (these refer more to interoperability in general systems and can apply to chatbot interoperability as well in specific cases). The **Interoperable EU Act**[19] aims to provide guidelines and conditions to enable interoperability and exchange of data between the different administrations, citizens and businesses (European Commission, 2022a). Together, these regulations can influence how chatbots could interoperate with one another, while ensuring transparency, protection of user privacy and data safety.

---

[12] AI Act : Regulation - EU - 2024/1689 - EN - EUR-Lex (europa.eu)
[13] There exists the eIDAS2 which has developed interoperability of national electronic identification schemes across Member States. While not directly relevant for interoperability, more information about it can be found here.
[14] GDPR : Regulation - 2016/679 - EN - gdpr - EUR-Lex (europa.eu)
[15] E-Privacy Directive : Directive - 2002/58 - EN - eprivacy directive - EUR-Lex (europa.eu)
[16] Regulation (EU) 2018/1725 : Regulation - 2018/1725 - EN - EUR-Lex (europa.eu)
[17] Data Governance Act : Regulation - 2022/868 - EN - EUR-Lex (europa.eu)
[18] Data Act : Regulation - EU - 2023/2854 - EN - EUR-Lex (europa.eu)
[19] Interoperable EU Act : Regulation - EU - 2024/903 - EN - EUR-Lex (europa.eu)

1.6.2 Foreseen impact of these Acts on interoperability

AI Act

The EU AI Act (European Parliament & Council of the European Union, 2024) is a regulatory framework for AI technologies. By understanding where your technology aligns with this regulation, you need to ensure full compliance as it will become applicable in 2025 and non-compliance could lead to substantial fines. For chatbot interoperability specifically, high-risk, transparency and GPAI obligations might apply. More information on its main principles can be found in Appendix B6.

High-risk and transparency obligations (Section 2, Chapter III & Article 50): Chatbots typically fall under limited risk category and adhere to transparency obligations specified in *Article 50*. It's compulsory to include disclaimers, inform users about system capabilities, risks, privacy, data use, and ensure they're aware they interact with an AI system. Users must also be notified if any content is artificially generated. In some circumstances, chatbots may be classified high-risk, leading to additional obligations found in section 2, Chapter III. For such high-risk AI systems, providers must establish a thorough risk management system throughout the entire lifecycle, considering interoperability aspects. Additionally, AI system providers generating synthetic content need to ensure outputs are marked as artificially generated or manipulated in a machine-readable format.

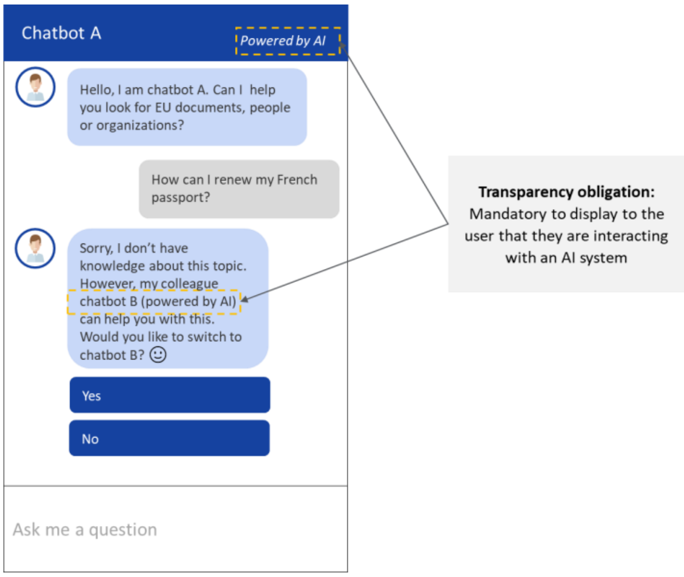


Figure 24. AI Act chatbot interoperability compliance example

It's also crucial to define each chatbot's role along the AI value chain. The AI Act distinguishes between providers, who develop AI systems, and deployers, who use them. Most obligations lie with providers, but deployers too must ensure providers comply with requisite obligations, potentially leading to additional responsibilities if high-risk chatbots are involved in the network.

GPAI (Article 53): Under *Article 53* of the AI Act, LLM-based chatbots may need to adhere to additional GPAI obligations. This applies if a chatbot is responsible for providing an LLM. The obligations include maintaining up-to-date technical documentation and making it available to downstream providers. The responsibility for these obligations rests solely with the GPAI model creators. If a chatbot utilizes a GPAI model not developed in-house, it isn't bound by the obligations in Article 53. Although important to know, these obligations primarily pertain to developers of LLM-based chatbots and not interoperability as they focus on standalone bot systems rather than systems sharing.

GDPR

The extensive use of personal data in today's technological environment calls for robust protection mechanisms, with GDPR emphasizing transparency. Information shared by a chatbot, including those in an interoperable network, must be concise, easily accessible, and understandable, ensuring users understand their personal data's collection and use. Chatbots should clearly state their data processing purposes, inform individuals about any risks, rules, safeguards, and rights related to data processing. The GDPR differentiates between data processors, joint data controllers, and data controllers, each with unique responsibilities. This is crucial as chatbots might act as joint controllers in an interoperable setting, but this depends on the type of companies involved in interoperability practices. The GDPR also requires the creation of independent supervisory authorities for enforcing compliance and mandates a data protection impact assessment for riskier data processing operations.

For interoperable chatbots, GDPR ensures free data flow across member states and standardizes rights and responsibilities for data controllers and processors, facilitating efficient data exchange. This framework ensures equal privacy protection within the EU, preventing restriction or prohibition of data movement due to personal data protection, which is crucial for chatbots operating across different regions. The EU's GDPR is key to protecting citizen privacy and personal data. Important GDPR concepts impacting chatbot interoperability include:

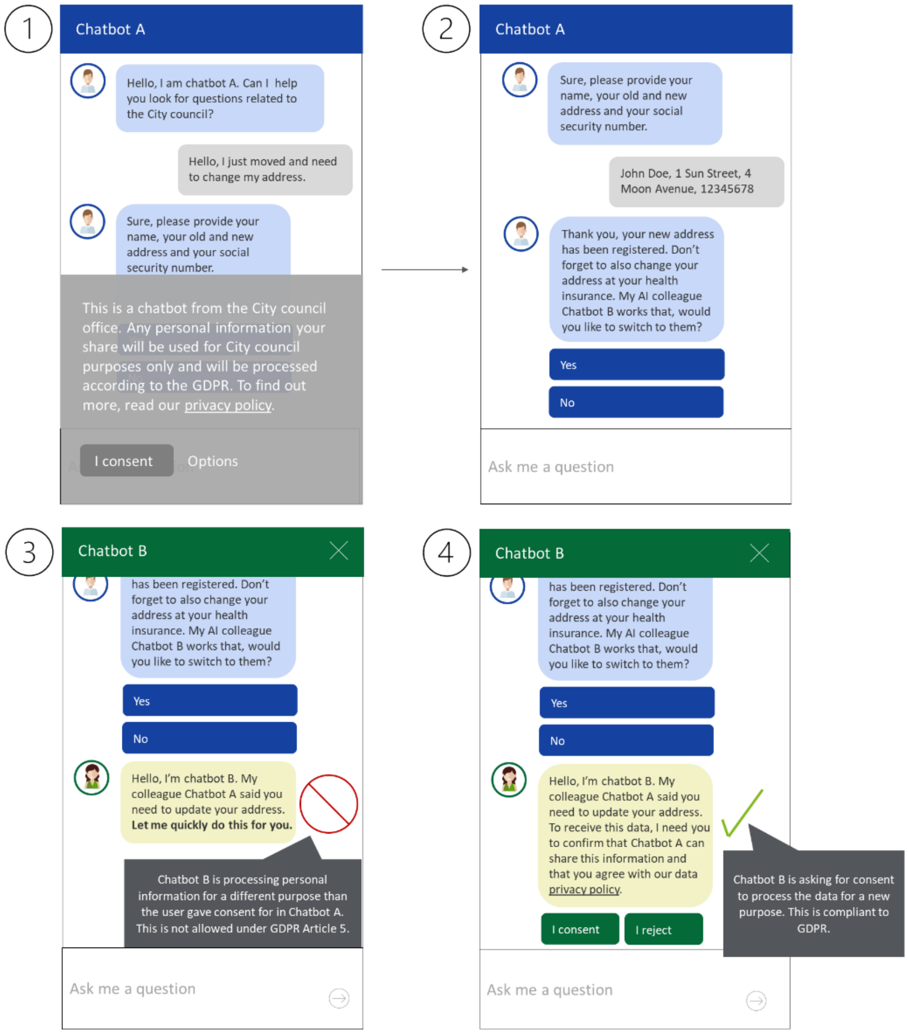


Figure 25. Example of interoperability and reusing data for another purpose

Article 5 - Principles relating to processing of personal data. Some key principles to consider include:

Lawfulness, Fairness & Transparency: Data processing must be lawful, fair, and transparent.

Purpose Limitation & Data Minimization: Data collection should be specific, explicit, legitimate, and limited to necessary processing.

Accuracy & Storage Limitation: Data must be accurate, up-to-date and stored for a limited period.

Integrity and Confidentiality: Data should be securely processed, protected from unauthorized access, accidental loss, or damage.

For interoperable chatbots, understanding data sharing nuances and applying anonymization techniques for conversation data is vital.

Article 7 - Conditions for Consent. These articles delve into the aspect of consent, typically:

- **Demonstration & Distinct Consent:** Controllers must prove data subject consent and consent requests should be clear & distinct.
- **Withdrawal of Consent:** Subjects can withdraw consent anytime without affecting prior legal data processing.
- **Assessment of 'Freely Given' Consent:** The service delivery shouldn't depend on the subject's consent to process unnecessary personal data.

A visual example of what this could look like can be found in Appendix B7.

Article 12 - Transparent Information & Communication. This article looks into the transparency of information, communication and modalities for the exercise of the rights of the data subject. What stands out is:

- **Clear Information & Communication:** Controllers must present data processing information in an accessible, easy-to-understand, transparent manner.
- **Information Suitable for Children:** Direct child-addressing information should be specifically tailored to children's understanding.

- **Standardized Icons:** Icons enhancing process understanding can complement provided information and should be machine-readable in electronic format. Clear Information & Communication: Information on data processing should be concise, clear, transparent, and accessible.

In interoperability cases, users should be clearly informed about data sharing between chatbots.

For general data protection, users must know about data exchanges between chatbots. If no personal data is processed, even though GDPR isn't directly applicable, it's good practice for enhancing transparency and trust. Interoperability contracts should clearly define individual chatbot roles and responsibilities regarding data protection. Explanation of algorithmic decisions is recommended for user understanding, transparency in decision-making, and trust-building, despite debates around GDPR's scope on explainability.

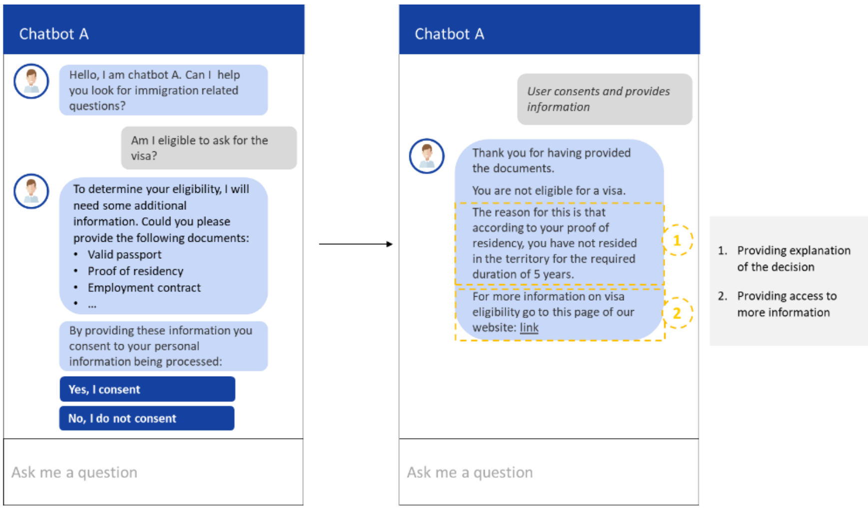


Figure 26. GDPR explainability consideration

ePrivacy Directive / Regulation (cookie law)

The ePrivacy Directive, a supplement to GDPR that addresses electronic communications, cookies, and digital marketing, plays a pivotal role in chatbot interoperability. The directive classifies cookies into four categories: strictly necessary, preferences, statistics, and marketing. Chatbots, considered non-essential website features, fall under preference cookies, requiring active user consent. Obtaining this can be challenging as active consent rates are typically low (<0.1%) (Utz, Degeling, Fahl, Schaub, & Holz, 2019). To enhance user consent for chatbots, two potential strategies are:

- **Option A - Accept Cookies in Website:**
  1. <u>Explaining Functional Cookies:</u> Use a consent management tool to explain why each cookie type is necessary, encouraging consent for functional cookies required for chatbot availability.
  2. <u>Separate Chatbot Option:</u> Use a consent management tool to add a specific checkbox for chatbot functionality, allowing users to consent to chatbot-associated cookies without consenting to all other functional cookies.

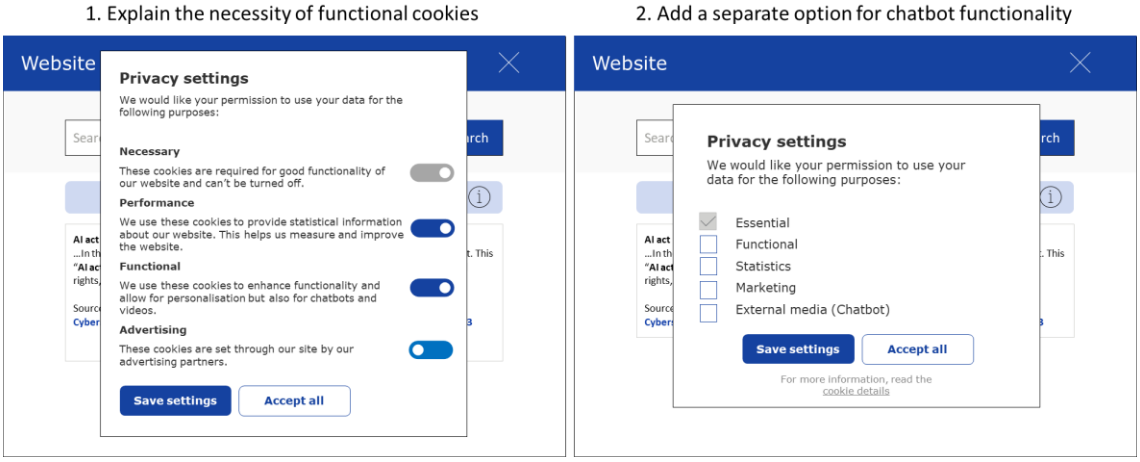


Figure 27. Option A - Accept cookies in the website

- **Option B - Accept Cookies in Chatbot:**
  1. <u>Landing Page for Chat Opt-In:</u> Create a dedicated landing page explaining the necessity of functional cookies for chatbots, providing users an opportunity to consent.
  2. <u>In-Chat Disclaimer:</u> Allow chatbot windows to pop up but restrict user's capability to type until they consent to the necessary cookies, providing brief explanations for cookie requirements and offering users another consent opportunity.

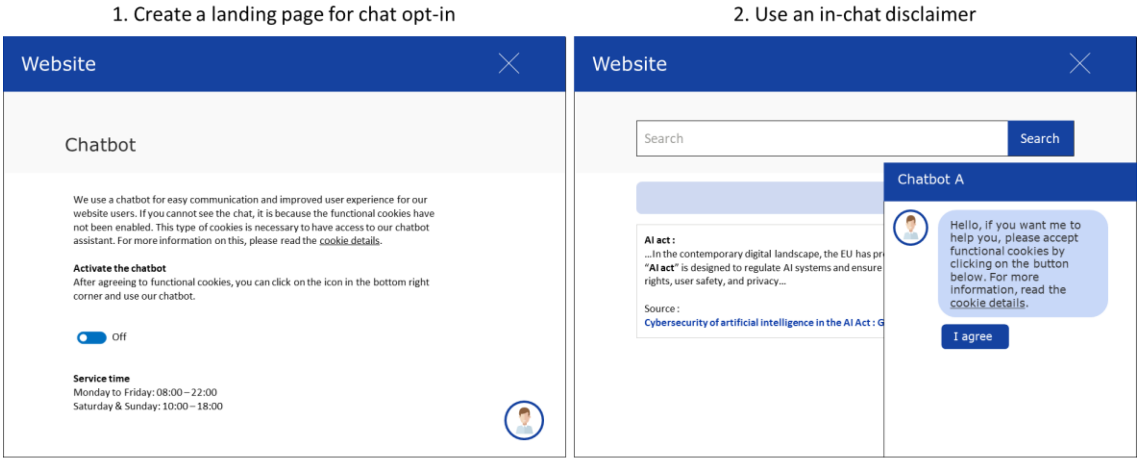


Figure 28. Option B - Accept cookies in the chatbot.

Chatbots use cookies or similar technologies to remember user interactions and customize service, falling under 'preferences/functional' cookies. These enable user conversation and input tracking, improving user experience. In an interoperable network, the initial consent should cover contributing bots, or a similar one be sought. Consent is typically obtained through a notice instructing users to agree to cookies for their initial and continuous conversation tracking, complying with set directives.

Data Governance Act & Data Act

The European Commission's Data Economy Strategy introduces a single market for data aiming for ethical and responsible data usage. Key components include the Data Governance Act and the Data Act.

The Data Governance Act aims to build a trustworthy data sharing system cross-sectors and Member States. It involves rules for reusing publicly available data, data intermediation services for data sharing processes, tools for data altruism allowing voluntary data sharing, the European Data Innovation Board for oversight, and establishing trust in international data flows (European Commission, 2024a).

The Data Act regulates data usage by different entities, requiring data holders to make data available under reasonable and transparent terms (European Commission, 2022b). It ensures data obtained through connected products or related services is accessible, with specific rules ensuring equal treatments for large companies and SMEs. It asserts that any data sharing agreement must not put product security at risk (European Commission, 2024b).

While these acts facilitate data sharing and interoperability, their relevance to chatbot interoperability is limited due to minimal user data processing involved. In chatbot interoperability, the focus is more on technical standardizations concerning message structure and API compatibility rather than individual user data.

Interoperable Europe Act

The Interoperable Europe Act aims to harmonize interoperability and data exchange standards across Europe for easier information exchange and a unified digital single market. It plays a critical role to reach Europe's Digital Targets – making all key public services accessible online by 2030. It covers various sectors, including chatbots, and mainly involves:

- **EU Cooperation:** Establishing a cooperative framework to outline a common interoperability agenda.
- **Mandatory Assessment:** Ensuring public services are "interoperable by design" through scheduled assessments.
- **Interoperable Europe Portal:** Setting up a one-stop-shop for sharing and reusing solutions among public administrations.
- **Innovation and Policy Support:** Offering training and regulatory sandboxes for solution development and scaling.

Some aspects of the Interoperable Europe Act should be considered specifically for chatbot interoperability:

- **Interoperability Assessment (Article 3):** Public sector entities need a mandatory interoperability assessment for network alterations that enable electronic delivery or management of public services. This assessment needs a description of the planned operation, its impact, alignment with the European Interoperability Framework, information on used APIs, and relevant data for cross-border exchange.
- **Interoperable Europe Portal (Article 8):** This single-access-point portal provides information about cross-border network interoperability and aids organizations in adopting interoperable solutions. The portal could help share chatbot interoperability projects and provide a registry of chatbots.
- **Enhanced Governance (Chapter 4):** Improved governance could lead to uniform guidelines for chatbot design, programming, and integration into broader public service networks. The establishment of the 'Interoperable Europe Board' could foster interoperability standard agreement, aiding chatbot integration across the EU.

1.7 Implementation framework

This chapter proposes the structure to create an interoperability PoC end-to-end. Following the proposed phases and using these templates would support development of interoperability according to the current best practices and guidelines as captured in previous phases of this study. A table can be found in AppendixB8 B8 which presents the anticipated deliverables to be generated over an interoperability between chatbot project. Each deliverable outlines the scope and content it should include, these remain examples and can be enhanced as the project evolves.

The framework below aims to enhance interoperability, by guiding public institutions in a structured method to connect their chatbots in order to overcome challenges and move towards a more interoperable network of chatbots within Europe. For each phase, the intended goal is described as well as key prerequisites and the deliverables that should be considered. Where feasible, general guidelines for consideration and templates or checklists are included to support an interoperability PoC from design, through testing to implementation.

	Phases			
	A. Initiation phase	B. PoC Development	C. Testing	D. Deployment & Monitoring
Activities	• A1. Define project goals • A2. Describe interoperable network • A3. Define & align on governance • A4. Define functional design of interoperable network • A5. Define technical structure • A6. Prepare testing • A7. Finalize contractual alignments • A8. Consolidate project discovery	• B1. Set-up interoperable architecture • B2. Build interoperability structure • B3. Complete PoC	• C1. Conduct testing • C2. Validate testing (Iterate → B)	• D1. Define deployment pipeline • D2. Deploy to production • D3. Manage & monitor release
Description	Preliminary preparation to understand the as-is and key considerations to take into account	Testing the feasibility of the interoperability by building the PoC	Test the PoC solution, document, validate the testing results and reiterate on the PoC.	Setting up the solution and tracking its functionality to ensure correct operation
Pre-requisites	• Objectives of interoperability • Chatbots list	• Chatbots selection & assessments • Feature selection • RACI Matrix of governance • Technical architecture plan • Project timeline & milestones	• Working interoperable PoC	• Regulatory compliance • Working interoperable PoC • Tests validations • Access permissions • Contingency plan
Key deliverables	D1.1 Functional design plan D1.2 Technical architecture plan D1.3 Testing plan D1.4 Contractual documentation D1.5 Project plan	D2.1 Working system	D3.1 Sprint reports & test logs D3.2 Sign off completion certificate	D4.1 Deployment pipeline D4.2 Live interoperable chatbots D4.3 Monitoring reports

Figure 29. Four phases to implement a chatbot interoperability PoC

1.7.1 Phase A: Initiation

A. Initiation phase | B. PoC Development | C. Testing | D. Deployment & Monitoring

Goal: The goal of the initiation phase is for all parties who want to interoperate to align on all the necessary components. This phase includes all aspects of project discovery to determine not only which chatbots will be integrated, how the integration will be facilitated, which technologies will be used, etc. but also what the contractual aspects will be that govern this interoperable network and which roles and responsibilities will fall on which actors.

Activities: The following key activities are proposed to cover all aspects of the project initiation phase.

Table 9. Phase A: Activities descriptions

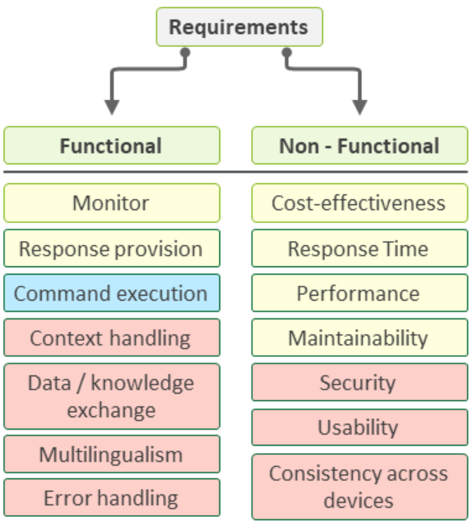
Activity + Description	Deliverables	Related Links
A1. Define project goals Consider the following aspects: • **Select relevant bots to be included in the network:** Conduct extensive research on the available bots, their capabilities, and their compatibility • **Define user base:** Comprehend citizens and businesses requirements and expectations from the chatbots to ensure that the services provided meet user needs. • **List augmentation benefits**: List how it can make your services more efficient.	Key features list (part of D1.1: Functional design plan)	Section 1.3.2
A2. Describe interoperable network • **Define the conversational flow:** Define the conversational flow of your chatbot including purpose, scenario, elements. • **Identify the conversational requirements for the chatbot**: Define the conversation scenarios and the dialogues required to handle them. Select the conversational flow to implement as seen in 1.4.2.2. • **Design conversations taking into account user intent and context**: Create conversation flows that understand and adapt to each user's unique needs. For more details on intents refer to Section 1.5.1.1, Figure 54.	Key features list (part of D1.1: Functional design plan)	Sections 1.4.3.3, 1.4.2.2, 1.4.2.4, 1.5.1.1

A3. Define and align on governance
• **Define and align the governance:** Identify and allocate roles and responsibilities, clarify expectations, and align all team on the governance structure.
Approaches to consider in the governance
• **Centralized**: In a centralized approach, governance would typically fall under a central group comprised of members from each institution involved. This central group would be responsible for oversight, decision-making, and policy-setting for the centralized PKD or IL. When it comes to maintenance, there are a couple of options:
  o IT Team Formation: Institutions could collectively decide to form an "IT Team" by pooling members from each institution. This team would be tasked with the maintenance, security, and updates of the PKD or IL.
  o Single Institution Responsibility: Alternatively, if the project was initiated by one particular institution, that institution could be designated as responsible for the maintenance and upkeep of the PKD or IL. This would include tasks such as ensuring data quality, security, and the implementation of updates.
• **Decentralized:** In a decentralized approach, each institution is responsible for maintaining its own PKD or IL. This includes ensuring the quality, security, and relevance of the information and protocols within their respective directories or layers. However, to ensure seamless interoperability and consistency, a cross-institution collaborative group can be established. This group would:
  o Facilitate Regular Communication: Institutions would regularly inform each other about potential issues, updates, or changes within their chatbots.
	RACI Matrix (D1.5: Project plan)	Section 1.5.1.2

Activity + Description	Deliverables	Related Links
◦ Collaborative Problem-Solving: When problems are identified, institutions can work together to find solutions that can be uniformly applied. • Update Synchronization: By sharing updates made to their chatbots, institutions can ensure that other institutions can also update their PKD or IL accordingly, maintaining the consistency and efficacy of the interoperable network.		
A4. Define functional design of interoperable network • **Define functional and non-functional requirements:** Functional requirements of an interoperability system guide its operational features, while non-functional requirements relate to system quality, ensuring efficiency, maintainability, and reliability. These requirements should be prioritized based on their importance. • **Define mechanism for sharing domain of knowledge:** Select which approach (PDK or IL) will be used to share domains of knowledge in the network. • **Identify and select suitable vendors (if required)**: In case the Bots are not all readily built and in production, some vendors might need to research and evaluate different vendors based on various criteria. • **Design interface**: Align UX/UI elements with the interoperable network. **Select redirection:** Select your redirection options for chatbot interoperability *(e.g., see **Error! Reference source not found.**)*	D1.1: Functional design plan	Sections 1.4.3, 1.5.1
A5. Define technical structures • **Define the needed architecture and technology stack**: Identify the required technology components (PKD or IL: see Section 1.5.1.1) to support chatbot interoperability keeping the requirements set-up in the previous activity in mind. • **Analyze and define the data structure for interoperability**: Establish the format and structure of data for effective information transfer that has been mentioned in section 1.4.2.1 Dataset considerations. This involves determining how different chatbots will interpret and understand the data exchanged among them, which concern the backend (with discrepancies, an intermediary layer may be needed). **Set up the host solution and manage the hosting environment settings**: Choose a reliable host *(e.g., AWS, Azure)*, and configure the environment.	D1.2: Technical architecture plan	Sections 1.5.1, 1.5.2
A6. Prepare testing • **Prioritize requirements:** Determine requirements' significance and their testing sequence, usually based on business value, risk, complexity, and impact. • **Define epics:** Define large units of work called Epics which can be disaggregated into smaller tasks based on the prioritized pre-defined requirements. • **Define User Personas:** Define fictional characters representing actual users and their behaviour, used for guiding design or test decisions. • **Draft User Stories:** Narratives illustrating users' perspective of interacting with the product, created based on the user personas. **Validate User Stories:** User stories are approved and signed off by stakeholders.	D1.3: Testing plan (incl. prioritization, epics, user stories, etc.)	Template B.9.2

Activity + Description	Deliverables	Related Links
A7. Finalize contractual alignments • **Draft necessary contracts:** Contract considerations - Project contract, NDA, SLA, Data Processing Agreement, Protection & Privacy, IPR, Copyright **Comply to regulations:** Legal compliance - Make sure you are compliant with existing laws, such as the AI Act and GDPR	D1.4: Contractual documentation	Sections 1.4.2.3, 1.6
A8. Consolidate project discovery • **Draft the project plan:** Timeline for the PoC development, the milestones, foreseen meetings and stakeholders involved. • **Set acceptance criteria and Definition of Done (DoD):** The acceptance criteria are used to determine if the PoC meets the desired goals and requirements. DoD defines criteria to meet for the PoC to be considered complete *(e.g., chatbots to provide answers in both English and a low-resource language, UAT completion).*	D1.5: Project plan Acceptance criteria and DoD (part of D1.3: Testing plan)	

Templates to accelerate phase A



When investigating interoperability between chatbots, it is crucial to understand and identify the functional and non-functional requirements that will serve as foundations in the development, deployment and maintenance of the system. Many of these will feed to UX/UI considerations or how domains of knowledge are shared (topics covered in sections 1.4.3 and 1.5.1). Below are some examples of functional and non-functional requirements. Additional requirements should be added and adapted to project goals[20] (more detailed requirements' description available in Appendix B.9.1)

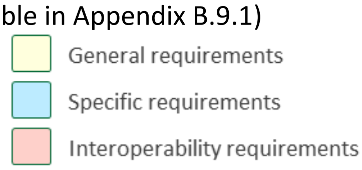


Figure 30. Example of functional & non-functional requirements

It is important to understand the key technical components that will make the interaction between chatbots possible. This section will highlight the major components that significantly contribute to the design, development and operation of interoperable chatbot systems and the elements that are common or are specific to each chatbot. For this the following technical architecture provides a blueprint for the possible system construction, this includes the backend components that handle data management to the interaction processing.

[20] Not all will be relevant, and the interoperability case might require additional ones

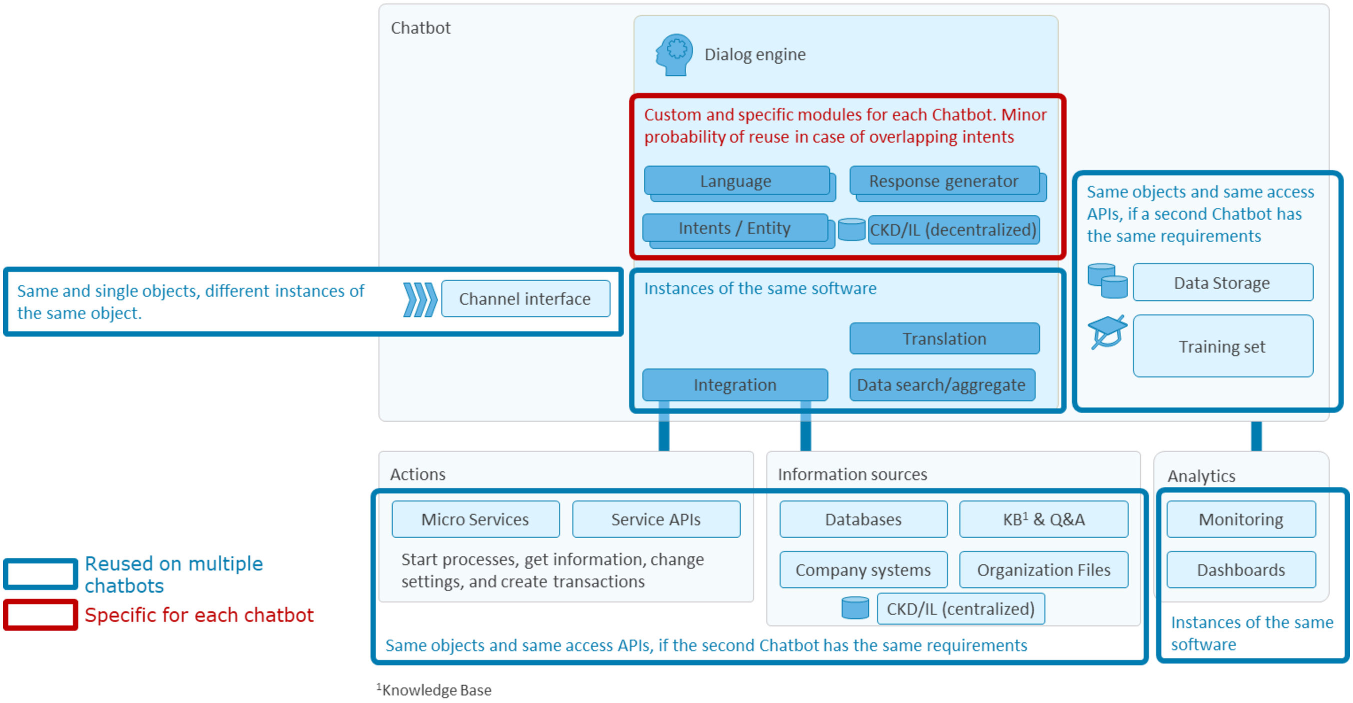


Figure 31. Architecture components: reused and customized components for each Chatbot

The components from the Figure 31 above can be mirrored in the information flow illustrated below, leveraging the example of chatbots interoperability in the case of a centralized PKD (primary knowledge directory) approach.

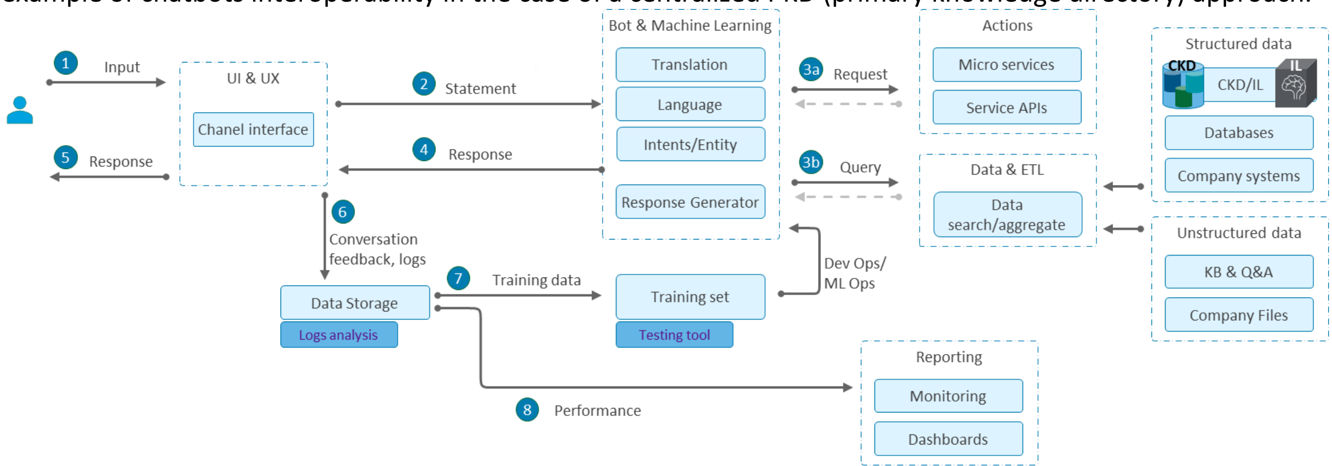


Figure 32. Information flow: PKD/ IL approach example

Regarding the testing preparation, User Acceptance Testing, also known as UAT or End-User Testing, is the last phase of the software testing process. During UAT, the software is tested by the real users who will be using the software in the real-world environment. UAT is important because it helps in validating that the system is ready for release. It confirms that the system meets the agreed business requirements and can handle tasks in real-world scenarios according to the specifications. The main purpose of User Acceptance Testing is to ensure that the software system is working as expected before it's moved into the live environment. If any issues or improvement areas are identified, it gives the development team an opportunity to resolve them based on the user's feedback. Figure 33 shows an overview of the UAT testing which will be explained in more details in the next sections.

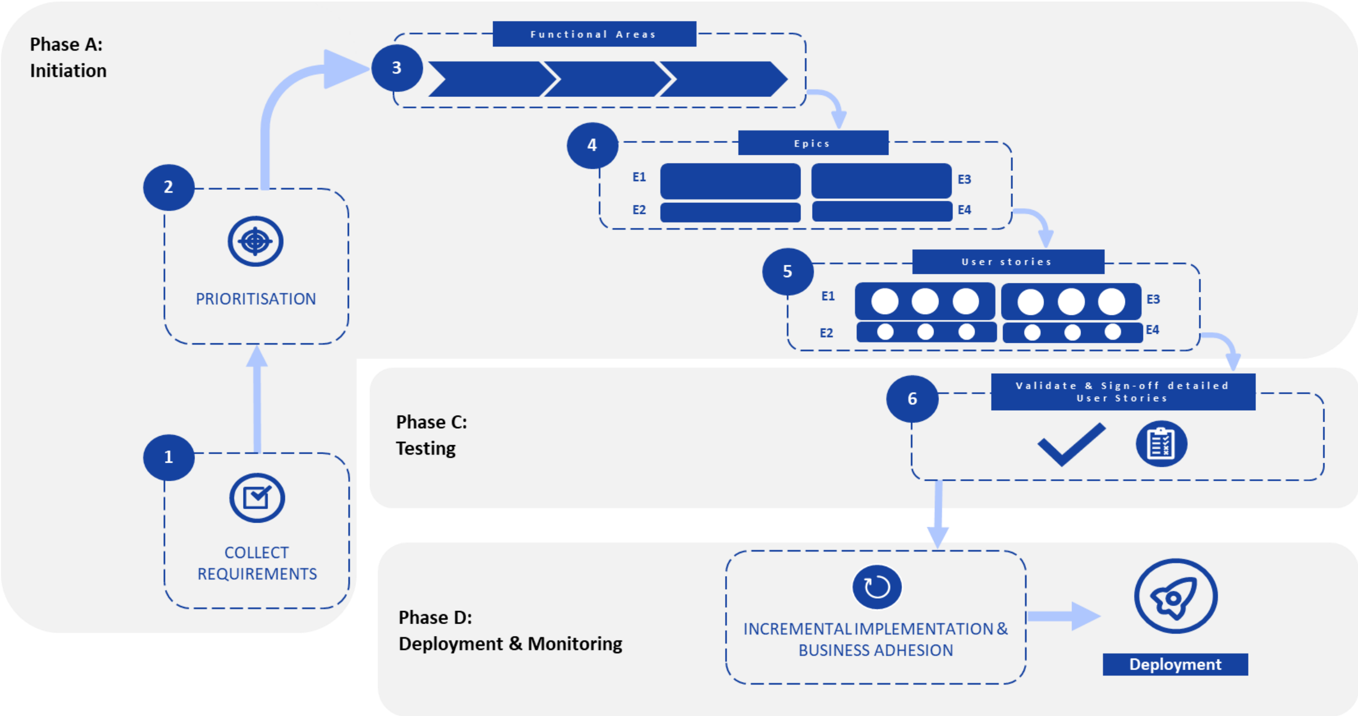


Figure 33. Overview of UAT testing process

A detailed description of steps 4, 5, 6 (Epics, User Stories, Validate & Sign-off detailed User Stories) is available in the Appendix B.9.2.

1.7.2 Phase B: PoC Development

A. Initiation phase | **B. PoC Development** | C. Testing | D. Deployment & Monitoring

The Proof of Concept (PoC) development aims to showcase functionality and assess the practicability of chatbot interoperability before building a complete system. It helps anticipate potential issues and establish expectations for the final product. The upcoming section will detail an agile implementation for the PoC development, covering setup of interoperable architecture, conversation element design, infrastructure building, and documentation. The PoC development will be an iterative process adapted based on testing phase outcomes.

Table 10. Phase B: Activities description

Activity + Description	Deliverables	Related Links
B1. Set up interoperable architecture • **Set up the development environment**: Establish a chatbot development environment that aligns with the latest bot version and mirrors the production environment. • **List and get all necessary components and accesses**: Assemble and secure all necessary components and access permissions for stakeholders to begin bot development, including network and tracking.	n/a	Sections 0**Error! Reference source not found.** 1.4.2.2 1.5.1.1 Phase A: 1.7.1 Phase D: 1.7.4

Activity + Description	Deliverables	Related Links
B2. Build interoperability structure **Backend** • **Develop approach selected:** Configure and connect interoperability systems per the Phase A-selected PKD or IL centralized/decentralized approach. Ensure the architecture setup is correct and bots can receive intents by performing basic functionality tests and network connectivity reviews. Legal permissions for accessing bot intents will depend on the chosen approach. • **Set interoperable confidence score (threshold):** if the confidence threshold is reached, the chatbot with the highest score will respond to the user query, this is linked to the trigger mechanism (see B10 and Figure 59). • **Implement security measure:** Implement encryption protocol for exchange of data between the chatbots (HTTPS). • **Program the translation layer between different chatbot platforms:** Develop a system that translates requests and responses. Different approach can be considered for translation that can depend on the languages of the chatbots or the chatbot type (e.g., pivot languages, human translation, machine translation, NLP), see section 1.4.2.4 Language considerations. **Frontend** • **Develop specific trigger mechanisms:** Based on the design and defined conversational flows, develop the trigger mechanism and conditions based on user interactions (see Appendix B10). • **Implement the logic for routing the interactions:** Implement the conversational flow elements paths along which requests and responses flow between different systems. This is like defining the rules around when and how messages get passed between both chatbots as seen in 1.4.3.1. The rules could be based on specific user needs, keywords, or other criteria. • **Implement UX/UI elements:** Develop and fine tune iteratively the UX/UI elements predefined in phase A (e.g., chatbot interface elements such as changes of colours when interoperability happens, feedback buttons, etc.)	n/a	Sections 0 1.4.2.1 1.4.2.2 1.4.2.4 1.4.3.1 1.5.1.1 Phase A: 1.7.1 Phase D: 1.7.4 Appendix B10
B3. Complete PoC • **Refine PoC:** Iteratively optimize and refine the PoC with the test phases (see phase C) and fix discrepancies identified in testing. • **Validate PoC:** Make sure acceptance criteria and DoD have been reached (e.g., defined response time).	D2.1 Working PoC system	Section Phase C: 1.7.3

This section ventured through the PoC development, reviewing the key stages necessary to bring the concept into reality. These insights derived will be instrumental in guiding the direction of subsequent Testing and Development sprints. Concluding with Phase B, the Proof of Concept (PoC) Development, and transitioning into Phase C, Testing. These two phases move in lockstep, each influencing and shaping the other in a continuous feedback loop (sprint). The results from the Testing Phase directly feed into the next sprint of the PoC Development, allowing for precise and focused refinements. Conversely, the newly advanced output of the PoC Development is subjected to evaluation in the Testing Phase. It is through this iterative process that we take steady strides towards the realization of the PoC.

1.7.3 Phase C: Testing

A. Initiation phase › B. PoC Development › **C. Testing** › D. Deployment & Monitoring

This section deals with the evaluation of interoperable chatbots. The goal of this phase is to evaluate the functionality, reliability and efficiency of the interoperability between the chosen chatbots. The table below goes into further details for each phase. For templates on how to accelerate phase C, consult Appendix B11.

Table 11. Phase C: Activities descriptions

Activity + Description	Deliverables	Related Links
C1. Conduct testing • **Perform tests:** Test the predefined test cases prepared in A6 to ensure the PoC does not have defects or unexpected behaviours. The test cases can be categorized as "met/not met/potential to meet". • **Log defects:** Document the defects seen in the testing, it is important to keep track and reproduce the defect to fix it. • Prioritize: Based on the priority matrix of the requirements, the defects will be assigned a scope and score to define the priority of the defect (low to high). They can also be prioritised for sprint inclusion (e.g., Defect ID 2 – low priority - will be looked at in sprint 2) • **Fix:** Based on the reproduced errors and priority level, find and solve the defects. Confirm fix by reproducing the same defect with the same conditions. • **Scope of next sprints & retest:** The testing and PoC development are iterative and will take place in several sprints. The defects fixed as well as other requirements to build will be part of the next sprint to develop in the PoC and test again.	D3.1 Sprint reports & test logs	Phase A: 1.7.1
C2. Validate testing • **Deliver and Iterate UAT:** The UAT process is reiterated until all requirements are covered and no more issues (moderate and high) are discovered with the PoC before the final delivery and the "Definition of Done" has been reached. • **Validate:** Validate testing by relevant stakeholders.	D3.2 Sign off completion certificate	Phase B: 1.7.2

Concluding the testing phase, this approach ensures that the interoperable network meets the business needs and works as expected in real-world scenarios. It is iterative, responsive, and user-focused making it a key element in successful PoC development. After the successful completion of testing, the next phase focuses on integration into the user environment with continuous tracking.

1.7.4 Phase D: Deployment & Monitoring

A. Initiation phase › B. PoC Development › C. Testing › **D. Deployment & Monitoring**

The goal of the deployment and monitoring phase is to ensure the smooth integration and interaction of the chatbots within the user environment, while continuously supervising the system to confirm that the solutions is working well. This phase also aims to use the monitoring insights to optimize and improve the interaction between the chatbots based on user feedback and system observations.

Table 12. Phase D: Activities descriptions

Activity + Description	Deliverables	Related Links
D1. Define deployment pipeline • **Define necessary deployment components:** Key components are necessary to set up the deployment pipeline such as Version Control System (VCS), build server, automated testing, artifact repository, deployment automation. • **Commit stage:** Trigger commit code to the VCS. This will allow code changes to be fetched from VCS and for the build server to compile and run the code. • **Automate testing:** Execute various test to ensure code functionality, and reliability (e.g., unit tests, integration tests). • **Set-up and deploy to staging testing:** Once all testing have been validated, create a staging environment that will look like the production environment. Deploy the solution there, this will allow to test under real-world conditions providing a final check before the production. • **Implement stability or stress testing:** Before launching, the interoperable chatbot system must be thoroughly stability tested for handling varied real-world scenarios, peak loads, high data volume, and sustainable performance, preventing crashes, slow responses or data loss when used by consumers. • **Validate with stakeholders:** Get final approval from stakeholders to move to the production deployment.	D4.1 Deployment pipeline	n/a
D2. Deploy to production • **Define production deployment strategy:** Define and select deployment strategy such as canary releases, blue-green deployments. • **Deploy to production environment:** Trigger deployment to production environment.	D4.2 Live interoperability chatbot	n/a
D3. Manage & monitor release • **Define KPIs:** List measurable KPIs that tie in with business objectives and customer expectations. Relevant KPIs could be response time, user satisfaction rate, accuracy of information, number of successful interactions, or problem resolution rate. • **Provide continual assessment:** KPIs should be perpetually tracked and evaluated from the moment of deployment and throughout the system's lifecycle. This enables prompt detection of any potential issues or discrepancies and allows for timely rectification. • **Implement performance analysis:** Collect and analyse data on individual chatbot performance within the interoperable network. This helps to identify any weak links and optimize overall system performance. • **Optimize:** Apply insights derived from monitoring and analysis to continuously refine and optimize the interoperable chatbots, align closely with KPIs, and improve user experience.	D4.3 Monitoring reports	n/a

Monitoring and Key Performance Indicators (KPIs) play an integral role in maintaining the efficacy and efficiency of the interoperable chatbot system. They not only indicate the current performance level but also provide valuable insights highlighting areas for improvement. Regular monitoring backed by robust KPIs ensures constant system optimization, improved user satisfaction, and effective error handling. In a rapidly evolving AI landscape, it is this iterative cycle of monitoring and evaluation that ensures the chatbots remains effective, accurate, and user centric.

1.7.5 Example: PoC Implementation framework applied

A PoC was conducted to test the theory listed in the study and validate the feasibility of the proposed approach to interoperability between chatbots.

Project description

The project was launched in 2024 in connection with the Digital Europe Programme (DEP).
For this PoC on interoperability, two public institutions contributed with their bots:

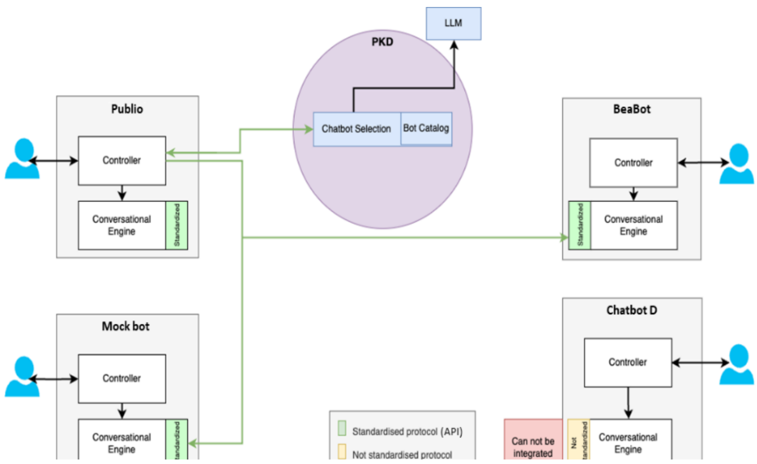
- **The Publications Office of the EU:** Publio – a bot helping search documents, people, organizations and legislation across the European Union
- **The Federal Public Service (FPS) BOSA**: BeaBot – provides assistance to users to gain information on Belgium public services (such as jobs vacancies, application procedures, etc.)

As Publio is available in English, French and Spanish and BeaBot in French and Dutch, French as the only common language was selected for the PoC.

PoC Approach - Interoperability setup and flow

The approach selected during this PoC was the semi-centralized **Primary Knowledge Directory (PKD)** (more information in *1.5.1 Feasible options to facilitate interoperability*). Key aspects considered for setup:

- **Intent & entity mapping:** This involves noting the shared intents and entities from the different chatbot systems and creating a mapping between them.
  Both Publio and BeaBot provide job and internship information among other functionalities. A mock bot was also created to simulate how a third bot (or an additional number of other similar bots) would impact the interoperability network. This bot replicated a German bot providing internship and other public service information to simulate additional scenarios and interactions and can be triggered by asking for Germany rather than Belgium.



The technical setup of the PKD included an LLM to process user queries, evaluating them against the intents and capabilities of all available chatbots to determine the most suitable one for responding (for accurate routing of user queries). For example, when trigger phrases included context on job, internship or the country “Belgium”, it would very likely be redirected to BeaBot. The PKD exposed a REST API with an endpoint to handle user queries and routing decisions.

Once the PKD selected the most appropriate chatbot, communication occurred directly between the host chatbot and the contributing chatbot. The PKD's role was limited to intent evaluation and routing, preventing unnecessary delays.

- **Dialog management:** This involves understanding how each chatbot generates responses based on identified intents and entities, manages conversation flows, and tracks context – in the context of interoperability. For the PoC, the conversational style where all conversations are facilitated within one bot's interface was selected (for additional information refer to '*Option C: Host bot hosts contributing bot in same interface*' in *1.4.3.1 Conversational flow elements).*
  Publio hosts the contributing bot, BeaBot from FPS BOSA within the same interface. The interoperability was unidirectional, meaning that only the host bot's interface was modified to include the contributing bot, while the contributing bot interface on their website remained unchanged. To incorporate BeaBot and differentiate it from Publio in the interface for clarity for users, front end modifications have been added such as icon, colour as seen in **Error! Reference source not found.**.
- **Flow termination**: The standard API needs to include a termination marker to indicate when control of the conversation should be returned to the host bot. In the PoC, this consisted of final and error messages, which required hardcoding a list of flow termination messages within Publio.

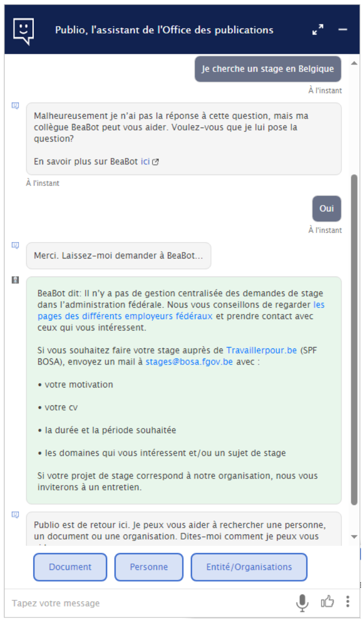


Figure 35 – PoC dialogue flow

Target audience

The PoC was designed to assist users (European citizens and other) to obtain relevant information regarding internship and job searches throughout the region. The target audience for this PoC was French speaking users.

Timeline

Overall PoC completion: Three months

Phase A - Initiation**:** 1 month (October 2024):

Workshops to design the solution

WS1: Define interoperable network (functional design)

WS2: PoC technical design overview workshop

WS3: Access setup (API requirements)

WS4: Testing phase **definition** & test plan

Phase B – PoC **Development &** Phase C - T**esting:** 2 months (November to December 2024):

PoC **solution development**

Two testing sprints completed

PoC acceptance

More information on the phases in 1.7 Implementation framework.

Testing

Two rounds of user testing have been conducted to validate the acceptance criteria defined at the start of the project. This testing aimed at having an initial insight on how the users interacted with the chatbots and validate the PoC based on acceptance criteria. Acceptance criteria included both functional and non-functional requirements, see Table 31 in the Appendix. These acceptance criteria have been validated through the two-testing using six defined user stories and nine test cases presenting detailed features to test.

Table 13. PoC development: User stories and test cases

ID	User story
US1	**Belgian citizen** looking for where to find the national identification number on the ID card
US2	**Student** looking for a potential internship in the public sector
US3	**International citizen** apply to job position
US4	**Candidate** wanting to become a civil servant
US5	**Candidate** looking for another job in the public sector in Belgium
US6	**Candidate** looking for regulatory document

ID	Test case	Description
TC1	Auto trigger for interoperability	• Trigger intent defined for each user story facilitates a switch from Publio to BeaBot • Specifically test if the correct Bot is called for the query or not • Assess the understandability of the flow ' BeaBot says: xyc',
TC2	Switch on demand	• User can ask to switch back to Publio flow at the end of each flow
TC3	Reach conversation end	• Interoperability moves till end of BeaBot flow
TC4	Unique media display	• Publio & BeaBot have similar functionalities (no emojis)
TC5	Error message	• Error intent gives message for Publio and for BeaBot if triggered
TC6	Interoperability trigger message	• The bot provides user with context that interoperability is available when first triggered
TC7	Consent message	• When a switch is triggered, user is notified and provides consent to interact with the proposed bot • User says Yes, trigger switch • User says No, go back to Publio
TC8	Interaction tracked	Extract of count of switches between bots successfully triggered
TC9	Multiple option buttons	Suggestion need to be selected by user to continue conversation

Results

Challenges

Some key challenges encountered during PoC:

- **Setup of PoC environment**: Had to duplicate BeaBot in a sandbox environment for the PoC (e.g., not all intents have been included for BeaBot).
- **Defining integration points in conversation flows:** Ensure information is not repeated. Finding right entry point in interoperability flow after trigger (e.g., not having BeaBot suggested options without context).
- **Definition of redirection threshold**: Adjust PKD threshold to all bots in interoperability network is used at best knowledge level.
- **Page related information**: If a contributing chatbot includes information that requires to be on their specific website, the interoperability should account for this to avoid user confusion.
- **Dealing with media within text**: Considering Publio had different interface sizes where BeaBot's picture support needed adjustments on the resizing in different environments.

Key findings

The following critical success factors contributed to the achievement of the PoC:

- **Sufficient project sponsor support**: All entities involved had a buy-in into PoC goals at senior organizational level.
- **Defined governance & project management**: Frequent touchpoints and communication between all parties with reactive teams with well-defined roles, responsibilities and timeline.
- **Quick setup of environments**: Access provision, API updates and sharing of bot documentation. No delays to start the PoC.
- **Technical support**: This included having a dedicated team focused solely on the development of the PoC.

The interoperability between two public chatbots was successfully achieved on the Publio interface in under three months, a testament to the seamless cooperation between both institutions. Publio & BeaBot could interoperate and communicate collaboratively on a unified interface.

1.8 Conclusion

The exploration of interoperability between public administration chatbots promises to bring forth a spectrum of advancements that will significantly enhance public service delivery, create a collaborative network, and increase user adoption.

Key findings

Our research indicates that numerous EU programs and initiatives are advancing the development of interoperability. However, no universal interoperability standards have emerged thus far. The study discovered that several organizational interoperability projects are currently underway across Europe, with similar aims and multiple common challenges. These challenges include the previously mentioned absence of standards, as well as the lack of a centralized registry of existing chatbots, which leads to missed opportunities among public organizations. Additionally, issues related to multilingualism and the use of translation tools between chatbots were frequently highlighted.

Moreover, several key considerations to take into account for interoperability is detailed, such as considering the conversational flow of each chatbot planning to interoperate by considering the purpose of each chatbot and the extend of media support (e.g., upload of images or documents, maximum characters accepted or the use of voice). The direction of flow (one-directional or a multi-directional) also plays a significant role in the design. In the context of interoperability of public institutions in the EU, multilingualism and incorporating multiple languages in a network of interoperable chatbots is key. The study considers how to include low resource for which there is less data or a limited digital presence (e.g., using linguistic experts as part of the interoperable project or making use of translation techniques such as pivot languages, MT or using recent technology advantage though LLMs and translation APIs. In terms of security, it is important to ensure that the chatbot only communicates with desired chatbots and protect themselves from abuse.

Other important findings from the study pertain to the impact of UX/UI considerations on interoperable chatbots. The selected trigger mechanism or redirection option will define how the user will be redirected from the host bot to a contributing bot in the network. The study outlined three different options: (A) the host bot directly provides the answer from the contributing bot, (B) the host bot redirects the user to the contributing bot's interface either sharing the content or not, and (C) the host bot hosts the contributing bot in the same interface. Our findings show that different options are better based on the different use cases and certain characteristics of the chatbots in the interoperable network. For example, in scenarios where direct answers can be provided, option A is more effective, while option B is more suitable for cases where the question triggers a predefined flow because additional information is needed, however a combination of the different triggers can also be applied within one interoperable network. Once interoperability is triggered, considering the consent messages and return options, feedback mechanism and monitoring of the different bots in the network needs to be pre-defined and agreed by all participating institutions in the network.

Next, the study focused on viable approaches to create interoperability. First, the distinction was made between the options of having of centralized knowledge repository (PKD) and an interoperability layer (IL). A PKD can be seen as a global repository that catalogues the available chatbots, their domains of expertise, and the types of queries they can handle. It allows other chatbots in the network to be aware of the domain of knowledge of all bots within the network. Here, the PKD would hold the metadata from various chatbots which would encompass details on their capacity, intent, functionality allowing them to interact/ transfer to other chatbots. An IL is an intermediate layer between the host bot and the contributor bots. Unlike the centralized directory it does not store the metadata of the all the chatbots, however it has information on all other bots in the interoperable network, which the host bot could transfer to. Both of these options can be deployed in a centralized, semi-centralized, or de-centralized way. In a centralized approach, the user query is directly forwarded to the PKD or IL which decides which bot in the network should answer the query. In a semi-centralized approach, the host bot first tries to answer the user query and only if it is not possible it will be forwarded to the central PKD or IL which will choose the contributing bot most appropriate to answer the query. In a de-centralized approach, each bot in the network has their own instance of a PKD or IL which decide locally which bot can best answer the user query. The preferred approach and setup can be selected based on the type of bots in the network and an agreement on responsibility for maintenance and monitoring as well as the setup, should be agreed by all parties involved.

The regulatory aspect of interoperability is also important to consider. The study goes over multiple regulations which could be relevant for interoperability (e.g., the AI Act, GDPR, e-privacy Directive and the Interoperable EU Act). The main takeaways from this section are that transparency should be at the forefront of interoperability especially regarding new AI technologies being developed and special care should be taken when private information from users is involved.

Future directions

Based on these insights, we anticipate the establishment of more concrete interoperability standards in the near future. This trend is already evident with the publication of the Interoperable Europe Act in 2024, which signifies a growing commitment to developing and standardizing interoperability within public services. We believe that the introduction of the Interoperable Europe Act signifies a step in the right direction. Moreover, we expect the Act to potentially include an official portal to share knowledge across institutions, which could act as a centralized chatbot registry. In the future, we hope to witness technological advancements that enable enhanced data-sharing protocols, allowing chatbots across various platforms to easily exchange data with immediate synchronization. As interoperability continues to evolve, there may also be legislative efforts requiring chatbot developers to provide explanations for the decisions made by their algorithms. This would ensure greater transparency and accountability in the use of artificial intelligence.

Furthermore, through our interactions with various organizations and drawing on their experiences, we can affirm that collaboration between organizations is crucial for achieving interoperability. This should become a greater focus moving forward. Our exchange with other organizations also showed that the integration of generative AI is on the rise. Indeed, it could be used to build more intelligent, adaptable chatbots that could exchange data and interact with each other. Generative AI could allow easier facilitation of multilingual support, a topic that is increasing in importance and which would ensure public services can serve the public in the native language and enhance cross-border communication.

Based on our exploration of viable approaches for interoperability, the findings show that each option (PKD or IL, and centralized, semi-centralized or de-centralized) is viable. Choosing between the different options will depend on aspects such as the type of chatbots in the interoperable network, the amount of bots to connect, the documentation, cost and energy efficiency, inherent conversational structure and the preference for hosting and commitment to maintenance. Finally, the findings of the study culminated in the creation of a general implementation framework which can be followed by organizations who desire to set up an interoperable network. It covers the four phases necessary, namely the initiation phase, the PoC development, the testing phase, and the deployment and monitoring.

In conclusion, the evolution of interoperability between public administration chatbots is set to have a positive impact on the way public institutions interact with citizens. By leveraging these innovations, public institutions can pave the way for a more integrated, efficient, and citizen-centric digital public service ecosystem. This study not only sheds light on the current state of interoperability but also offers a roadmap for achieving a future where seamless communication redefines public service delivery.

2 Appendix

A. References

Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with applications, 2*.

Akkineni, H., Lakshmi, P., & Sarada, L. (2021). Design and Development of Retrieval-Based Chatbot Using Sentence Similarity. *Lecture notes in networks and systems*, 477-487. doi:https://doi.org/10.1007/978-981-16-2919-8_43

Alves, D., Thakkar, G., & Tadić, M. (2020). Evaluating language tools for fifteen EU-official under-resources languages. *arXiv.org*. doi:https://doi.org/10.48550/arXiv.2010.12428

Avandegraund, M. (2024). *How to Use a Voice Chatbot for Customer Service and Beyond*. Retrieved from BotsCrew: https://botscrew.com/blog/voice-chatbots-for-customer-service/

Babu, A., & Boddu, S. B. (2024). BERT-Based Medical Chatbot: Enhancing Healthcare Communication through Natural Language Understanding. *Exploratory Research in Clinical and Social Pharmacy, 13*. doi:https://doi.org/10.1016/j.rcsop.2024.100419

Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication, 56*, 85-100. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0167639313000988

Botpress. (2023). *List of languages supported by ChatGPT*. Retrieved from Botpress: https://botpress.com/blog/list-of-languages-supported-by-chatgpt

Cappello, P., Comerio, M., & Celino, I. (2017). *BotDCAT-AP: An Extension of the DCAT Application Profile for Describing Datasets for Chatbot Systems.*

ChatBot. (2023 (Updated in 2024)). *Key Chatbot Statistics You Should Follow in 2023*. Retrieved from ChatBot Blog: https://www.chatbot.com/blog/chatbot-statistics/

Church, K. W. (2018). Emerging trends: APIs for speech and machine translation and more. *Natural Language Engineering, 24*(6), 951-960.

CORDIS. (2017). *Competitiveness and innovations frameowrk programme*. Retrieved from https://cordis.europa.eu/docs/projects/cnect/1/325211/080/deliverables/001-eSENSD67EuropeanInteroperabilityReferenceArchitecture.pdf

Core Vocabularies Working Group. (2024). *Core Public Organisation Vocabulary*. Retrieved from https://semiceu.github.io/CPOV/releases/2.1.1/

Del Gratta, R., Frontini, F., Khan, A. F., Mariani, J., & Soria, C. (2014). The LRE Map for under-resourced languages. *Workshop Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era, Satellite Workshop of LREC, 14*. Retrieved from Workshop Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era, Satellite Workshop of LREC.

Dhanani, S. (2023). *Build A Custom AI Chatbot Usin You rOwn Data: A Complete Guide For Developers*. Retrieved from Medium: https://medium.com/locusive/build-a-custom-ai-chatbot-using-your-own-data-a-complete-guide-for-developers-c323ee96c410

Elkhodr, M., Shahrestani, S., & Cheung, H. (2016). The Internet of Things: New Interoperability, Management and Security Challenges. *arXiv.org*. doi:https://doi.org/10.48550/arXiv.1604.04824

EUR-Lex. (2005). *Electronic interchange of data between administration IDA programme*. Retrieved from https://eur-lex.europa.eu/EN/legal-content/summary/electronic-interchange-of-data-between-administrations-ida-programme.html#:~:text=The%20IDA%20(Interchange%20of%20Data,and%2For%20the%20Community%20institutions.

EUR-Lex. (2009). *The IDABC Programme*. Retrieved from https://eur-lex.europa.eu/EN/legal-content/summary/the-idabc-programme-2005-2009.html

European Commission. (2017). *New European Interoperability Framework: Promoting seamless services and data flows for European public administrations*. Retrieved from Publications Office of the European Union: https://ec.europa.eu/isa2/sites/default/files/eif_brochure_final.pdf

European Commission. (2022a). *New Interoperable Europe Act to deliver more efficient public services through improved cooperation between national administrations on data exchanges and IT solutions*. Retrieved from https://ec.europa.eu/commission/presscorner/detail/%20en/ip_22_6907

European Commission. (2022b). Data Act: Commission proposes measures for a fair and innovative data economy. *European Commission*. Retrieved from https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1113

European Commission. (2024a). Data Governance Act explained. *Shaping Europe's digital future*. Retrieved from https://digital-strategy.ec.europa.eu/en/policies/data-governance-act-explained

European Commission. (2024b). Data Act. *Shaping Europe's digital future*. Retrieved from https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1113

European Parliament & Council. (2009). Directive 2009/24/EC of the European Parliament and of the Council on the legal protection of computer programs (Codified version) (Text with EEA relevance). *Official Journal of the European Union*.

European Parliament, & Council of European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Da. *Publications Office of the EU*. Retrieved from https://op.europa.eu/en/publication-detail/-/publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1/language-en

European Parliament, & Council of the European Union. (2002). Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications). *EUR-Lex*. Retrieved from https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex%3A32002L0058

European Parliament, & Council of the European Union. (2024). REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL laying down harmonised rules on artificial intelligence and amending Regulations (Artificial Intelligence Act). *EUR-Lex*. Retrieved from https://data.consilium.europa.eu/doc/document/PE-24-2024-INIT/en/pdf

European Union. (n.d.). *Languages*. Retrieved from https://european-union.europa.eu/principles-countries-history/languages_en#:~:text=The%20EU%20has%2024%20official,%2C%20Slovenian%2C%20Spanish%20and%20Swedish.

Ferman, M. (2018). Towards best practices for chatbots. Retrieved from https://dspace.library.uvic.ca/server/api/core/bitstreams/dd2daec0-45fe-4486-894d-791d9cbfdaae/content

Fromet, E. (2021). *Cultural differences in communicating with chatbots*. Retrieved from Radboud University: https://theses.ubn.ru.nl/server/api/core/bitstreams/54af85d6-1241-477e-a80a-640412cbe807/content

Gain, B., Appicharla, R., Chennabasavaraj, S., Garera, N., Ekbal, A., & Chelliah, M. (2022). Low Resource Chat Translation: A Benchmark for Hindi-English Language Pair. *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas, 1*, 83-96.

Gain, B., Appicharla, R., Chennabasavari, S., Garera, N., Ekbal, N., & Chelliah, M. (2022). Low Resource Chat Translation: A Benchmark for Hindi-English Language Pair. *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas, Volume 1: Research Track*, 83-96.

Gerontas, A. (2020). Towards an e-Government semantic interoperability assessment framework. *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, 767-774.

Gonçalves, D. L. (2022). *Digital public services based on open source: case study on Bürokratt*. Retrieved from European Commission: https://joinup.ec.europa.eu/collection/open-source-observatory-osor/document/digital-public-services-based-open-source-case-study-burokratt

Goodwin, M. (2024, April 09). *What is an API?* Retrieved from IBM: https://www.ibm.com/topics/api

Gupta, A., & Hathwar, D. (2020). Introduction to AI Chatbots. *International Journal of Engineering Research and Technology (IJERT), 9*(7). doi:https://doi.org/10.17577/ijertv9is070143

Hasal, M., Nowaková, J., Saghair, A., Abdulla, K., Snášel, V., & Ogiela, L. (2021). Chatbots: Security, privacy, data protection, and social aspects. *Concurrency and Computation: Practice and Experience, 33*(19).

IEEE Standards Association. (2016). *IEEE Approves New Standards Project IEEE P2413TM - Standard for an Architectural Framework for the Internet of Things (IoT)*. Retrieved from https://standards.ieee.org/news/2014/ieee_p2413.html

Jackson, S. (2023). *Check out ChatGPT's guidelines for difficult conversations, including rules for discussing culture war topics and responding to users trying to trick the chatbot*. Retrieved from Business Insider Netherlands: https://www.businessinsider.nl/check-out-chatgpts-guidelines-for-difficult-conversations-including-rules-for-discussing-culture-war-topics-and-responding-to-users-trying-to-trick-the-chatbot/

Jia, J. (2009). CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning. *Knowledge-Based Systems, 22*(4), 249-255. doi:doi:https://doi.org/10.1016/j.knosys.2008.09.001

Kim, J., Chua, M., Rickard, M., & Lorenzo, A. (2023). ChatGPT and large language model (LLM) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine. *Journal Of Pediatric Urology, 19*(5), 598-604. doi:https://doi.org/10.1016/j.jpurol.2023.05.018

Kim, J.-K., Kim, Y.-B., Sarikaya, R., & Fosler-Lussier, E. (2017). Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2832-2838).

Kocur, S. (2023). *Confidence Score*. Retrieved from Chatbot.com: https://www.chatbot.com/help/settings/confidence-score/

Laumann, F. (2022, January 12). *Challenges in using NLP for low-resource languages and how NeuralSpace solves them*. Retrieved from Medium: https://medium.com/neuralspace/challenges-in-using-nlp-for-low-resource-languages-and-how-neuralspace-solves-them-54a01356a71b

Leitold, H. (2010). Challenges of eID Interoperability: The STORK Project. *Privacy and Identity Management for Life*, 144-150.

Miessner, S., Hagstöm, V., Halonen, L., Humalajoki, M., Ilmarinen, O., Kiviranta, M., . . . Wires, M. (2019). *Starting up Smoothly - Experiment Evaluation*. Retrieved from Helsinki: Finnish Immigration Service; Finnish Tax Administration; Finnish Patent and Registration Office: https://migri.fi/documents/5202425/0/Starting+up+Smoothly+experiment+evaluation_CMYK.PDF/87688320-dfef-9246-6c24-c1ac8e436103/Starting+up+Smoothly+experiment+evaluation_CMYK.pdf

Miessner, S., Hagström, V., Halonen, L., Humalajoki, M., Ilmarinen, O., Kiviranta, M., . . . Wired, M. (2019). Starting up Smoothly. Retrieved from https://migri.fi/documents/5202425/0/Starting+up+Smoothly+experiment+evaluation_CMYK.PDF/87688320-dfef-9246-6c24-c1ac8e436103/Starting+up+Smoothly+experiment+evaluation_CMYK.pdf

Mirza, N. A. (2023). *Navigating Boundaries: What Not to Try on an AI Chatbot*. Retrieved from LinkedIn: https://www.linkedin.com/pulse/navigating-boundaries-what-try-ai-chatbot-noor-abbas-mirza

Pacheco, V. (2022, August 29). *What are the most widely spoken EU languages?* Retrieved from AbroadLink: https://abroadlink.com/blog/most-widely-spoken-eu-languages

Parliament, E., & Union, C. o. (2009). *Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs*. Retrieved from EUR-Lex: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32009L0024

Paul, A., Latif, A. H., Adnan, F. A., & Rahman, R. M. (2019). Focused domain contextual AI chatbot framework for resource poor languages. *Journal of Information and Telecommunication, 3*(2), 248-269.

Paul, M., Finch, A., & Sumita, E. (2013). How to choose the best pivot language for automatic translation of low-resource languages. *ACM Transactions on Asian Languages Information Processing (TALIP), 12*(4), 1-17.

Pohrebniyak, I. (2024). *LLMs & Generative AI solutions for enterprise business*. Retrieved from Master of Code Global: https://masterofcode.com/blog/llms-and-generative-ai-solutions-for-enterprise-business

Publications Office of the European Union. (2015). *e-Government Core Vocabularies handbook*. Retrieved from https://ec.europa.eu/isa2/sites/isa/files/publications/e-government-core-vocabularies-handbook.pdf

Ramesh, K., Ravishankaran, S., Joshi, A., & Chandrasekaran, K. (2017). A survey of design techniques for conversational agents. *International conference on information, communication and computing technology*, 336-350.

Ranathunga, S., Lee, E. A., Skenduli, M. P., Shekhar, R., Alam, M., & Kaur, R. (2023). Neural Machine Translation for Low-resource Languages: A Survey. *ACM Digital Library, 55*(11), 1-37. doi:https://doi.org/10.1145/3567592

Rybakova, M. (2022). *4 Evolving technologies that are empowering chatbots*. Retrieved from AiThority: https://aithority.com/guest-authors/4-evolving-technologies-that-are-empowering-chatbots/

Schuster, S., Gupta, S., Shah, R., & Lewis, M. (2018). Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*.

Scotti, V., Sbatella, L., & Tedesco, R. (2023). A Primer on Seq2Seq Models for Generative Chatbots. *ACM Computing Surveys, 56*(3), 1-58. doi:https://doi.org/10.1145/3604281

Šoštarić, M., Pavlović, N., & Boltužić, F. (2019). Domain adaptation for machine translation involving a low-resource language: Google AutoML vs. from-scratch NMT systems. *Translating and the Computer, 41*, 113-124.

Suta, P., Lan, X., WU, B., Mongkolnam, P., & Chan, J. (2020). An Overview of Machine Learning in Chatbots. *International Journal of Mechanical Engineering and Robotics Research*, 502-510. doi:https://doi.org/10.18178/ijmerr.9.4.502-510

Teo, P. (2021, February 4). *How to Make Conversational AI Work for Low-Resource Languages*. Retrieved from KeyReply: https://www.keyreply.com/blog/conversational-ai-low-resource-languages

Tirosh, O. (2024). *European Languages: Exploring the Languages of Europe*. Retrieved from Tomedes: https://www.tomedes.com/translator-hub/european-languages#:~:text=Europe%20is%20home%20to%2024,a%20huge%20amount%20of%20variety.

Tsang, S. (2022). *Review — Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation*. Retrieved from Medium: https://sh-tsang.medium.com/review-googles-multilingual-neural-machine-translation-system-enabling-zero-shot-translation-bd230aa9ef7f

Tsang, S.-H. (2022, February 5). *Review — Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation*. Retrieved from Medium: https://sh-tsang.medium.com/review-googles-multilingual-neural-machine-translation-system-enabling-zero-shot-translation-bd230aa9ef7f

Union, E. P. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Da*. Retrieved from EUR-Lex: https://eur-lex.europa.eu/eli/reg/2016/679/oj

Utz, C., Degeling, M., Fahl, S., Schaub, F., & Holz, T. (2019). (Un)informed Consent: Studying GDPR Consent Notices in the Field. *arXiv.org*. doi:https://doi.org/10.48550/arXiv.1909.02638

Valencia, O. A., Thongprayoon, C., Jadlowiec, C. C., Mao, S. A., Miao, J., & Cheungpasitporn, W. (2023). Enhancing Kidney Transplant Care through the Integration of Chatbot. *Healthcare, MDPI, 11*(18).

Verma, S. (2024). *Voice Assistants vs. Text Chatbots: Business AI Choice*. Retrieved from Chat360: https://chat360.io/blog/voice-assistants-vs-text-based-chatbots-choosing-the-right-ai-for-your-business/

Wołk, A., Skowrońska, H., & Skubis, I. (2021). Multilingual Chatbot for E-Commerce: Data Generation and Machine Translation. *PACIS 2021, Proceedings, 232*, 1-1.

Zaiets, S. (2021). *Translation with Pivot Languages: How to Do it Right (and Easy)*. Retrieved from Gridly: https://www.gridly.com/blog/pivot-language-localization/

B. Additional Content

B1. Additional information on EU interoperability standards

B.1.1 EU interoperability programs

Since as early as 1995, interoperability was on the EU agenda, with the Interchange of Data across Administration (IDA) programme setting the path. The Digital Europe Programme (DEP) is the most recent program, planned to run until 2027 by promoting digital transformation across Europe and interoperability as one of the levers for this. The paragraphs below present examples of past and ongoing programmes and initiatives of interoperability in Europe.

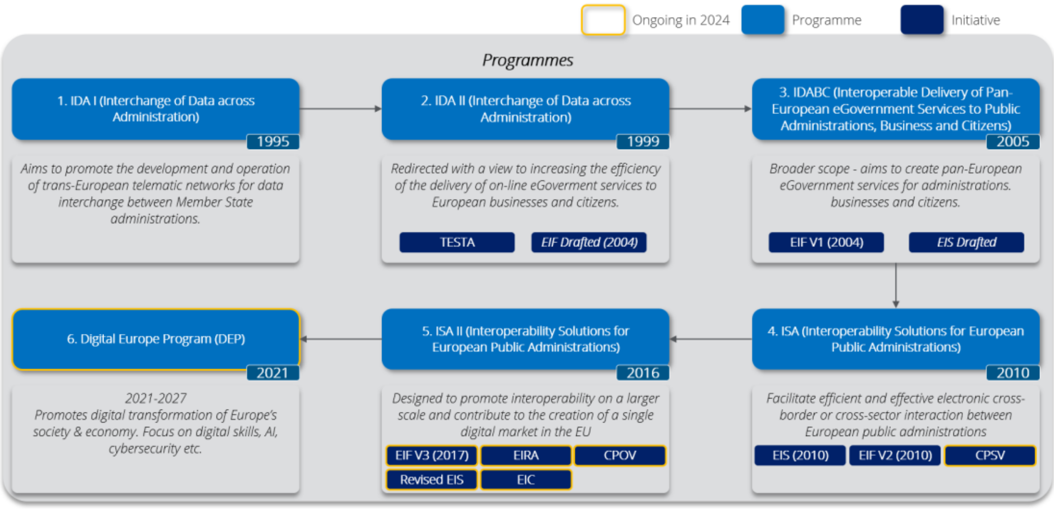


Figure 36. European Interoperability Programmes

Interchange of Data across Administrations (IDA) – (1995-1999)

IDA programme (EUR-Lex, Electronic interchange of data between administration IDA programme, 2005) was launched by the European Commission in 1995 and closed in 1999. Its objective was to support the implementation of the EU's internal market through the establishment of trans-European telematic networks between administrations of Member Nations, allowing the exchange of information.

More extensively, the objectives pursued with IDA programme were the following:

- **Improve administrative cooperation and efficiency**: The goal here was to facilitate the digital exchange of data and information across different European public administrations to improve efficiency, service delivery, and cooperation.
- **Promote a more unified Europe**: Through fostering interoperability among EU nations, IDA aimed to bring about a more interconnected Europe.
- **Develop telematic networks**: Part of the IDA's objectives included the development of the necessary infrastructure to facilitate these exchanges, including the creation and maintenance of telematic networks for data exchange.
- **Enhance data protection**: In line with increasing digital exchanges, IDA also aimed to ensure the secure transmission of information to safeguard personal data and privacy.

The emphasis was on developing infrastructure, standardizing formats, and incorporating new ICT-based processes for electronic government. This includes A2A - Administrations to Administrations interactions, which refers to the exchanges between public authorities. IDA was able to achieve its objectives by creating sector-based networks which allowed for the digital and seamless exchange of data across different EU administrative bodies.

Interchange of Data across Administrations – Phase 2 (IDA II) – (1999-2005)

It was the extension of the IDA programme, it ran from 1999 to 2005 (EUR-Lex, Electronic interchange of data between administration IDA programme, 2005). Its aim was to aid in the delivery of electronic public services and facilitate access by EU citizens and businesses to public sector information. IDA II also built on cooperation between European public administrations in areas such as debt recovery, bankruptcy, and border crossings.

In addition to retaining IDA's original objective, it also advocated for the growth of eGovernment services for businesses and citizens (A2B - Administration to Businesses and A2C - Administrations to Citizens). As such, its activities pertained to the relations between public administrations, citizens, and businesses. The first version of the European Interoperability Framework (EIF) was formulated and published in 2004 in the context of IDABC.

To summarize this programmes' objectives extended the previous IDA phase by:

- **Extending the benefits of the networks to EU businesses and citizens:** Enabling them to participate more directly and benefit from the unified data exchange framework.
- **Fostering the spread of best practices and innovative telematic solutions in administrations**: Further enhancing the efficiency and effectiveness of public administration processes.
- **Continuing to develop and maintain the infrastructure:** Necessary for safe and reliable data interchange.

Building on IDA's groundwork, IDA II implemented TESTA (Trans European Services for Telematics between Administrations), a telematics network, to achieve a wider and more efficient data exchange across different EU administrative services. One example is the use of fingerprints in the refugee office to identify asylum seekers and if they have already requested asylum in other Member States. When comparing fingerprints between Member States, the Eurodac system utilizes TESTA to transmit the fingerprints. With the implementation of TESTA, the transfer of fingerprint information was nearly immediate, significantly improving the response time from the previous 24-hour rate. This programme outcome resulted in greater benefits for EU businesses and citizens through better access to and interaction with public services. There was a notable increase in innovative telematic solutions and the dissemination of best practices. Data protection was further improved, and the EU administrations felt the positive effects of more streamlined operations and reliable data interchange.

Interoperable Delivery of European eGovernment Services to public Administrations, Businesses, and Citizens (IDABC) – (2005-2009)

IDABC was an extension of the IDA initiative (EUR-Lex, 2009). It operated from 2005 to 2009, aiming at delivering electronic government services across Europe in an agile and transparent way. A major part of this programme was to ensure the interoperability between information systems so that citizens and businesses could readily access government services.

This programme's primary objectives focused on:

- **Enhancing Administrative Efficiency and Effectiveness:** A primary goal was to improve administrative cooperation and efficiency within public services across the EU.
- **Achieving Interoperability:** IDABC aimed to foster interoperability between different European public administrations, enabling efficient data exchange and communication, notably on the basis of a European Interoperability Framework (EIF).
- **Enabling Cross-Border Services**: Recognizing the increasing mobility of citizens and businesses within the EU, the programme aimed to ensure that public services function efficiently across national borders.
- **Involving Citizens and Businesses:** While the focus was primarily on administrative processes, there was a clear objective to extend the benefits of the programme to EU businesses and citizens by improving their interactions with public administrations.

IDABC ensured interoperable e-Government services by establishing guidelines, providing tools, and supporting the use of open standards to enable cross-border collaboration and data exchange. The first version of the European Interoperability Framework (EIF) was implemented as part of this programme. The European Interoperability Strategy (EIS) was developed during the IDABC programme as well and was adopted following a public consultation under the framework of the ISA programme. This programme provided the foundations of the Interoperability Solutions for European Public Administrations (ISA).

Interoperability Solutions for European Public Administrations (ISA) – (2010-2015)

This programme ran from 2010 to 2016. It was designed to improve cooperation among European public administrations, facilitate the efficient and effective delivery of European public services to the benefits of citizens and businesses. It especially dealt with interoperability and the use of IT in public administration.

The second version of the European Interoperability Framework (EIF) was launched in 2010 in this context. The European Interoperability Strategy (EIS) was drafted during the IDABC programme and released in the same year as the second version of the EIF, 2010. The Core Vocabularies, including CPSV, CPOV, among others, were also unveiled although CPOV was only developed during ISA II.

Core Public Service Vocabulary (CPSV) – As part of semantic interoperability, the CPSV provides a simplified, reusable, and extendable data model to describe a public service in a structured and semantically-relevant manner. It aids in harmonizing the way public services are described in catalogues and portals, promoting better interoperability among European public administrations. CPSV has known several revisions throughout the years to incorporate user feedback, evolving requirements, and lessons learned.

To resume, three objectives were considered for this programme:

- **Facilitate Efficient Electronic Cross-Border Interaction**: ISA sought to ensure that citizens, businesses, and public administrations in Europe could take full advantage of the digital single market.
- **Develop The European Interoperability Framework and Strategy (EIF/EIS)**: ISA developed the European Interoperability Framework (EIF) second version and the European Interoperability Strategy (EIS), which provided strategic guidance for more efficient and effective interoperability in European public services.
- **Offer Interoperable Solutions**: The programme also focused on providing specific interoperable solutions to support cross-border and cross-sector interactions.

ISA developed and promoted interoperable solutions, such as common frameworks and standards, which facilitated data exchange and collaboration between various national and regional administrations. This resulted in supporting the interoperability of eGovernment services, notably with the development of EIF and EIS, and the promotion of the sharing and reuse of IT solutions among European public administrations.

Interoperability Solutions for European Public Administrations – Phase 2 (ISA II) – (2016-2020)

The ISA II programme is the follow-up to the ISA programme. It began in 2016 and ended in 2020. The main goal of ISA II is to ensure that public administrations, businesses and citizens can seamlessly interact within a Digital Single Market, creating a more interconnected and efficient European administration through the use of digital means. The focus was on creating digital solutions that would allow public administrations, businesses, and citizens in Europe to gain from interoperable cross-border and cross-sector public services. A new version of the EIF was launched in 2017 as part of ISA II.

The European Interoperability Architecture (EIRA) was established to categorize and manage building blocks pertinent to interoperability, used to provide digital public services. The aim of EIRA is to support interoperability and repurpose when developing public services. EIRA v1.0.0 was released in 2016, v2.0.0 in 2017, and v3.0.0 in 2019. New editions of Core Vocabularies such as CPSV, CPOV, etc. were conceptualized and unveiled.

Core Public Organization Vocabulary (CPOV) – This interoperability asset from the ISA II Programme provides a basic conceptual model for describing public organizations in Europe. It aims to foster the seamless exchange of information between public bodies and facilitate the creation of trans-European digital public services.

They were both developed to promote better interoperability, providing a standardized model to describe public services and organizations. Evaluation instruments were also developed and launched, for instance:

Interoperability Maturity Assessment of Public Services (IMAPS) – This is an assessment tool designed to evaluate the interoperability maturity of a public service offered by public administrations. The tool helps identity gaps and areas of improvement while assisting the public administrations in structuring their efforts towards achieving higher levels of interoperability.

The Interoperability Quick Assessment Tool (IQAT) – This is a tool aimed at public administrations intending to self-assess their level of interoperability without carrying out an exhaustive analysis.

In conclusion this programme permitted to:

- Ensured coordination of interoperability activities at EU level
- Developed and operated solutions for the public administrations on the basis of businesses and citizens' needs
- Put in place necessary instruments to boost interoperability at EU and national level with the revised versions of the EIF, EIS, the implementation of the EIRA and EIC

In conclusion, EU programmes related to interoperability play a crucial role in facilitating communications across different networks and systems, fostering innovation and collaboration, and promoting a seamless digital single market across the European Union.

B.1.2 EU interoperability initiatives

The programmes lead to many interesting initiatives that can enhance administrative processes to create a more connected ecosystem among the Member States and citizens. The paragraphs below present examples of past and ongoing initiatives, to give examples of the outcomes of the above programmes as well as the basic structure of interoperability frameworks and best practices.

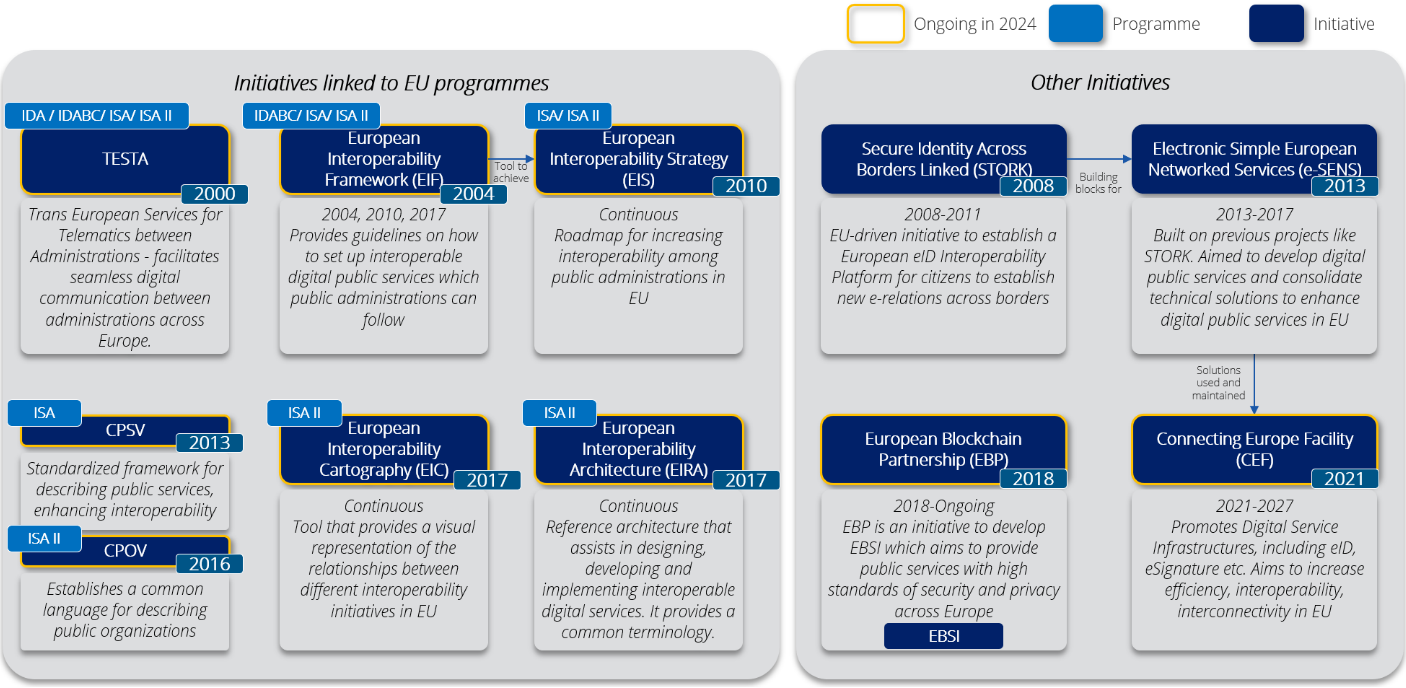


Figure 37. Zoom-in on European Interoperability Initiatives

Initiatives linked to EU programmes

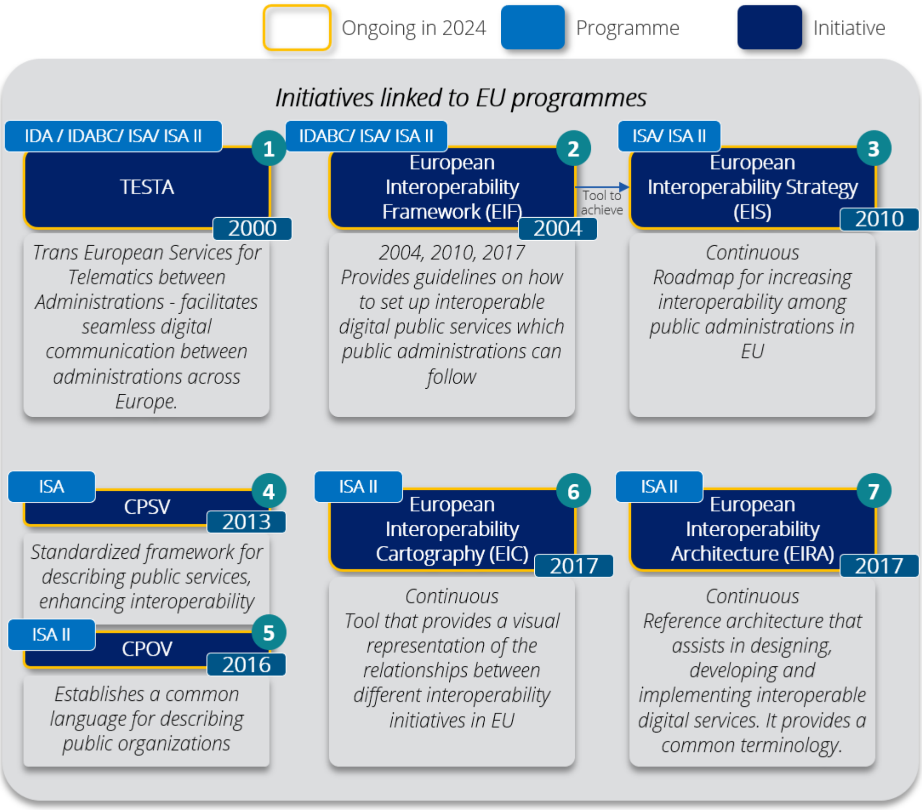


Figure 38. EU programmes initiatives

1. Trans European Services for Telematics between Administrations (TESTA)

As previously mentioned, TESTA is a network put in place to facilitate secure and efficient communication between public administrations across Europe. It evolved throughout the years, across the different programmes, to keep up with technological advancement. TESTA (1996-2000), focused on providing reliable network infrastructure for the exchange of electronic data (e.g., emails, documents). TESTA-II (2000-2006) expanded its reach to include more European administrations and incorporate more advanced features (e.g., security measures, increased capacity to handle large volume of data traffic). TESTA (2006-2013) introduced new services and protocols (e.g., electronic signatures, secure file transfer capabilities, RESTREINT UE to allow EU classified information exchange). The latest version TESTA-ng (2013-2020) aimed to provide more flexible and scalable platforms.

2. European Interoperability Strategy (EIS)

Set up in 2010 to complement the EIF, the EIS helps create an interconnecting public sector linking organizational operations, processes and data and serves as one of the key pillars for achieving interoperability. It provided an overarching strategic context for EU public service actions, intending to lead and coordinate interoperability efforts throughout Europe. The EIS corresponds to the governance of the interoperability policy, while EIF corresponds more to the conception/implementation. It aims to provide guidance and prioritise the actions need to improve exchange among the EU institutions whether across border of sectors.

3. Core Public Service Vocabulary (CPSV)

The CPSV is part of the e-Government Core Vocabularies which provide a starting point for promoting semantic interoperability among EU public administrations and reducing conflicts related to it. Core Vocabularies are expressed in three formats: conceptual model and spreadsheet, XML schema and RDF schema. The CPSV was created in 2012-2013 to capture the characteristics of a service offered by public administration such as the title and description of the services, the outputs it generates, locations of the public service (Publications Office of the European Union, 2015).

4. Core Public Organization Vocabulary (CPOV)

The CPOV is designed to support the exchange of information about public organizations. It aspires to become a common data model for describing the organisation in the EU (Core Vocabularies Working Group, 2024). Using the vocabulary will facilitate the process for institutions publishing data about organisations to:

- Develop common information systems,
- Link data from public organisations to other data sets,
- Manage a cross border repository of public services and organisations,
- Enable the creation of interoperable catalogues of public organisation in Europe and beyond,
- Link public service provided, budgets, and other types of resources with certain public organisations

5. European Interoperability Cartography (EIC)

Very limited information on the EIC is provided, it is a repository of interoperability solutions for EU public administrations, and it should be presented in a common format and in compliance with specific re-usability, interoperability criteria (that can be represented in the EIRA). No official start and end dates are given for the EIC as it is a guide providing a snapshot of the extent to which European countries have implemented EIF recommendations. It can be considered as an ongoing effort to visualize the application of EIF and other interoperability efforts.

The following table highlights the specific contributions made by EIC to EIF's recommendations, discussed in earlier sections. For instance, regarding recommendation 6 on reusability, EIC can assist by providing a list of interoperable solution (Solution: European Interoperability Cartography v1.1.0 | Joinup (europa.eu)):

Table 14. EIC relation to the EIF's recommendations

ID	EIFs	Layers	Recommendations
1	**Principles**	Principle 1: Subsidiarity and proportionality	Recommendation 1: Allows a list of solutions assessed as reusable and interoperable
2	**Principles**	Principle 4: Reusability	Recommendation 6: Provides a list of interoperable solutions, facilitating reuse and sharing
3	**Interoperability layers**	Interoperability governance	Recommendation 23: List of interoperability solutions which can be consulted by public administrations
4	**Interoperability layers**	Organisational interoperability	Recommendation 28: Present and document interoperable solutions using EIRA framework
5	**Interoperability layers**	Semantic interoperability	Recommendation 31: Guide public administrations in establishing information management
6	**Conceptual model**	Catalogues	Recommendation 44: Identifies solutions in a common way using EURA as an architecture reference

This tool represents a concise and feasible way for different countries to assess their interoperability status and progress – more information on the tool and if the repository of the list can be provided is pending their response.

6. European Interoperability Architecture (EIRA)

The EIRA was developed as part of the ISA programme (2015) and is still ongoing. It is a blueprint that contributes to the development of the EIF by promoting the same interoperability approach across all European public services. It is an essential tool to support the European digital single market. The EIRA provides a framework for achieving more effective and efficient electronic interactions among European public administrations, businesses, and citizens. Employing a clear and universally accepted vocabulary, EIRA facilitates interoperability in designing, describing and implementing pan-European public services. The EIRA has four main characteristics:

1. Common terminology to achieve coordination: Applied to all four views – legal, organisational, semantic, technical (Chapter #4 EIRA Glossary | Joinup (europa.eu))
2. Reference architecture for delivering digital public services (Chapter #2 Key Concepts and Archimate® Notation | Joinup (europa.eu), Chapter #3 Views, Viewpoints and Architecture Building Blocks | Joinup (europa.eu))
3. Technology and service-oriented architecture (SOA) style
4. Alignment with EIF and TOGAF: The views of EIRA correspond to the interoperability level in the EIF (legal, organisational, semantic and technical interoperability). EUIRA reuses terminology and paradigms from TOGAF (architecture patterns, building blocks, and views).

The EIRA will promote greater awareness and adoption of the principles and recommendations outlined by the EIF. It will also provide organizations a common language of Architecture Building Blocks (ABB) for the design and comparison of the solution architecture of the solutions. It particularly benefits architects, business analysts and portfolio managers within the public administrations by responding to some of the above needs, characteristics mentioned.

6. European Interoperability Framework (EIF)

This section covers additional information regarding the three sections of the EIF.

EIF Section 1: Underlying principles

The 12 principles laid out in the EIF sets the basis recommendations for interoperability. In the table below, the principles are briefly explained and out of the 19 recommendations of this section, the recommendations most relevant for interoperability are added in the last column. A more extensive table with all the recommendations can be found in Appendix 0.

Table 15. EIF underlying principles and recommendations

ID	Principle	Description	Recommendations
1	**Subsidiarity and proportionality**	This principle guides EU decisions to be citizen-oriented and minimal.	
2	**Openness**	This refers to freely accessible data, flexible open-source software, and open specifications that facilitate reuse and customization.	Recommendation 2: Publish the data you own as open data unless certain restrictions apply. Recommendation 3: Ensure a level playing field for open-source software and actively consider its use, taking into account the total cost of ownership.
3	**Transparency**	This involves making administrative rules and processes visible, ensuring interface availability with internal systems for interoperability and data reuse, and respecting legal frameworks.	Recommendation 5: Ensure internal visibility and provide external interfaces for European public services.
4	**Reusability**	This denotes adopting and sharing proven solutions, which enhances interoperability, saves resources, and advances the EU's digital single market.	Recommendation 6: Reuse and share solutions and cooperate in the development of joint solutions when implementing European public services.
5	**Technological neutrality and data portability**	Administrations should prioritize functionality to minimize dependencies and adapt to technology evolution.	Recommendation 8: Avoid imposing technology-specific or disproportionate solutions on citizens, businesses, and other administrations. Recommendation 9: Ensure data portability for European public services without unjustified restrictions, if legally possible.
6	**User-centricity**	European public services should be designed considering user needs, offering multi-channel delivery, a single contact point, and utilizing systematic user feedback.	

ID	Principle	Description	Recommendations
7	**Inclusion and accessibility**	Inclusion should bridge social and economic divides, while accessibility ensures services are available to disadvantaged groups.	Recommendation 14: Ensure accessibility of all European public services for all citizens, including disadvantaged groups. Comply with recognized e-accessibility specifications for digital public services.
8	**Security and privacy**	Citizens and businesses interacting with public authorities must be assured of a secure and trustworthy environment that complies with relevant regulations, including data protection and electronic identification.	Recommendation 15: Define a common security and privacy framework and establish processes for public services to ensure secure and trustworthy data exchange between public administrations and in interactions with citizens and businesses.
9	**Multilingualism**	Multilingualism is vital in European public services to cater to diverse Member State users and ensure effective usage.	Recommendation 16: Use information systems and technical architectures that cater for multilingualism when establishing a European public service. Decide on the level of multilingualism support based on the needs of the expected users.
10	**Administrative simplification**	Public administrations should aim to streamline their processes to lessen the legislative compliance burden.	
11	**Preservation of information**	The law mandates long-term storage of decisions and data, requiring electronic records to be preserved and updated for longevity and integrity.	Recommendation 18: Formulate a long-term preservation policy for information related to European public services and especially for information that is exchanged across borders.
12	**Assessment of Effectiveness and Efficiency:**	Assessing the value of interoperable European public services involves considering factors like return on investment, adaptability, efficiency, transparency, and user satisfaction.	Recommendation 19: Evaluate the effectiveness and efficiency of different interoperability solutions and technological options considering user needs, proportionality and balance between costs and benefits.

The 12 principles of the European Interoperability Framework (EIF), and the associated recommendations, play a crucial role in supporting interoperability. By prioritizing user-centricity, openness, and transparency, the EIF ensures that the needs of various stakeholders are considered, fostering collaboration, and understanding. The principles of reusability and interoperability by default promote the efficient integration and cooperation of different systems and services. Emphasizing security, privacy, multilingualism, and the use of standards ensures the protection of data, accessibility, and seamless information exchange.

Finally, through cooperation and collaboration, the EIF encourages shared learning and best practice sharing across EU member states. In the specific case of interoperable chatbots, they enable seamless communication and information exchange between different public institutions, improving service delivery, enhancing user experience, and reducing administrative burdens for citizens.

EIF Section 2: **Interoperability model**

The European Interoperability Framework (EIF) also outlines four layers of interoperability: legal, organizational, semantic, and technical, along with a cross-cutting component of integrated public service governance and a background layer of interoperability governance. The EIF's interoperability layers work together to ensure that public institutions and eGovernment systems and services can interoperate effectively, aiming to enable efficient and seamless digital service delivery across Europe. This section covers 14 of the 47 recommendations. Each layer comes with a set of recommendations. Below is a brief explanation of each layer and their recommendations:

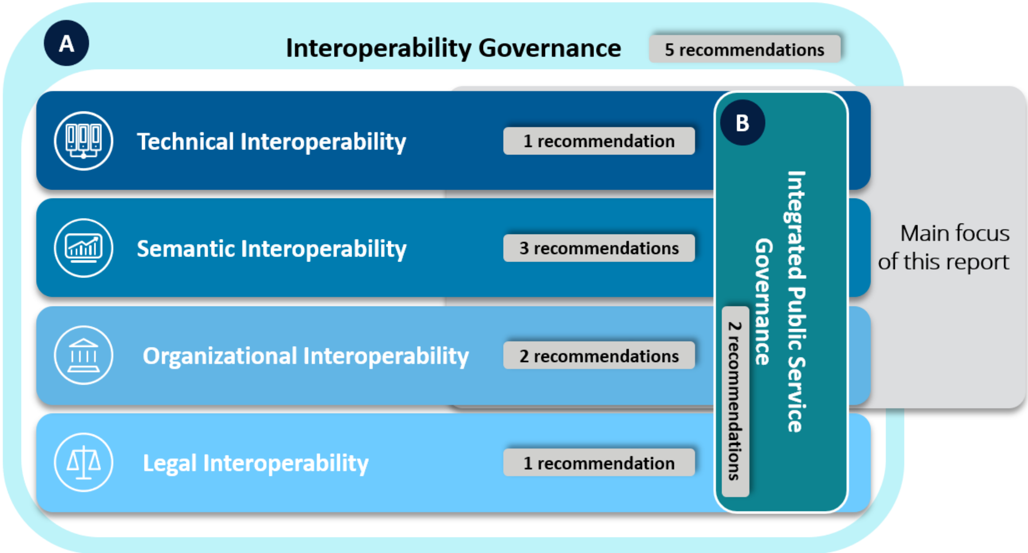


Figure 39. EIF governance model: Interoperability levels, cross-cutting components of integrated public service governance (B) and background layer of interoperability governance (A) and their 14 recommendations

Table 16. EIF interoperability layers and recommendations

ID	Layer	Description	Recommendations
1	**Technical interoperability**	This layer focuses on the communication of data between systems, servers, and applications. It ensures there is a functional communication infrastructure in place that allows for data exchange.	Recommendation 33: Use open specifications, where available, to ensure technical interoperability when establishing European public services.
2	**Semantic interoperability**	This layer involves the ability of systems to not only exchange data but also to interpret and understand the shared data, attributing correct meaning to it.	Recommendation 30: Perceive data and information as a public asset that should be appropriately generated, collected, managed, shared, protected and preserved.
3	**Organisational interoperability**	This layer concerns how well different organisations can work together seamlessly. This involves goals, governance, processes, and often requires coordination between organizations.	Recommendation 29: Clarify and formalise your organisational relationships for establishing and operating European public services.

ID	Layer	Description	Recommendations
4	**Legal interoperability**	This layer focuses on ensuring that organisations operating under different legal frameworks, policies and strategies are able to work together.	Recommendation 27: Ensure that legislation is screened by means of 'interoperability checks', to identify any barriers to interoperability.
A	**Interoperability governance**	This refers to decisions on interoperability frameworks, institutional arrangements, organisational structures, roles and responsibilities, policies, agreements, and other aspects of ensuring and monitoring interoperability at national and EU levels.	Recommendation 21: Put in place processes to select relevant standards and specifications, evaluate them, monitor their implementation, check compliance and test their interoperability. Recommendation 22: Use a structured, transparent, objective and common approach to assessing and selecting standards and specifications. Take into account relevant EU recommendations and seek to make the approach consistent across borders. Recommendation 23: Consult relevant catalogues of standards, specifications and guidelines at national and EU level, in accordance with your NIF and relevant DIFs, when procuring and developing ICT solutions.
B	**Integrated public service governance**	A cross-cutting component that involves improving governance of interoperability activities, establishing cross-organisational relationships, and streamlining processes	Recommendation 25: Ensure interoperability and coordination over time when operating and delivering integrated public services by putting in place the necessary governance structure.

The interoperability layers work in a complementary, interconnected manner – legal frameworks shape organisational processes, which in turn define semantic and technical interactions, all under a larger umbrella of governance.

EIF Section 3: Conceptual model

This section of the EIF aims to define a conceptual model for integrated public services, promoting the idea of interoperability-by-design as a standard approach for the design and operation of European public services. The conceptual model for public services covers the design, planning, development, operation, and maintenance of integrated public services at all governmental levels from local to EU level. The interoperability-by-design philosophy promotes reusability to ensure the easy availabilities of information and services. Below is an overview of what constitutes the conceptual model and out of the 14 recommendations of this section, the recommendations most relevant for interoperability will be included. A more extensive table with all the recommendations can be found in Appendix 0.

Table 17. EIF conceptual model and recommendations

ID	Layer	Description	Recommendations
1	**Conceptual model**	Promotes interoperability by designing European public services according to certain interoperability and reusability requirements. The public administrations identify, negotiate, and agree on a common approach to interconnecting service components at different national administrative.	Recommendation 35: Decide on a common scheme for interconnecting loosely coupled service components and put in place and maintain the necessary infrastructure for establishing and maintaining European public services.
2	**Internal information sources and services**	Reusability of IT solutions and data aids interoperability, improves quality, and saves money. This promotes a digital single market in the EU through new business models and open-source software.	
3	**Base registries**	Trusted sources of reliable data that can be used by others. They ensure data quality, integrity, and accessibility. Compliance with privacy regulations is necessary, and a data quality plan should be implemented.	Recommendation 37: Make authoritative sources of information available to others while implementing access and control mechanisms to ensure security and privacy in accordance with the relevant legislation
4	**Open data**	Promotes data reuse for transparency, competition, and innovation. It emphasizes interoperability and addresses barriers.	
5	**Catalogues**	Exist to help others find reusable resources, and commonly agreed descriptions are needed to enable interoperability between them.	Recommendation 44: Put in place catalogues of public services, public data, and interoperability solutions and use common models for describing them.
6	**External information sources and services**	Public administrations need to utilize services and information from external sources, such as third-party services and open data.	

ID	Layer	Description	Recommendations
7	**Security and privacy**	Crucial in public services. Adherence to privacy and security principles, compliance with data protection regulations, and implementation of risk management and business continuity plans are essential. Secure information exchange systems and mechanisms should be in place.	Recommendation 46: Consider the specific security and privacy requirements and identify measures for the provision of each public service according to risk management plans.

The EIF works towards technical integration through open specifications, aligning between administrations for improved electronic communication. This conceptual model promotes both reusability and interoperability of information across EU public administrations.

Other EU initiatives linked to interoperability

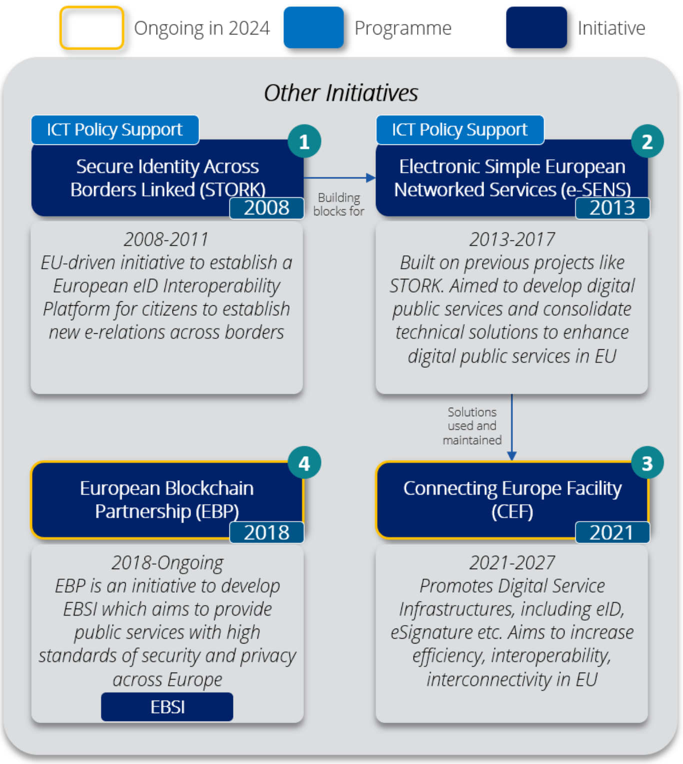


Figure 40. Other interoperability related initiatives

1. Secure Identity Across Borders Linked (STORK)

STORK ran from 2008 to 2011 and was an EU co-funded project under the ICT Policy Support Programme that aimed to develop and test a European eID interoperability platform that would allow citizens to establish new e-relations across borders, just by presenting their national eID. One if the issue the STORK had to address was the heterogenous nature of eID in Europe. Another issue regarding data protection requirements had the STORK identify that obtaining explicit user consent is necessary to establish legitimacy of cross border eID processing (Leitold, 2010).

STORK project's focus was mostly on technical interoperability, involving the development of a platform for secure and effective cross-border online identity checks. It also had relevance to semantic interoperability since common understanding of identities and access rights was required across different national systems.

2. Electronic Simple European Networked Services (e-SENS)

This initiative took place from 2013 to 2017. It was a pilot to broaden the use of electronic exchange of information (both ways) in a safe and secure way to help businesses and public administrations across Europe. E-SENS aimed to develop the digital public services across Europe. This project was a part of the EU's efforts to create a Single Digital Market.

The project aimed primarily at improving semantic interoperability, as it focused on creating shared understanding and common specifications (e.g., documented eDelivery, e-Document, e-Invoicing, eSignature, eID) for digital public services across Europe. Nonetheless, it also had implications for technical interoperability by developing protocols for data exchange, and organizational interoperability through coordinated practices across nations.

It can be linked to EIRA, while EIRA concentrates on the use of the building blocks (external interface), e-SENS focuses on the architecture specifications of the building blocks (internal specifications) (CORDIS, 2017).

3. Connecting Europe Facility (CEF)

The CEF is operational since 2014 and is still ongoing. The goal is to create powerful, high-speed broadband networks and digitally interconnected services, aiding the connections between the EU, the national, regional, or local levels. This is a key EU instrument to facilitate cross-border interaction between public administrations, businesses, and citizens, by deploying digital service infrastructures (DSIs) and building networks in areas that have market failures.

The CEF primarily affects technical and semantic interoperability by building digital service infrastructures for effective communication and understanding of data across borders. Additionally, it enhances organizational interoperability by facilitating cross-border interactions between public administrations, businesses, and citizens.

4. European Blockchain Partnership (EBP) & European Blockchain Services Infrastructure (EBSI)

The EBP was set up in 2018 and is still ongoing. It is a declaration signed by 29 European countries, including EU members and Norway, that commit to cooperate in establishing a European Blockchain Services Infrastructure (EBSI).

EBSI was launched in 2020 and is still ongoing. It is a joint initiative from the European Commission and the European Blockchain Partnership (EBP) to deliver EU-wide cross-border public services using blockchain technology, which aims to achieve technological and semantic interoperability. EBSI primarily influences technical interoperability by deploying blockchain technology for public services delivery. However, it also has an impact on semantic interoperability, given that common understanding of data and identities across the blockchain is essential. Its organizational interoperability comes in coordinating a network of nodes across EU nations for validating transactions and maintaining the blockchain.

B.1.3 Additional information on Open Voice Network

The Open Voice Network (OVON) is a non-profit association dedicated to advancing the value, credibility, and usefulness of voice assistance in daily life. It is focused on two initiatives:

- The **Open Voice TrustMark Initiative** is promoting and creating ethical solutions to make conversational AI worthy of trust — promoting ethical principles across focus areas of privacy and security, media and entertainment, health and wellness, and consumer education.
- The **Open Voice Interoperability Initiative**[21] develops specifications and software that enable conversational AI voice and chat assistants to collaborate, using a standard messaging API.

[21] Please find the link to the initiative: Interoperability Initiative - Open Voice Network and its github Open Voice Interoperability Initiative · GitHub

The **Open Voice Network Interoperability Initiative** aims to facilitate the seamless integration of voice and conversational AI in a manner similar to the functioning of the web. The envisaged future allows users to easily access and utilize any conversational assistant and language model that meets their needs, akin to browsing web pages. The path to this goal includes defining, developing, and promoting standards like an open, universal API. The OVON is defining a public API that will enable every conversational assistant that has implemented this API to connect, communicate, and transfer content and control with any other conversational assistant that has also adopted the API. This will allow any standards-abiding conversational assistant to effortlessly interconnect, communicate, and transfer content with other such assistants across platforms and language models. The initiative visualizes an ecosystem of standards-based, cooperating conversational assistants much like the web. These standardized assistants would enable users to conveniently switch between different assistants and language models according to their information needs, mirroring the ease of navigation among web pages.

Architectural patterns: Three architectural patterns of interoperability conversational interaction have been identified see the figures below.

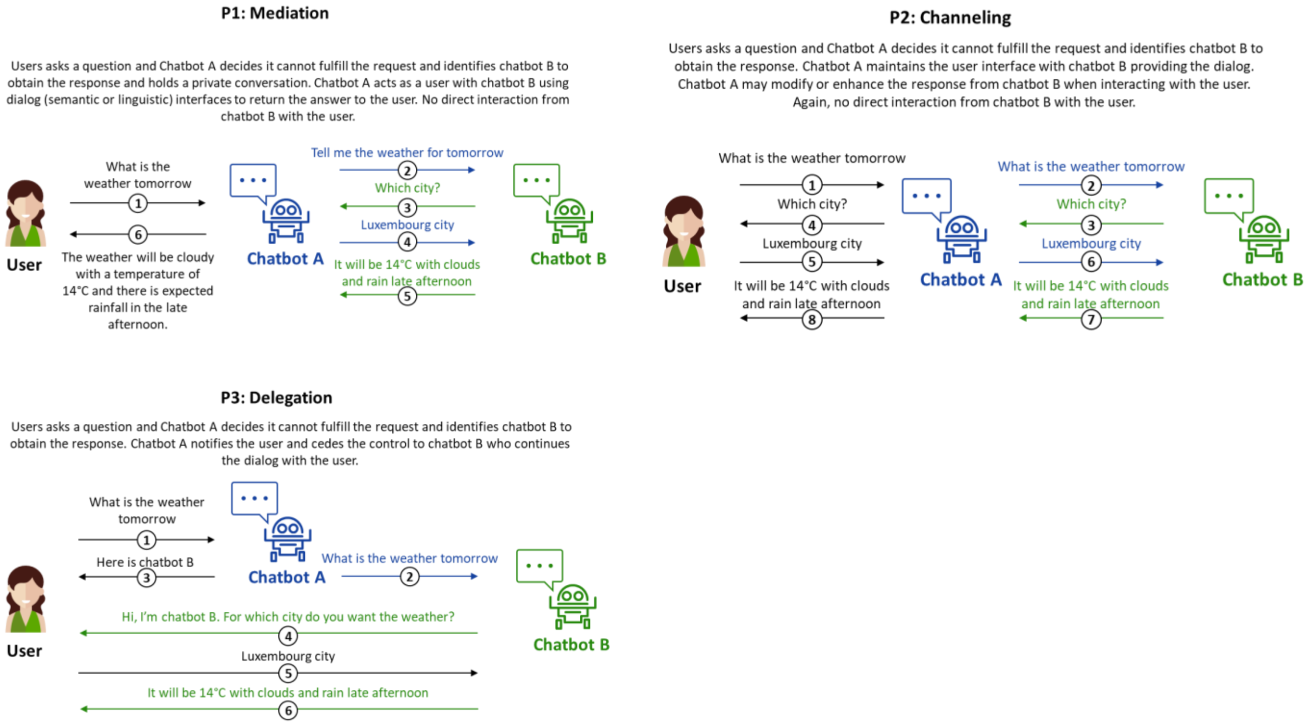


Figure 41. Architectural patterns outlined by the OVON

The difference between Mediation and Channelling lies in the interaction dynamics: For Mediation, Assistant A and Assistant B hold a private behind-the-scenes conversation, where Assistant A often already has most or all of the information needed from the user before reaching out to Assistant B for additional input, translating user requests and responses could be achieved. For Channelling, Assistant A acts more as an intermediary, continuously relaying and translating user requests and responses between the user and Assistant B, enabling ongoing dialogue and interaction.

Standard message formats: OVON indicates that Inter-assistant operability hinges on standard message formats like the Message Envelope and Dialog Events. The Message Envelope houses key conversation details, while Dialog Events comprise both the spoken or written exchanges and their corresponding metadata.

Standard development process: Standard protocols are essential for interoperability among conversational assistants from diverse vendors. These standards ought to be built and maintained openly, transparently, and jointly, in a way that's accessible to all, and that results in a lightweight approach.

The process of OVON to develop interoperability protocols for conversational assistants includes steps like:

1. Evaluating case studies of conversational assistant interoperability.
2. Publicizing requirements and specifications for universal review. Pertinent documents are published.
3. Hosting open webinars and demonstrations about ongoing and future work.
4. Making document and code repositories freely available. Experiential code is kept on platforms like GitHub.
5. Encourage developer participation. They have partnered with external developers which validate the specifications through implementation and testing. One example is the collaboration with Estonia's government to implement protocols in their Bürokratt government information system.

Future direction: OVON plans to investigate how users and conversational assistants can discover and locate conversational agents. Key information includes metadata detailing the assistant's usage and capabilities, URL indicating the assistant's name or location, and access instructions. These insights can be sourced from social media, assistant-associated websites, and a universal registry of interoperation assistants akin to the Domain Name System (DNS).

Simultaneously, security and privacy concerns like message encryption, trust establishment between interoperating assistants, data sharing policies, data ownership identification, user access rights, and adherence to the principles of the Open Voice Network Trustmark Initiative should also be addressed.

In our exploration of chatbot interoperability outside of Europe, we have discovered that there is a very limited number of documented cases. However, despite the limited availability of cases, we have encountered live and operating examples that highlight the practicality and effectiveness of chatbot/service interoperability in various regions.

While the interoperability landscape beyond Europe may not be as extensively explored, the existence of these operating use cases serves as a testament to the growing recognition of the importance of chatbot interoperability globally. As the demand for seamless communication among chatbots continues to increase, it is likely that we will witness the development of more standards and solutions in the future.

Following this overview of interoperability among chatbots, it is essential to look at the implications of these initiatives. The ensuing section will unpack the potential benefits, inherent challenges, and potential risks encountered or associated with inter-chatbot communication.

B2. Additional information of current state of interoperability

B.2.1 Overview of study participants

In this section, we have conducted a market overview of various public institutions in Europe that have implemented chatbots or have established interoperability between chatbots. Our approach involved reaching out to these institutions, inviting them to participate in our study, and share their experiences to contribute to the advancement of chatbot interoperability.

The table presented below showcases the chatbots that have been implemented or are currently in development among these institutions, highlighting the growing trend of integrating virtual assistants in government services.

Table 18. Overview of workshop participants

ID	Country	Institutions	Anonymized	Topic	Languages	Technology	Status
1	Austria, Belgium, Croatia, Greece, Latvia, Lithuania, North Macedonia, Poland, Portugal, Romania, Slovakia, Sweden	Pilot project for EU Fiscalis	Tax institutions across Europe	Tax	Official languages of each participating country	API, centralized database, translation service	Interoperable PoC coming soon
2	Belgium	Belgian Digital Transformation Office (BOSA)	Digital institution		Dutch, French	NLP intent-based chatbots (Presteria and IBM Watson), LLMs in the future	PoC in development
3	Estonia	Joint Agency of enterprise Estonia and KredEx	Business and Innovation organization	Living, working, and doing business in Estonia	English, Estonian, Russian	LLM & human-defined intents	Production
4	Finland	Finnish Immigration Service (Migri), Finnish Patent and Registration Office (PRH), Development and Administration centre for ELY Centres and TE Offices (KEHA Centre)	Three administrative institutions	Starting a business, immigration, looking for a job	English, Finnish	Intent based, Boost.ai	Pilot until 2024, now discontinued

ID	Country	Institutions	Anonymized	Topic	Languages	Technology	Status
5	Latvia	State Revenue Service	Tax institution	Tax	Latvian, English, Russian		Production Introduced 5 years ago
6	Luxembourg	Service central de Législation Luxembourg	Legislative institution	Law	French	Mistral	PoC
7	Luxembourg	List Luxembourg	Research organization	Not a chatbot, but a framework to automatically develop chatbots			Development
8	Poland	National Tax Administration	Tax institution	Tax			Production
9	Portugal	Agency of administration modernization	Administrative institution	Immigration	16 including Hindi, Bengali, Urdu	ChatGPT 3.5	Development
10	Portugal	Independent (Cabinet of justice)	Legislative institution	Justice	English	ChatGPT 3.5	Production

B.2.2 Full overview of interoperable chatbots

In this appendix, we will provide an overview of interoperability between chatbots projects that have been achieved or are being achieved. This will shed light on their functionalities, integration, and the level of interoperability in existence. This discussion will help to underscore the importance of interoperability in enhancing the effectiveness and efficiency of digital tools.

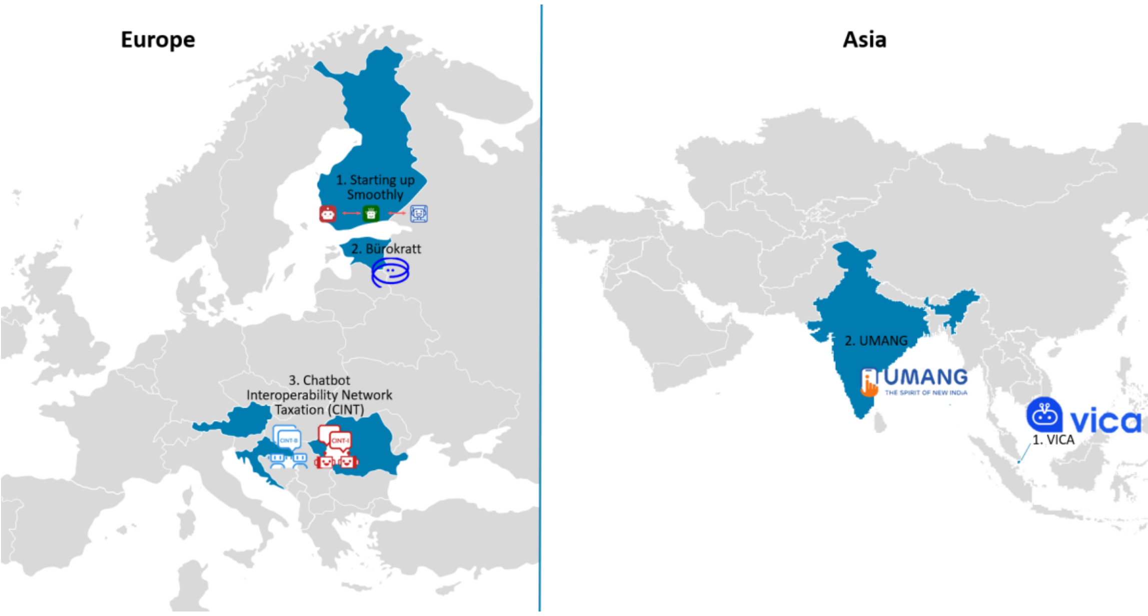


Figure 42. Chatbot interoperability across the globe

B.1.1.1 Interoperability programs implemented in the EU Public Sector

This appendix offers in-depth about three out of the four interoperability use case: *Starting up smoothly*, *Bürokratt*, and *CINT*. *Orchestrator implementation* is not included as no detailed information is available.

Case study 1: Starting up Smoothly

Project description
The purpose of the collaboration was to pilot a common chatbot service, a network of three chatbots, which could answer questions related to starting a company (PRH), taxation (Vero) and immigration to Finland (Migri). This project targeted foreign entrepreneurs in Finland, with main services that might be of interest to them (Miessner, et al., 2019). The goal is to have the user interact using only one interface to ask questions to all three chatbots. The AI decides which chatbot should answer the question best, depending on the topic of the question. In the chatbot network, each organization has its own chatbot and brain (dispatcher) setup. The brain holds the information about which other organizations the specific chatbot can refer the user to. The chatbot functions as the automated customer service, providing responses to user inquiries. To read the complete experiment conducted, please find the link Starting up smoothly study drafted by them[22] To guide the experiment, four research questions were drafted: A. Can we serve customers through a common channel? B. Should the customers be aware of organisational silos? C. How can we collaborate across organisational silos, budgets and resources? D. How to take another organisation on board in a chatbot network?
Target audience
The project named “Starting up Smoothly” was specifically designed to assist **foreign entrepreneurs** in initiating their businesses Finland. This explains the significance of connecting the selected three services, as it aligns with the project’s objective: • Starting a company (PRH)

[22] Starting+up+Smoothly+experiment+evaluation_CMYK.pdf (migri.fi).

- Taxation (Vero)
- Immigration to Finland (Migri)

The service is provided in English, allowing to reach a wider audience and directly addressing the needs of foreigners.

Timeline

The development timeline of the project lasted **12 months**, from March 2018 to March 2019, and was divided in two phases.

Phase 1: Chatbot network demo (March 2018 – June 2018)

Throughout the experiment phase, the focus was on actively involving two chatbots, Migri and Vero. The primary objective was to conduct a compelling demo to public servants at the end of the phase. To achieve this demo, three subphases took place: Discover (March-April), Define (April), Develop (April-June).

Phase 2: Chatbot network pilot (June 2018 – March 2019)

In the second phase, an additional chatbot joined the collaboration, PRH (August 2018). The goal of this phase was to pilot the common service of the three organisations to real users. By the end of this phase the interoperability between chatbot was available on the organisation's respective websites.

Technology

Each organisation has their own instance of a boost.ai-environment, which they oversee, with Migri providing the technology to all three organisations. The chatbots having the same technology facilitated the interoperability process, they did not however have the same instances.

The AI models are trained independently, and the three instances are only connected through an intelligent dispatcher (brain), which decides if a question is for one organisation or the other based on a probability threshold.

Interoperability setup and flow

Chatbot included in the interoperable network

Three institutions and chatbots were part of the project. All three chatbots can provide a larger number of answers that when functioning alone, thanks to the collaborative project effort. This harmonization ensures consistency and facilitates seamless interactions with the end user having one entry point for all the required information.

Institution	**Finnish Immigration Service (Migri)**	**Finnish Tax Administration (Vero)**	**Finnish Patent and Registration Office (PRH)**
Chatbot name	Kamu	VeroBot	PatRek
Languages	Finnish & English	Finnish & English	Finnish & English
Description	Answers migration related questions and was made to be friendly and answer questions to the point	Answers tax related questions and was made to be friendly and answer questions to the point	PatRek chatbot is able to answer questions related to setting up a company
Feature	Can use emojis selectively to make discussions more humane (never more than one in a row, positive emojis and country flags)	It can use emojis selectively to make discussions more humane (maximum one in a row)	NA

Figure 43. Starting-up smoothly

The bots all understand both Finnish and English however they have different designs, characteristics and features that are not available across the board. For example, Kamu & VeroBot can send emoticons (such as flags and friendly faces) to humanize the conversations, whereas PatRek only sends text. When working with interoperability, a mechanism to interconnect such similarities and differences should be considered.

How the interoperable network is managed

The “Starting up Smoothly” project used the concept of a brain as a dispatcher to facilitate interoperability.

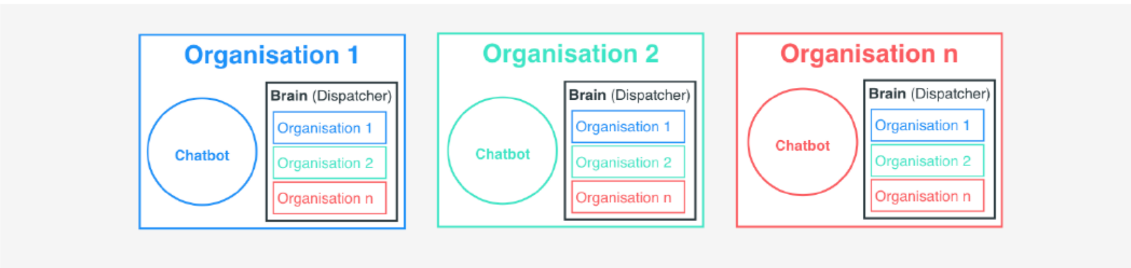


Figure 44. Multibot architecture overview (Miessner, et al., 2019)

When a user inputs a question into their device chatting with Kamu, the question is sent to Kamu’s dispatcher *(brain)*. The brain knows all other chatbots in the network and dispatches the question to all of them. In return, it receives a score from all connected chatbots indicating how likely each chatbot is to have an answer to the question. A probability threshold was set to decide which chatbot to transfer to. The brain decides which chatbot should answer based on the score. In case VeroBot is more likely to have a better answer, the user receives the suggestion to be transferred to VeroBot and after they agree the answer is sent directly to the chatbot of Vero (skipping the dispatcher), who then sends an answer back to the user’s device in the same interface. Keeping the same interface permits to preserve the conversation history and to avoid confusion for the user - the chatbot name and chat colour change to help them understand they are now interacting with a different chatbot. There the answer is displayed, and the user can react to it.

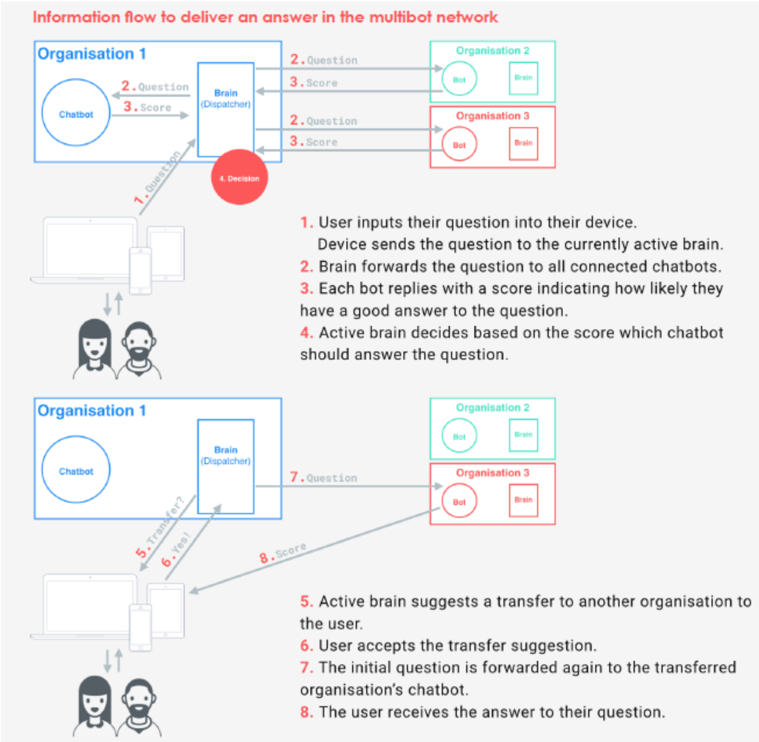


Figure 45. Multibot network answer delivery

Defining transfers between the chatbots

The goal of defining transfers was to understand how the chatbots interact with each other. Three types of transfers were considered:

- **Manual transfers** - Handovers where users explicitly ask to talk to a certain chatbot or organisation.
  Example: A user is on the Vero chatbot and has a question on immigration and asks to switch to Kamu from Migri. "I want to switch to Migri".
- **Reactive transfers** - Transfers where users ask about content that a different chatbot knows about.
  Example: A user after asking PatRek questions about setting up a company in Finland, asks about Taxation which is the expertise of Vero. PatRek offers to transfer to Vero. "What about taxes when setting up a company?"
- **Proactive transfers** - Handovers where the users are proactively given more information than they may expect.
  Example: A user that wants to set up a company is informed by PatRek that they will need to understand tax obligations in Finland and suggest transferring to Vero, the Tax administration bot.

Combining three types of transfers between the chatbots supports users to switch smoothly from one bot to the other, from one domain of knowledge to the next — and back. Being able to switch between the three organisations' chatbots in three different ways facilitates effortless flow of the user experience. From technical point of view, manual transfers were introduced first, followed by reactive transfers. Proactive transfers where the last types of handovers to be implemented. They were based on the user needs the three organisations identified together.

Testing

During the course of the project, several rounds of user testing were conducted to gain valuable insights into the perception and usage of the three chatbots by immigrants. This testing aimed to understand how users interacted with the services in real-life scenarios to improve both the content and technical capabilities of the chatbots.

The initial round of user testing took place in the early stages of the project and involved immigrants who had recently arrived in Finland. This was instrumental in obtaining feedback on the user experience and identifying areas for improvement.

A second round of testing was carried out prior to the pilot in September 2018. This phase specifically focused on testing the first version of PatRek's content and conversation logic, refining the system based on user feedback. In November 2018, two subsequent rounds of testing were conducted, encompassing all three chatbots and examining how they worked in conjunction. These rounds revealed a number of technical issues, but interestingly, it was observed that users did not hesitate to transfer between the chatbots. Additionally, positive feedback was received regarding the seamless transition between the different chatbot interfaces. To capture further insights and gather feedback on the ongoing pilot, an additional round of user testing was initiated in March 2019. This particular round aimed to understand potential development areas and gain an understanding of how users perceived the current pilot.

The research paper suggested several valuable recommendations for user testing, such as involving representatives from each organization during the testing process, jointly reviewing the results, and consistently testing with real users throughout the development cycle. This approach ensured alignment with customer needs and facilitated the incorporation of user suggestions to enhance content and answers.

Two main challenges were identified in this project.

- **Finding user testers:** The target audience being foreign entrepreneurs, so the goal was to recruit immigrants to test the solution which brought a more limited pool of potential user testers. User feedback was considered a crucial success factor for the delivery of this project.
- **Collaborating (organisational level):** Another "challenge" linked to collaboration was having to ensure consistent government lingo between the different institutions when writing chatbot reply-texts. The choice of words become even more challenging when there is no proper English translation available. One of the key learning was that maintaining consistency among key words and expressions is crucial for a smooth user experience and should be discussed with partners.
  Some other insightful key factors to succeed in inter-organisation collaboration were the following:
  - Arrange a time and/or place regularly (weekly/bi-weekly meetings)
  - Brainstorm together
  - Establish shared practices & channels to communicate
  - Maintain transparency for content work (share know-how)
  - Have a common goal

Results

During the period from November 2018 to June 2019, a significant amount of user engagement was observed with the three chatbots. Specifically, there were over **53,000 conversations** with the chatbots, which averaged around **240 daily chats**. Additionally, there were more than **3,100 transfers** made through the chatbot service. These results highlight the considerable usage and impact of the chatbot platform within the given timeframe. Furthermore, it is worth noting that two (Kamu and PatRek) of the three chatbots are still live and actively serving users. The most popular transfer being reactive (72% of all transfers), showing that users do not intently switch between chatbots but rather ask a specific question, followed by proactive (12% of all transfers).

To conclude the Staring-up smoothly study (Miessner, et al., 2019) addressed the research questions. The following is a summary of the key answers to the questions:

A. Can we serve customers through a common channel?

In summary, the research indicates that it is indeed possible to serve customers through a common chatbot channel. The qualitative data from the user tests revealed that customers were pleased to have access to a chatbot that consolidated information from three public agencies, considering it a modern and efficient service delivery model.

B. Should the customers be aware of organisational silos?

In considering whether customers should be aware of organizational silos, the experiment openly began with the assumption that customer awareness of these silos is necessary. As a result, the chatbot network was implemented in a way that displayed three different avatars, requiring customers to confirm when switching between chatbots. This approach aimed to prevent confusion and frustration for customers, ensuring they do not seek information from the wrong service channels or visit incorrect service points. Despite not exploring alternative methods, the experiment concludes that maintaining distinct chatbots is the recommended approach moving forward.

C. How can we collaborate across organisational silos, budgets and resources?

In terms of collaborating across organisational silos, budgets and resources, the project outlines several key factors for successful collaboration from their perspective.

Firstly, adopting a customer-centred mindset and aligning different perspectives towards a common goal is crucial. Secondly, openness to experimentation and trying new approaches is necessary for successful collaboration. A common physical workspace and shared digital tools facilitate effective collaboration. Building a collaborative team, with shared working practices and co-created methods, contributes to success. Clear leadership and plans, along with separate budgets through architectural design decisions, enable cross-organizational collaboration. While the project acknowledges that some aspects are still aspirational, it firmly believes that successful collaboration is achievable.

D. How to take another organisation on board in a chatbot network?

The onboarding process consisted of four phases: initiating the collaboration, starting the collaboration, defining and sharing team working practices, and ultimately defining the new chatbot. While acknowledging that there may be alternative approaches, future onboarding efforts would prioritize understanding the development culture and backgrounds of the new organization's team members to minimize confusion and maximize efficiency.

Interoperable network after the pilot

The original network kept working after the pilot, until VeroBot left the network in 2021. In 2022, Aino chatbot (owned by Development and Administration centre for ELY Centres and TE Offices (KEHA Centre)) joined the network. At the time, the target audience became wider: it contained foreigners willing to work in Finland, too. Interoperability was found very useful tool, but requirement to use the same technology didn’t make it simple to keep in use. In 2024, the project met its final end.

Case study 2: Bürokratt

The goal is to create an Estonian state virtual assistant. Bürokratt is an interoperable network of chatbots on the

Project description

website of public authorities that allows people to obtain information from these authorities through a chat window. It provides individuals, or users, with the opportunity to access direct public and information services using virtual assistants.

Target audience

Bürokratt, an interoperable network of AI applications, aims to provide **Estonian citizens** the ability to easily access public services using virtual assistants. It would be able to allow them to apply for passport renewal for instance and remind them ahead of time of upcoming actions required, similarly to Siri (Grzegorczyk, 2021). As for the languages, the tool used an open-source machine translation supporting seven languages: Estonian, Latvian, Lithuanian, English, Finnish, German, and Russian.

Timeline

Some of the key milestones of the project include:

- 2020: The vision and concept paper were drafted with the outline of the purpose, features and technical requirements. The first pilot project was also carried out.
- 2021: Development was focused on speech-enabled pilot projects, the machine translation and speech recognition features. Development of the core solution kicking off in the second semester of the year – including conducting 80 case studies in the public sector.
- 2022: Implementation of the chat function, by the end of the first quarter, citizens had access to the first public services available by selected agencies through the tool.
- 2024: It appears that the chatbot is currently inactive and not operational – the Bürokratt team needs to be contacted to learn more and implement the tool.

Technology

Bürokratt can be hosted in two ways, on the Government Cloud and locally on your own servers.

Bürokratt has integrated the following primary components (Gonçalves, 2022):

- Neurotõlge - open-source machine translation engine. Neurotõlge uses natural language processing techniques (NLP) to combine computational linguistics with statistical and machine learning models. Neurotõlge supports seven languages: Estonian, Latvian, Lithuanian, English, Finnish, German, and Russian.
- Neurokõne - text-to-speech tool. with speech synthesis sounds based on neural networks and trained by Estonian news resources. Developed by the Natural Language Processing research group of the University of Tartu, the tool can portray the natural sound and intonation of speech. Its repository is available on GitLab.
- Texta Toolkit - text analytics tool. used by public agencies to improve the efficiency of work processes and automatise routine activities through machine learning using MultiLingual Preprocessor (MLP) techniques. The source code for Texta Toolkit is available on GitLab.
- Kaldi - web-based speech-to-text tool that uses deep neural networks (DNN) and speaker diarisation models to segment and co-index audio recordings.

Interoperability set up and flow

Information not available.

Testing

Information not available.

Results

By the end of 2020, 41 AI solutions had been deployed in the government sector and companies of the private sector were also using AI systems to improve their own business processes. In November 2021, more than 30 organisations had carried out over 80 AI projects using these components. By the end of 2021, an alpha version - basic version of the software with limited feature - of Bürokratt was implemented in three agencies: the Police and Border Guard Board, the Consumer Protection and Technical Regulatory Agency, and the National Library. During 2022, the agencies were moving towards the beta version of Bürokratt, and citizens were able to have their messages forwarded and processed using AI technologies.

The source code of the chatbot is open source and available for access. However, it appears that the chatbot is currently inactive and not operational or that it can be installed as it is open source. For installation, cost related to installation, management, maintenance, technical requirements and so on need to be considered.

Case Study 3: CINT

Project description

The Chatbot Interoperability Network on Taxation (CINT) is an EU collaborative initiative. The objective is to provide comprehensive information and support in native language for EU individuals regarding tax obligations.

To reach this objective, CINT has the following goals:

1. providing guidelines on the business requirements necessary for the interconnection of chatbots by tax authorities at the level of the European Union, so that they have a set of recommendations to follow if, in the future, they decide to interconnect such a service;
2. identifying the optimal technical solution for the interconnection of chatbot systems that will be part of a specific bot network;
3. addressing information requested by national and non-resident taxpayers in a structured manner by documenting at the national level most addressed questions with the purpose of increasing tax compliance and offering taxpayer a seamless experience by using one service for receiving the requested information;
4. finding an optimal solution both for tax administrations that already have a chatbot and also for national tax administrations that do not have the service implemented and only having a frequently asked questions database with the purpose of addressing the different stages of development of e services in the digital era.

The project looks at two group perspectives:

a) CINT-B: Focusing on information for chatbots (e.g., questions and answers needed by taxpayers). They organize and catalogue data for universal use among all chatbots, ensuring systematization.
b) CINT-I: Focusing on technical challenges of interconnecting the chatbots, this includes language, platform differences and mitigating risks related to these challenges.

To reach interoperability between chatbots, a first collaboration linking twelve tax administrations' chatbots or databases will be undertaken.

The interconnection of chatbot systems of tax administrations can be beneficial from several points of view. Interconnected chatbot systems can streamline processes, making fiscal administrations more efficient. They can handle repetitive tasks, freeing up human resources for more complex issues. Such interconnection can provide better service delivery by offering personalized experiences for taxpayers based on the data the chatbot systems have access to. Relevant data about the taxpayers subject to different types of tax obligations can be obtained. This information may be used to prepare more detailed administrative guidance, or even design future normative reforms. Also, with the help of machine learning and natural language processing, chatbots can help in forecasting, risk management, as well as in other areas of tax administration. So, the interconnection of fiscal administrations' chatbot systems can lead to improved efficiency, better service delivery, and enhanced decision-making capabilities.

Target audience

The project seeks to provide assistance and support to EU citizens and more specifically taxpayers in order to exercise their fundamental freedoms. By catering to the specific needs and challenges faced by this target audience, the project aims to enhance their overall experience and ensure that they receive the necessary guidance and information pertaining to tax-related matters that are particularly relevant in each European country. In terms of languages, the goal is to incorporate a translation tool to be able to include as many European languages as possible in order to provide the content of the answer in the language required by the citizen via an automated way.

Given that the activity of the tax administrations is in a continuous process of modernization and adaptation to the economic and social realities in order to be able to meet the expectations of taxpayers (national and international) in terms of providing high quality and prompt services, one of the objectives of the tax administrations is the migration of taxpayers to the electronic environment. In this sense, tax administrations have developed over time a series of electronic services and new functionalities specific to these remote services, through which taxpayers can fulfil their tax obligations, without the need for physical presence at the units' offices. Now that the electronic services are implemented cross border interoperability is the next logical step for tax administrations and using the digital tools to offer support to taxpayers. The goal for tax administrations is to offer easy to use e services therefore the project was designed having in mind the user needs: to find out the desired information in an easy structured way, to address its common questions in an user friendly way because the user is never leaving the national chatbot in order to find out tax information about other European countries with the condition that the countries for which he is requesting tax information are part of the interoperable network.

The interconnection of the European taxation chatbots will provide support for taxpayers and so will further enhance the European Union's vision on the freedom of movement and the strengthening of trade facilitation by the continuous provision of the correct information to the interested citizens and companies on a daily basis.

The initiative offers the user in this case the taxpayers the opportunity to obtain the needed information from one place this being the national tax chatbot and also thanks to the use of e-translation tool the user has the possibility to obtain the answer in its own language.

Timeline

Key milestones:

- 2023: Kick-off of the project
- 2024: A PoC is expected to be completed in Q4 2024
- 2025: The project is anticipated to conclude in May 2025

Technology

Centralized hub

A centralised hub is required to ensure that the latest information is made available at low effort. This is because each country only has to provide and update the relevant information once in its own chatbot. The other chatbots can present this information to citizens immediately and without any additional personnel effort or lead times. The alternative would be decentralised maintenance of all relevant information for all member states in each individual chatbot. This would entail very high personnel costs and would lead to long lead times and accordingly to divergent and sometimes outdated information.

To ensure seamless availability of up-to-date information, a centralized hub is crucial. This allows each country to provide and update relevant information within its own chatbot, which can then be promptly accessed by other chatbots without human intervention or delays. Implementing a decentralized approach would result in higher costs, longer lead times, and potential disparities in information.

The centralised approach minimizes effort by having a single central hub instead of managing numerous remote chatbots for data retrieval and integration with very high personnel costs that would lead to long lead times and accordingly to divergent and sometimes outdated information.

A chatbot network with each State responsible for contributing information to the network would minimise risks of incorrect or misleading information and ensures each country maintains and enhances its chatbot.

Automated translation

Automate language would allow content to be provided in numerous languages. Experiences from countries employing commercial translation providers have shown promise. The EU translation service would be the preferred option to leverage in order to achieve this.

The use of the EU eTranslation tool for providing the answer in the desired language, solving in this way a common barrier among European countries that is the language, will conduct to an increased cooperation between tax administrations in Europe by offering a solution for taxpayer's assistance in an international manner.

Interoperability setup and flow

Chatbot included in the interoperable network

Twelve countries are currently involved for the development of the Proof of Concept (PoC): Austria, Croatia, Lithuania, Latvia (with chatbots), Romania, Portugal, Poland, North Macedonia, Slovakia, Belgium, Sweden (with databases) and Greece (both with chatbot and database).

While some tax administrations have implemented chatbots, others do not have its own chatbot. In this context the project had in mind a solution that will fit all tax administrations needs and developed a model that can accommodate both chatbots and data bases. In POC some countries opted to participate with chatbot, others that do not have its own chatbot opted to participate in PoC with database or in some cases where it was possible with a mix of both. Nevertheless, the existing chatbots, based on the developed API, have the potential to connect to databases.

How the interoperable network is managed

As previously mentioned in Technology, the solution considered is to have a centralized API requiring minimal changes to internal chatbot systems.

Defining transfers between the chatbots

No information yet available.

Testing

No information yet available.

Result

At its current phase, the project remains in its early stage with a Proof of Concept (PoC) that is nearing completion. With the completion of the PoC, the project will make significant progress and it will lay the foundation for future development and implementation involving numerous tax interoperability chatbots across Europe.

B.1.1.2 Interoperability programs implemented in non-EU Public Sector

This appendix will shift beyond the borders of Europe to examine the interoperability of chatbots in other regions. By doing so, we aim to broaden our understanding and uncover additional standards and solutions that exist in the global context.

Table 19. Overview of non-European interoperable chatbots or related projects

Chatbot name	AlphabotSG - Virtual Intelligent Assistant (VICA)	ChatUMANG (Unified Mobile Application for New-age Governance)
Institution	Singapore Government Technology Agency	Developed by: Ministry of Electronics and Information Technology (MeitY), National e-Governance Division (NeGD)

Description	• Singapore GovTech's next-generation Virtual Assistant platform
	• AlphabotSG is a super chatbot that learns from various agency chatbots, to help with a range of common matters and resources in just one place
	• When the user asks a question, it finds the best answer from the most relevant agency
	• If there isn't a confident answer by any agency, it gives you the best possible response using the context of the interaction and a new large language model
	• 40 supported knowledge bases
		• UMANG provides a single platform for all Indian Citizens to access India e-Gov services ranging from Central to Local Government bodies
		• Users can explore seamless access to many government services and schemes at one place, ensuring hassle-free and transparent experience for citizens
		• UMANG has integrated:
		    • 207 departments
		    • 1882 services

Technology	• Unified chat frontend for common branding across all government ministries and agencies • VICA is engine-agnostic and will be able to leverage the latest NLP technology to achieve better performance and accuracy • Hosted in GCC-AWS, SaaS Cloud Service for 60+ government agencies	NA
Chatbot status	Live AlphabotSG by VICA	Live UMANG - One App, Many Government Services

B.2.4 Challenges and Risks linked to the interoperability layers

Table 20. Challenges linked to the interoperability layers

Phase	Challenges	Mitigation actions	Interoperability layer[23]			
			Technical	Semantic	Organisational	Legal
Pla nni ng	**Infrastructural investment –** Requires significant technical	• Form collaborations to share expertise and infrastructure	X		X	

[23] Interoperability layers described in 0 In an era characterized by rapid technological advancement, the capacity for systems to work together seamlessly—termed interoperability—has emerged as a cornerstone of digital transformation. With the rise of generative AI, chatbots have jumped back at the forefront of user experience with generative AI assistants enhancing current intent-based chatbots. Interoperability to allow the exchange between different systems and a larger knowledge sharing is an essential capability for enabling chatbots, often developed by diverse entities, to communicate, exchange data, and use the exchanged information effectively. Interoperability holds particular importance for public institutions where the promise of streamlined communication, resource optimization, and enhanced public service delivery hinge on the robust exchange of information across various platforms.

This study delves into interoperability between public administration chatbots, including those operating across different knowledge domains. Public administration chatbots are virtual assistants designed to manage inquiries, disseminate information, and provide services to the public. These chatbots operate in various knowledge areas such as healthcare, education, transportation, and social services. As the drive towards digital governance intensifies, the necessity for these AI-driven tools to interact and share information seamlessly has become increasingly apparent. Effective interoperability between these chatbots would not only enhance service delivery but also ensure a unified citizen experience across different service domains which leads to increased user adoption.

The study is organized into several comprehensive sections to explore the subject in depth. First, "What is Interoperability" establishes a foundational understanding of interoperability, its types, and its significance in digital public services. Next, "Current State of Interoperability" analyses existing practices within public administration chatbots, including a review of existing chatbots working towards interoperability, as well as benefits, challenges, and risks associated with it. "Key Considerations for Interoperability" then addresses critical factors for success, encompassing technical, and organizational dimensions like key requirements for interoperability and UX/UI considerations. The section "Viable Approaches" explores strategic and technical solutions, offering insights from successful implementations. "Regulatory Outlook" examines the role of legislation and policy, providing an overview of current regulations and potential legislative changes. Finally, "Implementation Framework" presents a structured roadmap for public institutions to develop interoperable chatbot systems, detailing steps from planning and development to deployment and continuous improvement. By exploring these areas, the study aims to present a holistic view of interoperability in the context of public administration chatbots, offering valuable insights and actionable recommendations to foster a more integrated, efficient, and citizen-centric digital public service ecosystem.

The importance of chatbot interoperability extends beyond technical considerations; it embodies a shift towards a more collaborative and interconnected public sector. This collaboration can eliminate redundant information silos, thus promoting more accurate and timely responses to public inquiries. Moreover, interoperable chatbots have the

Phase	Challenges	Mitigation actions	Interoperability layer[23]			
			Technical	Semantic	Organisational	Legal
	resources and infrastructural investment to establish	• Use cloud-based platforms and open-source frameworks				
	Governance Models – Complexity in coordinating chatbot interoperability initiatives among different organizations	• Establish governance models and agreements to address data sharing, liability or dispute resolution • Facilitate communication through regular meetings, calls, workshops and having dedicated platforms/channels			X	
	Compatibility and information asymmetry – Incompatibilities and development challenges in diverse systems can cause incorrect responses due to varying platforms, protocols, and data formats (Elkhodr, Shahrestani, & Cheung, 2016).	• Encourage user feedback • Understand and investigate technical infrastructure • Choose chatbots with flexible or similar architecture for easier integration • Establish common protocols for seamless communication	X		X	
	Lack of centralized list of existing chatbots – Leads to missed opportunities for interoperability	• Encourage the development of a generalized database			X	X
	Maturity difference – Some institutions will have a fully developed chatbot while others are still working on theirs.	• Think of different ways of collaboration, i.e. maybe chatbot cannot interoperate			X	

potential to improve decision-making processes by aggregating data from multiple sources, thereby providing a more comprehensive view of various public administration functions. By establishing a standardized approach to chatbot interoperability, public institutions can ensure that they remain agile and responsive to evolving citizen needs. Furthermore, the study will highlight how successful chatbot interoperability can enhance citizen trust and satisfaction by providing consistent and coherent service delivery across different public sectors. Ultimately, this study seeks to underscore that achieving chatbot interoperability is not merely a technical goal but a critical component in the evolution towards a more efficient, transparent, and user-centric digital government. In the future, this progression can foster broader advancements, potentially paving the way for enhanced interoperability in various sectors.

What is interoperability:
Technical interoperability – Ensures functional communication of data between systems, servers, and applications.
Semantic interoperability – Involves the ability of systems to exchange data and comprehend data. This includes metadata (data about data) and data schemas that explain data organization.
Organizational interoperability – Concerns seamless collaboration between organizations. It involves the alignment of business goals, governance, processes and workflows, between different organizations.
Legal interoperability – Ensure that each organization operating under different legal framework can collaborate effectively without legal barrier.

Phase	Challenges	Mitigation actions	Interoperability layer[23]			
			Technical	Semantic	Organisational	Legal
		yet access to database can be shared				
Development	**Language barriers –** Dialogue translation of chatbots, especially low-resource languages and the lack of publicly available multilingual chat corpus (Gain, et al., 2022).	• Invest in AI translation technology or onboard translation company • Involve native experts in translation, particularly for culture-sensitive or subject-specific matters	X	X	X	
	Resource intensive – Building interoperability is time and energy consuming. The processing of data and network are resource intensive too, leading to potential budget issues.	• Determine optimal resource allocation • Align on budget within interoperable network	X		X	
	Question Redirection – A development challenge is effectively managing question redirection and corresponding answer display.	• Design clear decision-making algorithms for redirection rules of different questions • Interfaces should be transparent if user is redirected	X	X		
	Conversation Flow and Handoff – Managing flow, context retention and user experience consistently during cross-bot interactions has technical and design challenges.	• Define clear triggers for bot handoff • Keep users informed about the handoff process • Carefully design chatbot dialogues for coherent conversations, even when changing bots	X	X		
	Consent & session management – Obtaining user consent for data sharing across multiple chatbots, while managing session lifecycles for personalized cross-bot interactions.	• Continually test and monitor session management • Perform user testing to validate session management in chatbot interactions and gather improvement feedback. • Implement consent management mechanisms to ensure user control over information sharing	X	X		
	Error Handling and Recovery – Handling errors during cross-bot	• Implement a set of standardized error codes	X	X		

Phase	Challenges	Mitigation actions	Interoperability layer[23]			
			Technical	Semantic	Organisational	Legal
	interactions is challenging due to diverse chatbot error strategies and recovery capabilities.	across all chatbots. This will ensure uniform error handling across different chatbots. • Align on recovery strategy				
	Multilingualism – Lack of publicly available multilingual chat corpus (Gain, et al., 2022). This presents a challenge for chatbot interoperability in dialogue translation of chatbots with different languages, especially low-resource languages.	• Use eTranslation tool (will become open source in the future) • Consider native experts to be involved in translation	X	X		
Testing & Maintenance	**Lack of data protocol –** This is necessary to ensure data is up to date and maintained	• Focus on developing unified data standards and protocols & centralized repositories			X	
	Security and privacy – Chatbots might handle sensitive user data. Ensuring secure data sharing and regulation compliance can be complex when multiple chatbots interoperate.	• Make sure to use secure communication protocols, robust authentication and authorizations mechanisms, and anonymise user data if present	X		X	X

Table 21. Risks linked to the interoperability layers

Phase	Risks	Mitigation actions	Interoperability layer			
			Technical	Semantic	Organisational	Legal
Planning	**Inadequate interoperability standards** – Absence of universally accepted interoperability standards and guidelines creates complexity in uniform data handling and communication among chatbots.	• Follow existing standards such as the EIF where available • Establish industry standards and guidelines for chatbot interoperability			X	X

Phase	Risks	Mitigation actions	Interoperability layer: Technical	Semantic	Organisational	Legal
	Regulatory – Compliance is necessary, e.g., take into consideration legislative framework at European, national level. Aspects such as transparency also need to be taken into account.	• Conduct legal assessments to understand all relevant laws and regulations • Schedule regular audits and compliance checks			X	X
Development	**Misalignment of Responses -** Uncoordinated chatbots can produce contradictory responses. Lack of context may lead to repetitive or incomplete information, affecting the user experience.	• Define clear protocols and standards for consistent chatbot responses • Use context transfer mechanisms to keep user context during switches between chatbots		X		
	User experience inconsistencies – Poor interoperability may cause inconsistent branding and user experiences across platforms, leading to trust issues and limited chatbot capabilities.	• Establish banding & design guidelines • Clearly communicate the limitations to users and provide alternative options or fallback mechanisms to overcome platform-specific restrictions.	X		X	
Testing & Maintenance	**Data breaches –** Cross-system data exchanges may compromise sensitive data security due to weak integration, leading to potential unauthorized access or malicious activities (Hasal, et al., 2021).	• Utilize strong encryption methods and implement access controls and secure transfer protocols • Conduct security audits to minimize vulnerabilities	X			X
	Quality differences – Disparities in commitment and interest levels among different parties.	• Establish contracts such as SLA to outline quality expectations and responsibilities			X	
	Governance – Think about how interoperability will be maintained in the future	• Form a governance body comprising all participating institutions			X	
	Dependence on vendors / service providers – Chatbot interoperability could increase vendor dependence, limiting	• Contractual provisions can be used to limit/ prevent strong independence on vendors	X		X	

Phase	Risks	Mitigation actions	Interoperability layer			
			Technical	Semantic	Organisational	Legal
	the ability to switch vendors without putting network at risk.	• Make use of interoperable API which is agnostic of technology where possible				

B3. Additional information: Key considerations for interoperability

B.3.1 Pros and cons of chatbot types

This table provides an insight regarding the pros and cons of different types of chatbots.

Table 22. Chabot types : Pros vs. Cons

	A. Rules-Based Chatbots / Retrieval-Based Chatbots	
	B. AI/ML powered Chatbots	
	Voice Enabled Chatbots	
	Pros: • Hands-free (beneficial to people with certain disabilities, such as dyslexia or physical disabilities preventing people from typing)	**Cons:** • Might misinterpret spoken language, especially with accents or background noise • Not appropriate for crowded environment
	Context-Aware Chatbots	**Generative Chatbots (NLP)**
Pros	• Understand the context of conversations • Can provide a conversational experience similar to human interaction	• Able to generate human like responses • Can handle a wide variety of requests
Cons	• Needs vast amount of data and advanced programming to function correctly	• Can provide incorrect or nonsensical responses (hallucination) • Requires large resources to train & run

B.3.2 Licensing and contracting details

In this section, we will delve into some of the legal aspects of interoperable chatbots, including licensing agreements and contract terms. We will list the main points that are important to consider when setting up a contract for interoperability. We will go over each necessary document, explaining what each point entails and why it is important to consider.

A. Pre-contractual

1. **Non-Disclosure Agreements (NDA):** These are often used, during the pre-contractual phase, to ensure that sensitive and confidential information that may be shared between the parties during formal or unformal discussions/negotiations, are identified as such and should not be disclosed to any third parties, unless clearly authorized by law or duly circumscribed in the NDA. NDAs typically outline the types of information that are considered confidential, the duration of such confidentiality undertaking, the obligations of the receiving party to ensure due care when dealing with said information, and the permitted uses or disclosures of the confidential information. Note that an NDA does not offer any protection (such as a patent would do), but clearly identifies the information of confidential nature which should be treated as such and shared between the parties during this pre-contractual period. It aims to, for example, prevent unauthorized disclosure to third parties.

B. Onboarding

2. **Project Agreement:** Moving onto the contractual aspects, it is paramount to clearly define the scope of the services (bearing in mind the more concise the scope is the better), the role and responsibilities, and expectations of all parties involved in the Contract. To ensure smooth operations, there is also the need to have a clear strategy to ensure all parties will commit and satisfy their duties and define terms for dispute resolution should one party breach any of its obligations under said Contract. In addition to these, it is prudent to agree on review and termination clauses upfront to enable the smooth exit of the Contract and to circumvent any potential future complications. This Contract should also outline what works need to be done by which party, timelines for completion, and related costs. Other mandatory aspects included here are deliverables milestones and payment terms, and any assumption(s) where applicable.
3. **Service Level Agreement (SLA):** This defines the level of service expected by the customer from a supplier, laying out the metrics by which that service is measured, and the remedies or penalties, if any, should the agreed levels not be achieved. The SLA should cover availability, response times, incident resolution, and other key performance indicators that are critical to the success of the chatbot deployment.

C. Data Privacy

4. **Data Processing Agreement:** This agreement determines the rights and obligations of the parties involved in the processing of personal data (if any) performed as a part of the as a part of the interoperability efforts. It determines how data shared between parties should be handled, the subject matter, scope, purposes, duration of the processing etc. It plays a crucial role in ensuring compliance with applicable legal requirements regarding data protection. The data sharing agreement may also address data ownership, data security measures, data subject rights, such as the right to access, correct, or delete personal information, transfers of data from the EU to third countries (if any) and any other matter, as required by the applicable data protection legislation, especially the General Data Protection Regulation (GDPR). This is especially important in the case of interoperability as data sharing is an integral part of it, so data exchange conditions should be clearly defined between the involved parties.
5. **Data Protection & Privacy:** Data protection and privacy rules and principles are essential aspects to consider, especially in maintaining compliance with the GDPR as will be further explained in section 1.6. This becomes particularly important when there is data sharing or transfer between different institutions potentially located in different jurisdictions. It is crucial to take measures to safeguard the security of data processed as a part of the interoperability efforts and to make the processing transparent to individuals. Similarly, respect for privacy and data protection rights of individuals whose personal data is being processed should be ensured throughout all data collection, observing data minimization principle (i.e. to limit the collection of personal information to what is strictly and directly relevant and necessary to reach a specific purpose), processing, use, and sharing activities conducted as part of the interoperability efforts.
6. **Privacy notice:** A privacy notice is a statement describing how a website or business collects, uses, stores, and shares personal information. It fulfils a legal requirement to protect a customer or client's privacy.

D. Other Terms

7. **Intellectual Property Rights (IPRs):** In the EU, the relationship between IPRs and software interoperability plays a significant role. The first European Interoperability Framework (EIF), endorsed by the European Commission in 2009, is a guiding light for institutions providing digital services, aiming to facilitate interoperability within public service delivery (European Commission, New European Interoperability Framework: Promoting seamless services and data flows for European public administrations, 2017). The EIF tends to favour the use of open standards and specifications regarding IPRs and has been updated in 2017 in hopes to further facilitate interoperability. An example of this is

the underlying principle 2 of the EIF which is about openness of data, specification and software. The recommendations associated with this are recommendations 2, 3, and 4, but recommendations 32 and 33 are other examples of this.

Recommendation 2: Publish the data you own as open data unless certain restrictions apply.
Recommendation 3: Ensure a level playing field for open-source software and demonstrate active and fair consideration of using open-source software, taking into account the total cost of ownership of the solution.
Recommendation 4: Give preference to open specifications, taking due account of the coverage of functional needs, maturity and market support and innovation
Recommendation 32: Support the establishment of sector-specific and cross-sectoral communities that aim to create open information specifications and encourage relevant communities to share their results on national and European platforms
Recommendation 33: Use open specifications, where available, to ensure technical interoperability when establishing European public services

These recommendations relate to semantic and technical interoperability and emphasize the importance of using open specifications.

8. **Copyright:** The EU Software Directive provides some allowances with copyright law, stating reverse engineering to achieve interoperability is not an infringement, given specific conditions are met (Parliament & Union, 2009). The directive specifies four conditions in which reverse engineering is allowed. First, the acts of reproduction of the code should be indispensable to obtain necessary information to achieve interoperability of an independently created software program with other programs. Second, the reverse engineering should be performed by a person with a right to use a copy of a program or on behalf by an authorized person. Third, the information which the reverse engineering aims to obtain should not have been given to the concerned persons beforehand. Fourth and finally, reverse engineering should be confined to parts of the original program that are necessary in order to achieve interoperability. Besides these conditions, the directive makes it clear that the obtained information from reverse engineering can't be used for anything other than to attain interoperability. This means that any reverse engineering done outside of these conditions or with the goal to develop and sell a similar program infringes on copyright and is therefore prohibited. Reverse engineering can play an important role in interoperable chatbots as it can help to understand the functioning and structure of existing chatbots which could then aid in re-constructing it or customising it to be compatible with different platforms or bots. This could allow developers to identify ways to make additional chatbots interoperable or enhance their interoperability.

Having a clearer understanding on some of the basic requirements to consider for interoperability between chatbots; dataset considerations, conversation functionalities, licensing and contracting, we now turn our attention to another important point for interoperability, both technically and culturally, namely the impact of multilingualism. The ability of chatbots to communicate in multiple languages and incorporate low resource language could greatly enhance accessibility but introduces additional challenges in ensuring efficient interoperability.

B.3.3 Approaches and techniques in language processing and translation

Language translation can be achieved using two main categories of approaches:
Indirect approaches and techniques offer methods of enabling communication across languages using an intermediary language or step
Direct approaches apply translation techniques (e.g., machine learning techniques, or machine translation services to process natural languages) directly from source language to the target

B.2.3.1 Indirect translation approaches

Pivot languages are used to overcome language resource limitation. The pivot translation approach involves using a third language, rather than a direct translation between two languages, especially when bilingual resources are limited (Paul, Finch, & Sumita, 2013). This method initially translates the source language into the pivot language using statistical translation models trained with source-pivot language resources. Subsequently, the pivot language translation is translated into the target language using a second translation engine trained on the pivot-target language resources. Taking into account the languages to be employed as pivot languages for interoperability, selecting English and French as pivot languages appears to be suitable for the current phase. English being a high resource language with a large number of available resources, it is often chosen as a pivot language (Paul, Finch, & Sumita, 2013). This approach facilitates translation between languages with no available bilingual resources, for example Finnish and French in Figure 46. A potential drawback would be that the translation quality may decline in this two-step process, a small mistake in the first step could lead to an error in the target language (Zaiets, 2021). Previous pivot translation research typically chose the pivot language based on the availability of bilingual language resources and the language similarities between the source and the pivot language (Paul, Finch, & Sumita, 2013).

Figure 46. Pivot languages

Human translation involves the process of converting or interpreting text or speech from one language to another by a person rather than by an automated system. This method ensures accuracy, captures cultural nuances, and respects the original text's style and tone. Moreover, it can handle complex sentences and intricate meanings that machine translation might miss. When it comes to interoperability between chatbots, human translation could play a crucial role in ensuring the correct and appropriate responses. Despite the advancements in machine translation, there might be dialects or idioms that machines cannot or have difficulty translating properly, causing misinterpretations or communication gaps, hence reducing interoperability. Human translation can help fill these gaps, enhancing the functionality and interoperability of chatbots across languages and cultures. Looking at the drawback, human translation is a very time-consuming process. Another disadvantage could potentially involve the lack of consistency in terminology when involving multiple translators. Additionally, human translators may make error or emit biases (or on the contrary mitigate bias). Lastly, comes the cost of hiring professional translators.

B.2.3.2 Direct translation approaches

There are a few distinct methods to move from one language directly to another:

Natural Language Processing (NLP): The advancement in Machine Learning enables chatbots to decipher user input using NLP. This allows chatbots to understand, respond and learn from conversations, providing a context-driven output without the need for predefined replies (Adamopoulou & Moussiades, 2020). However, as already mentioned in Low resource languages, processing low-resource languages, with limited training data may lead to reduced translation accuracy. Despite this limitation, ongoing progress in NLP and ML shows promise in improving translation capabilities. Techniques such as transfer learning and multilingual model can be useful in increasing NLP performance for languages with less available data.

- *Cross-lingual transfer learning* is a technique where a pre-trained model is used as the starting point for a different but related problem – the idea is that languages share certain similarities in their structure (Teo, 2021). For example, a model trained on English can be used as a base to train a model on another language. This technique leverages prior knowledge in combination with new information. Several transfer learning methods can be used depending on the training data available. One method could be translating the training data when no language data is available (Schuster, Gupta, Shah, & Lewis, 2018). Models should perform better when the source language and the target language belong to the same family language, for example German, Dutch and Danish belong to the Germanic Family, and Portuguese, Italian and Spanish to the Romance family (Kim, Kim, Sarikaya, & Fosler-Lussier, 2017).

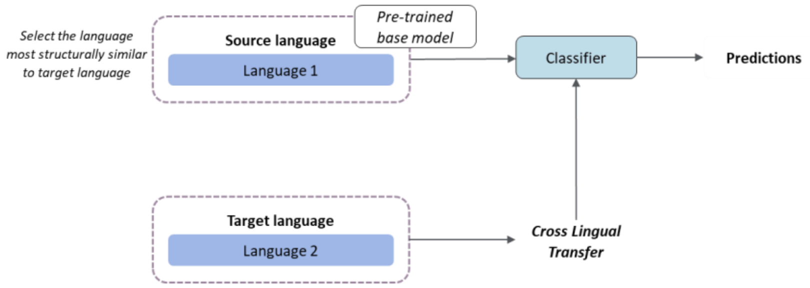


Figure 47. Cross-lingual transfer learning overview

- *Multilingual model* is a technique where a single model is trained on multiple languages (Laumann, 2022). It will learn the commonalities and differences. This capability can enhance a chatbot ability to generate responses in multiple languages, including low resources.

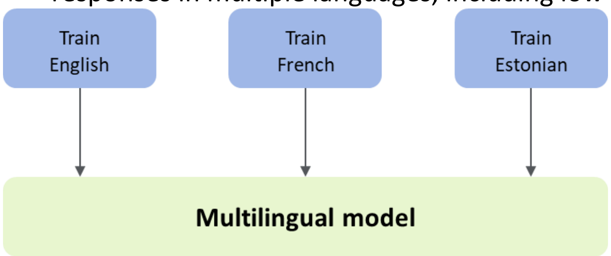


Figure 48. Multilingual model overview

Machine Translation regards the use of software to translate text (or speech) from one language to another. To address the challenge of data scarcity, one method is to integrate a monolingual chatbot with a machine translation engine (Wołk, Skowrońska, & Skubis, 2021). The two prominent trends in Machine Translation (MT) are Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). SMT, albeit less precise, demands fewer resources, while NMT models, which tokenize the text bodies into subword units, are more compact and accurate.

- *NMT* is an approach to automated language translation that relies on deep learning models, specifically neural networks, to predict the likelihood of a sequence of words, typically modelling entire sentences in a single integrated model. This approach enables the model to use the entire context of a sentence to predict the next word in a sequence, which yields better results compared to older phrase-based models. NMT models can be trained for specific language pairs (e.g., English to French), but they can also be constructed as multilingual models. An example of this is Google's Multilingual Neural Machine Translation System, which uses a single system to translate between multiple different languages (Tsang S.-H. , 2022).
- The BiLingual Evaluation Understudy (BLEU) score is the most widely used metric for evaluating machine translation, as it reasonably correlates with human evaluation. BLEU scores range from 0 to 1, where 1 signifies that the translated output is identical to the reference but 0 indicates it is entirely different.

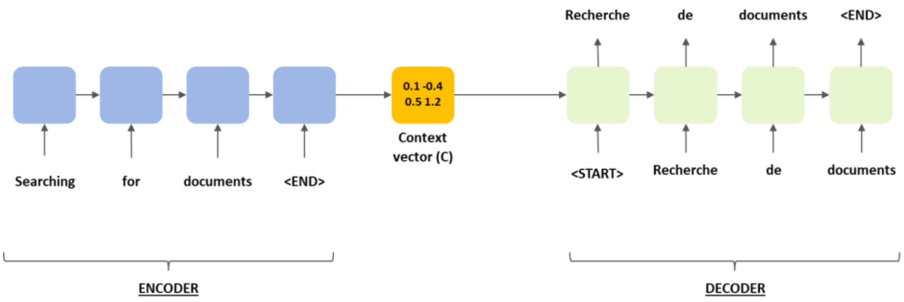


Figure 49. Neural Machine Translation overview

Large Language Models (LLMs): In the current stage LLMs marked a turning point in chatbot evolution. They are capable generating human-like text. These models are trained on a massive corpus of text, which equips them with the ability to translate and comprehend numerous languages, predominantly high-resource ones. For example, ChatGPT supports a wide range of European languages such as, but not limited to, Spanish, German, French, Estonian, Finnish, Greek, Irish, Italian, Latvian, Lithuanian, and more (Botpress, 2023). Meanwhile, LLMs performance with low-resource languages can be inconsistent due to less data availability. If sufficiently trained, they could potentially provide more reasonable translations and conversations. Continuous advancements in these models might eventually improve their efficiency with low-resource languages.

When considering NLP for low-resource languages, one can choose between using cloud-based models or developing a custom model. Custom NLP models could offer a targeted solution by providing specific training on limited data sets, potentially improving understanding and handling of such languages. However, looking at the drawbacks, the development and training of these customized models might necessitate substantial resources and domain expertise (Šoštarić, Pavlović, & Boltužić, 2019), making the process time consuming and expensive. On the other hand, using APIs of existing cloud-based NLP models could be a more convenient alternative, for simplicity and ease of implementation.

The application of Translation APIs, such as those provided by Google Translate or Microsoft Translate, can enhance a chatbot's multilingual capability. These resources offer broad language support and credible translation quality. For instance, Microsoft's Translation API supports various languages and is free up to a specific monthly limit. Translation APIs facilitate simple text translation between two languages, usually bi-directionally (Church, 2018).

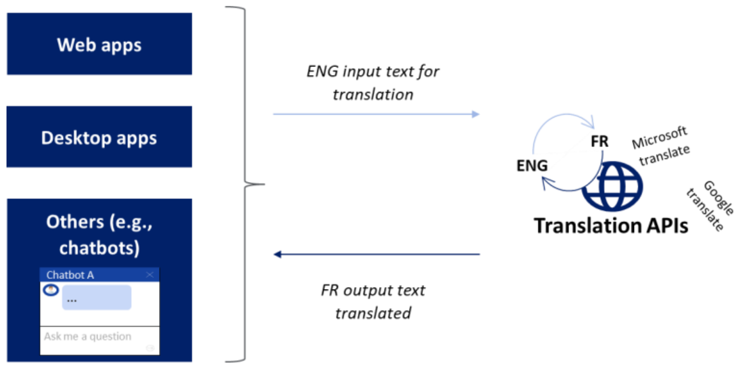


Figure 50. Translation API

The number of resources available, to a chatbot, for a language can significantly impact their output quality. Especially when transitioning from high resource languages, with abundant available data, to low resources languages. When assessing the quality of a chatbot’s response in low resource languages (this can also be applied for high resource languages), collaborating with native speakers associated with that language could be a viable option. This is because they might have previous experience in adding a low-resource language to a chatbot. An alternative could be to engage a professional translation firm to assess the bot's translations. This approach would also aid in understanding cultural contexts. For instance, the concept of social security might not exist or be understood in certain countries, making the term meaningless even when translated correctly into their language.

Multilingualism plays a significant role in enhancing interoperability of chatbots, several considerations have to be explored such as low resources languages, and different languages operation between chatbots which add layers of complexity to the interoperability. Looking ahead, our focus will shift from these language considerations to analysing the impact of UX/UI design on chatbot interoperability effectiveness.

B.3.4 API types and protocols

Types of APIs

APIs can be categorized based on their use cases, including data APIs, operating system APIs, remote APIs, and web APIs (Goodwin, 2024).

- Data (or Database) APIs: Facilitate the connection between applications and database management systems.
- Operating System (Local) APIs: Define how applications utilize operating system services and resources.
- Remote APIs: Specify how applications on different devices interact with each other.
- Web APIs: Enable the transfer of data and functionality over the internet using the HTTP protocol.
    - Most used APIs are Web APIs, the four types of web APIs are:
        - Open APIs: Public and accessible via HTTP, with predefined endpoints and formats.
        - Partner APIs: Connect business partners, requiring onboarding and login credentials.
        - Internal APIs: Private APIs used within an organization to improve internal productivity.
        - Composite APIs: Combine multiple APIs into one call, useful in microservices architecture.

API protocols

The rise of web APIs has led to protocols, styles, standards, and languages that define rules, data types, commands, and syntax. These API specifications standardize information exchange (Goodwin, 2024).

Simple object access protocol (SOAP) and representational state transfer (REST) are different approaches to API design. SOAP is a protocol, while REST is an architectural style defined by constraints. Both use HTTP for information exchange (Goodwin, 2024).

- REST: Lightweight, flexible, and easy to use, requiring less code. Supports multiple formats (plain text, HTML, YAML, XML, JSON). Often preferred for public APIs due to its simplicity and versatility.
- SOAP: A protocol that's more deterministic and robust, with built-in compliance and type checking. Often considered more secure and suited for applications with strict data integrity needs. Only supports XML format.

Choosing between REST and SOAP depends on the use case, with REST favoured for its support of multiple data formats and ease of use, while SOAP offers enhanced security and robustness (Goodwin, 2024).

B4. Additional information on UX/UI considerations

B.4.1 Redirection details

Table 23. Redirection return possibilities comparison.

	Option A: Host bot provides answer from contributing bot	Option B: Host bot redirects user to contributing bot's interface	Option C: Host bot hosts contributing bot in same interface	User interruption	Chatbot interruption
User prompt – If the user does not know the specific intent to trigger the return, the bot can simply ask to go back to the host bot after a certain period of time. This can also be actively asked by the user			X	X	
Add a button – Provide a clearly labelled button, such as "Return to Bot A" to easily navigate back.		X	X	X	
Usage notice – Inform users from the start about the presence of multiple bots.	X	X	X	X	X
Command alias – Create a simple command like "/home" that users can use to instantly return to the host bot. Can be combined with previous option.			X	X	

B.4.2 Media support

Some chatbots might support rich media inputs like images and voice messages while others might be limited to text-based only interactions as seen in the example of the Figure below. This difference could limit the information exchange. For instance, a chatbot designed to send photo responses will face issues on a platform that only supports text messages.

The simplest way to overcome this is to ensure that the type of intents facilitated from the host bot to the contributing bot only contains the type of input readable by the contributing bot. This can be set up in the interoperability design phase (the bots will know what media types are supported by each other and only communicate in the relevant ways). In option B below, it is clear that the Chatbot B does not support uploaded documents, and therefor Chatbot A extracts the relevant information from the document to make it consumable for chatbot B to provide the relevant answer.

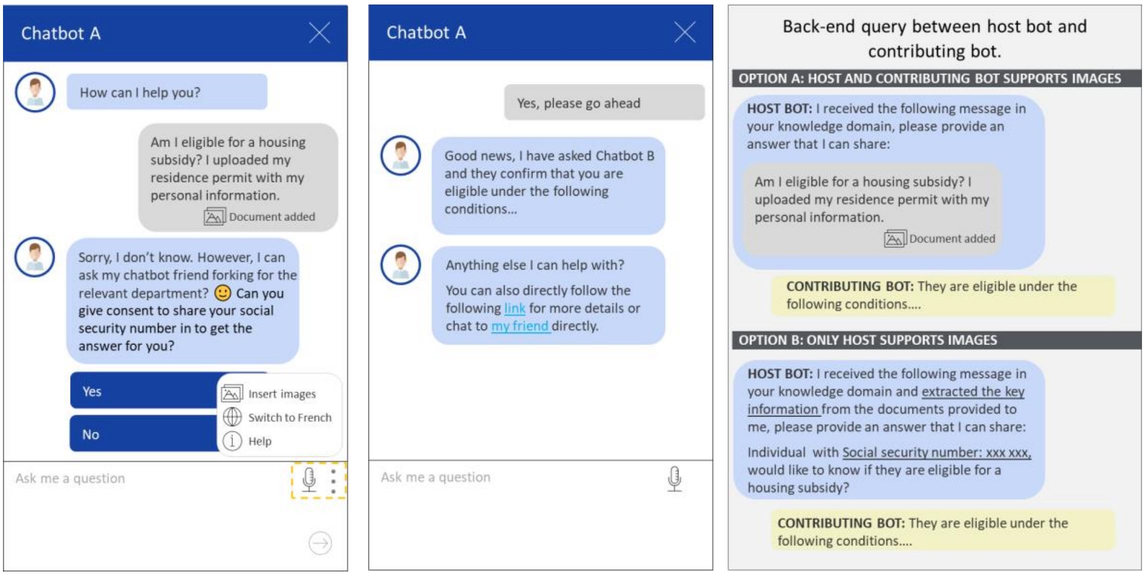


Figure 51. How to facilitate different media support within single bot interface

In the case of interoperability facilitated through switching interfaces between different bots (or hosting a contributing bot within the host bot's environment), there are additional considerations. To prevent a lack of response from the chatbot on unsupported media types, an effective solution could be to disable unavailable features when they switch to another chatbot. Rather than removing the features, which may disorient users familiar with those functions from the previous chatbot, greying them out may provide a better user experience and understanding (Pacheco, 2022).

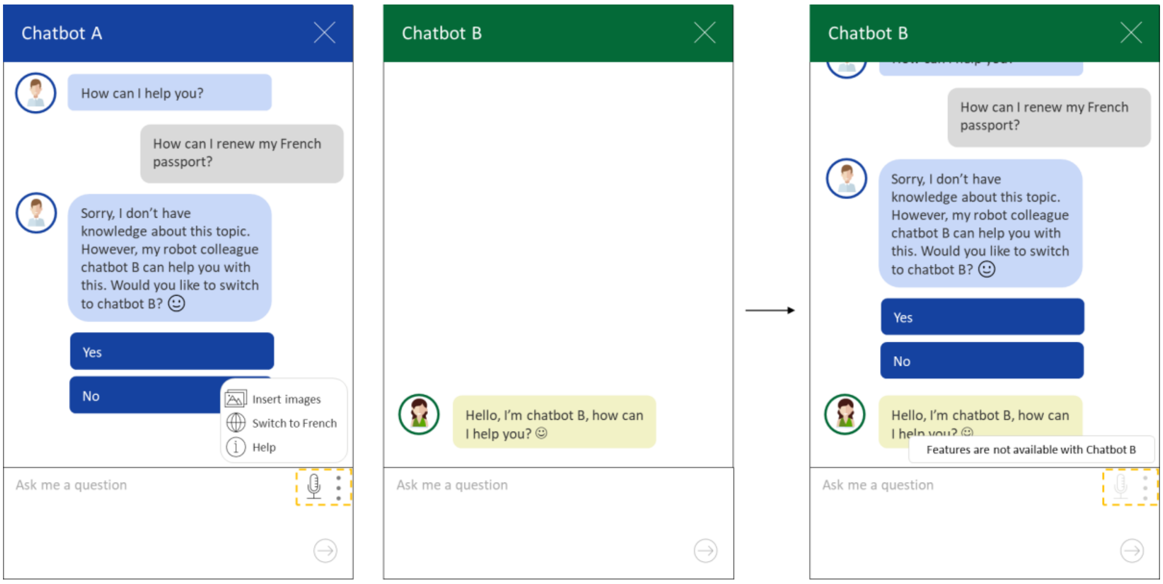


Figure 52. How to facilitate different media support between different bot interfaces

B.4.3 Interface differences

All examples demonstrate how the main chatbot could effectively handle redirection by leveraging the capabilities of other chatbots. This ensures that users can access the most relevant and accurate information, even if it resides in different chatbot systems. The visual representation of the chatbot might differ greatly between chatbots, size of interface, and more, see Figure 53. Chatbots interface differences

However, this should not affect interoperability as long as the response content is appropriately retrieved from the contributing chatbot and displayed within the host bot.

In the Interoperability display for Option A & Option C the host bot is contributing in the same interface, and thus the user will experience the same display, however in option B, the user is redirected and could have a different interface that might inhibit user experience if UX/UI best practices are not implemented across the board.

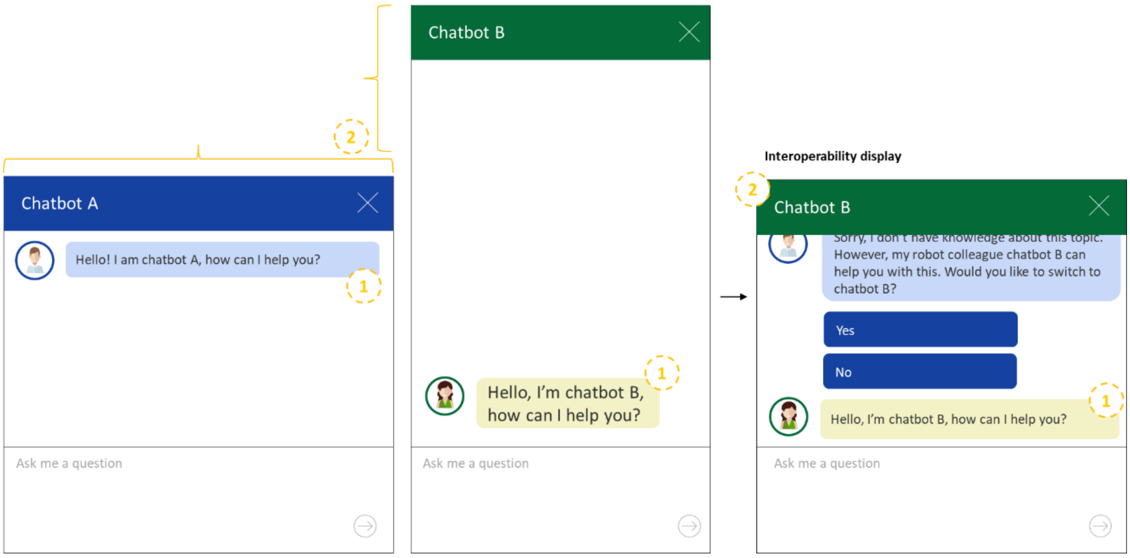


Figure 53. Chatbots interface differences

B.4.4 Disclaimer options pros vs. cons

The table below will delve into the advantages and disadvantages of the different options that have been showcased. Understanding the distinct strengths and limitations of the different options will allow to make more informed decisions about their application and which could be a better option to consider.

Table 24. Pros and cons of disclaimer options

	Option A: interoperability consent & source	**Option B: interoperability source**	**Option C: Start conversation with consent**
Provides transparency: Users are informed of interoperability	X		X
Maintains chatbot consistency: Distinguishing chatbots so users can recognize which chatbot they are interacting with	X	X	X
Requests consent: Asks user for their approval to switch	X		X

	Option A: interoperability consent & source	**Option B: interoperability source**	**Option C: Start conversation with consent**
chatbot and share conversation history			
Seamless transition: Redirecting users directly can result in a smoother conversation flow	X	X	X
Acknowledge source: Credit is given to the chatbot, and link is provided to switch to the website of other chatbot	X	X	
Visual clarity: The way the information is provided makes the chatbot that provided the information clear.	X	X	X
Immediate notice: The user is notified about the interoperability at the time it occurs, allowing users to understand what they are giving consent to.	X	X	

B5. Example of intent & entity mapping

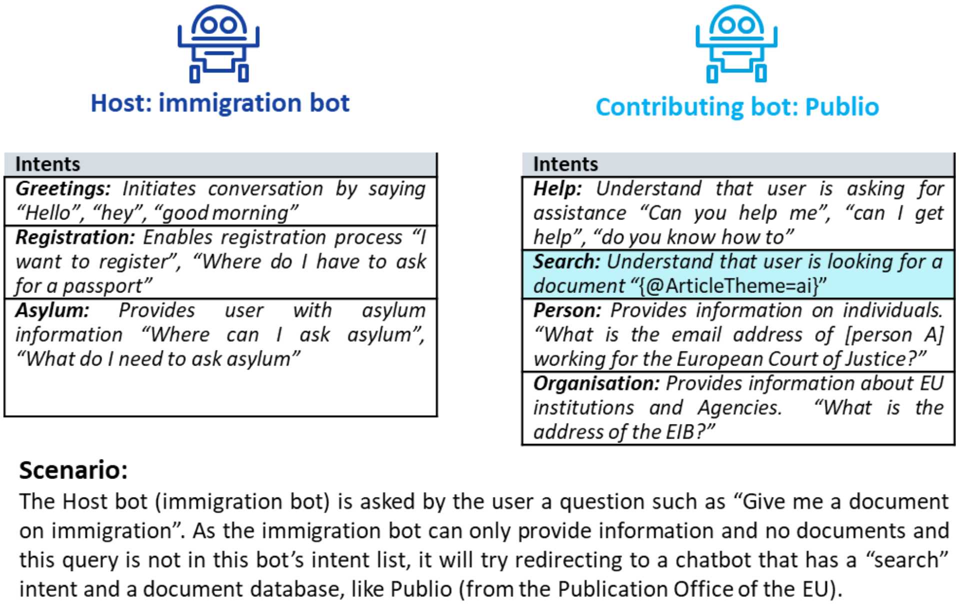


Figure 54. Example of intent & entity mapping with a scenario

In the context of interoperability, Capello et al. explained Bot Data Catalogue Application Profile (BotDCAT-AP) which enables the description of intents (actions users want to accomplish via chatbots) and entities (information units related to those intents) supported by a dataset and its access method. Designed to enhance chatbot interoperability, BotDCAT-AP standardizes dataset descriptions, easing access and reuse of external knowledge sources across various chatbots. This benefits both dataset owners, who can enable broader data utilization, and developers, who can more efficiently build and integrate applications (Cappello, Comerio, & Celino, 2017). BotDCAT-AP supports multiple access points like REST APIs, SPARQL endpoints, and SOAP-based web services, facilitating efficient retrieval and integration of data, which streamlines the development process and supports robust chatbot systems.

B6. Risk classification of the AI Act

The EU AI Act is based on a risk-based classification of AI systems as well as obligations regarding general-purpose AI which have to be considered in case some of the chatbots in the network are LLM-based. The different risk categories are as follows:

- **Unacceptable risk** are AI applications that pose a clear threat to the safety, livelihoods, and rights of individuals and as such, will be outrightly banned. This includes, but is not limited to, AI that manipulates human behaviour, exploits vulnerable groups, or uses 'social scoring' by governments.
- **High-risk** systems include a broad range of AI applications, such as those that enter public spaces or potentially impact an individual's legal status or rights.
- **Limited risk** systems include AI systems such as chatbots. For these systems, transparency obligations apply. For example, users should be aware they are interacting with a machine.
- **Minimal risk** represent the majority of AI systems, such as AI-enabled video games or spam filters. There are no specific requirements under the AI Act for this category.

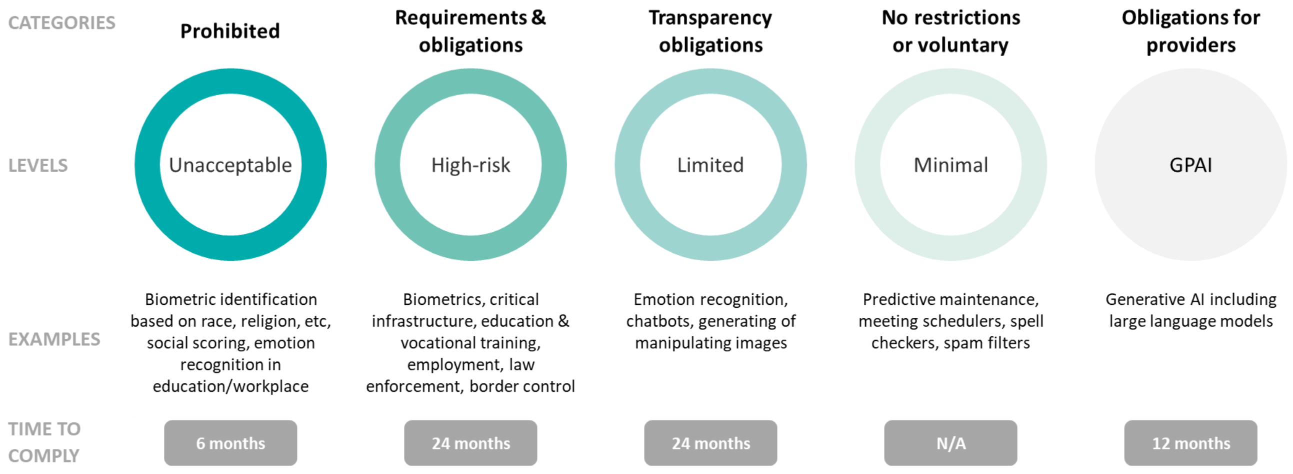


Figure 55. Overview of the AI Act

B7. Example of consent display options

In case consent-transfer between chatbots are needed for interoperability, the above guidelines should be followed. Below are presented two options, refer to Figure 56.

- Example A – Layover consent: Users are only asked once at the beginning of the conversation if they consent to their personal information being processed (if relevant in the case a chatbot is able to process data). This consent would be asked again when connecting to another bot.
- Example B – In-text consent: Users are only asked consent before they are required to provide personal information for a query they have asked, and the consent would be asked directly in the conversation.

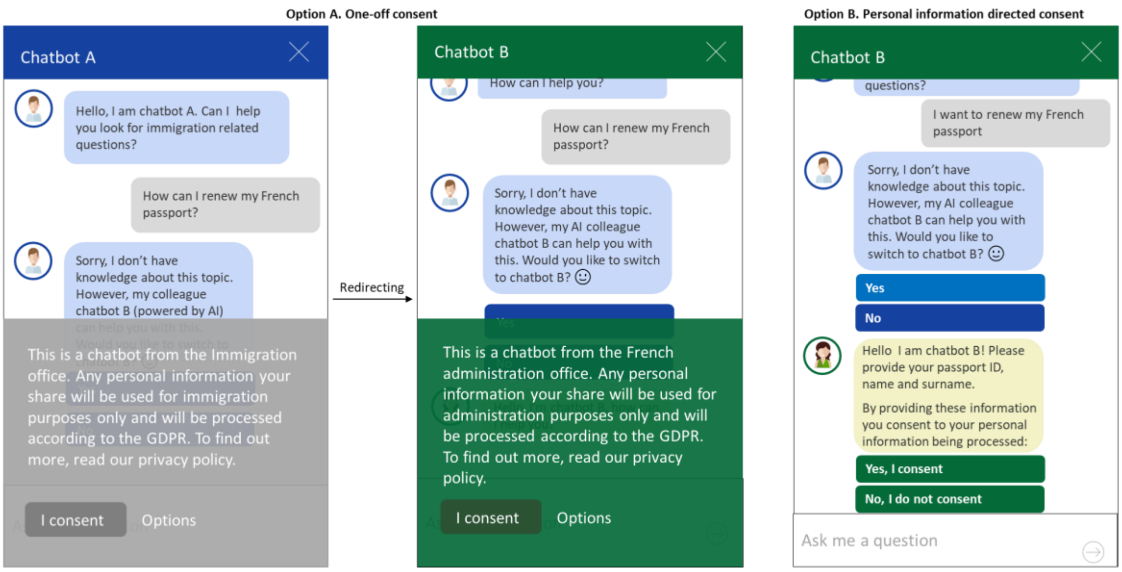


Figure 56. GDPR interoperability consent examples

B8. Overview of the potential deliverables of the implementation framework

The table below presents the anticipated deliverables to be generated over an interoperability between chatbot project. Each deliverable outlines the scope and content it should include, these remain examples and can be enhanced as the project evolves.

Table 25. Detailed overview of the potential deliverables

Phase	ID	Deliverable	Description (purpose)
Phase A	D1.1	Functional design plan	Plan outlining the functional infrastructure for deploying the chatbots, this includes the necessary functional design for interoperability, conversational flow & trigger mechanism (vendor selection, UX/UI elements, directional flow, requirements, intent, etc.). This also include key features list comparing different possible features to include in the interoperability, covering: • List chatbots considered (with their capabilities, topics, etc.), the user based targeted, the potential benefits for both chatbots • Chatbot and interoperability features to consider between the chatbots, the intents, inappropriate topics (to be excluded by both chatbots), conversational flow, list of languages of the bots • This should be done considering D1.3 user stories.
	D1.2	Technical architecture plan	Plan outlining the technical infrastructure to set-up for the PoC development (data structure, translation layer, routing interactions, host solutions, configuration, architecture diagram, etc.)
	D1.3	Testing plan	Document describing: • The requirements (functional and non-functional) prioritization by importance of tasks and issues to oversee in the interoperability between the chatbots. • The epics and related user stories linked to the epics, including the personas • Acceptance criteria & DoD: list the acceptance criteria (e.g., response speed, conversational robustness) and the DoD to be reached to consider the PoC complete and move to the production deployment.
	D1.4	Contractual documentation	These should cover project contract, NDA, SLA, Data Processing Agreement, Protection & Privacy, IPR, Copyright and consider regulations such as AI Act and GDPR. Legal document defining the roles, responsibilities, and liabilities of all parties involved from the development to the monitoring of the chatbots as well as the regulatory compliance.
	D1.5	Project plan	Document detailing the project work plan with the timeline for the PoC development, the key milestones, the risks and mitigations, foreseen meetings and stakeholders involved. This will include a RACI matrix presenting a table with the roles and responsibilities of stakeholders from the different organizations of the chatbots that must interoperate.
Phase B	D2.1	Working PoC system	The complete working system should reach the "Definition of Done" defined and the acceptance criteria set.
Phase C	D3.1	Sprints reports & test logs	Reports documenting the tests achieved, the defects identified, their prioritization and fixes applied as well as the scope of the next sprint.
	D3.2	Sign off completion certificate	Signed off completion certificate by stakeholders providing validation of UAT (all test validated in regard to the acceptance criteria and DoD). This certificate is necessary to move to the production deployment.

Phase	ID	Deliverable	Description (purpose)
Phase D	D4.1	Deployment pipeline	Document detailing the steps of the deployment pipeline from the necessary components, environment, additional testing to the final approval.
	D4.2	Live interoperability chatbot	Fully functional interoperable deployed chatbot in production that can interoperate with at least another chatbot on the designated platform.
	D4.3	Monitoring reports	Compiled report containing conversational, feedback and interoperability metrics on the chatbots.

B9. Implementation framework: Templates for Phase A – Initiation

B.9.1 Functional / non-functional requirements description

When investigating interoperability between chatbots, it is crucial to understand and identify the functional and non-functional requirements that will serve as foundations in the development, deployment and maintenance of the system. Many of these will feed to UX/UI considerations or how domains of knowledge are shared which are topics covered in sections 1.4.3 and 1.5.1. Below are some examples of functional and non-functional requirements, note that additional requirements should be added and adapted to the project goals.

General requirements (yellow)
Specific requirements (blue)
Interoperability requirements (red)

Table 26. Example of functional & non-functional requirements[24]

	ID	Requirement	Description
Functional requirements	F1	Monitor	Monitoring ensures everything is working as expected - it is important in maintaining system integrity, performance and accuracy. Metrics help evaluate the system such as response time, resource usage, error rates, error tracking, active users, feature usage, etc. *Example:* • *Administrators can access a dashboard showing the number of active users, common queries and the chatbot's response times*
	F2	Response provision	The chatbot(s) should be able to answer user queries. *Example:* • *If the user asks a question such as "What documents do I need to apply for citizenship", the chatbot(s) should be able to answer the user input.*
	F3	Command execution	The chatbots should be able to understand and execute commands or requests from the user. *Example:* • *The chatbot should be able to create and assign a ticket to a relevant department if a user requests technical support.*
	F4	Context handling	Ability to handle the conversation context when interacting with multiple systems. The chatbot should be able to pick up the conversation where it left off when switching between systems. *Example:* • *If a user asks following questions: "Where can I find tax information for Luxembourg?" and then "What about Slovenia?" the chatbot should understand the context and provide information about tax.*
	F5	Data / knowledge exchange	The chatbot should exchange data with other systems, pulling in necessary information to answer user queries. *Example:* • *When asked about the status of a visa process, the chatbot should fetch real-time data from the institution's database.*
	F6	Multilingualism	The chatbot might support multiple languages to cater to a diverse audience, in case multilingualism bots are in scope. It should have sound language processing capabilities, including translation and understanding. *Examples:* • *A user in France can ask the chatbot questions in French, and the chatbot will understand and respond in French.* • *A user in Germany can also interact with the same chatbot using German and still receive accurate responses.*

[24] Not all will be relevant, and the interoperability case might require additional ones

	ID	Requirement	Description
	F7	Error handling	Should a chatbot not respond, error messages should be issued. Offering a returned response and gauging satisfaction can ensure the system learns from its tested responses. Ability to identify and handle errors effectively, for instance to resolve or reroute the chatbots pathway seamlessly. *Example:* • *If a user makes a typing error or asks a question that the chatbot doesn't understand, it should guide the user to provide a valid input.*
Non-functional requirements	**NF1**	Cost-effectiveness	The system should provide optimal functionality and efficiency at a reasonable expense (this threshold should be defined). *Example:* • *The bot should optimize its use of compute and storage resources to keep operational costs to a minimum.*
	NF2	Response Time	Chatbots should be able to respond to user queries within acceptable time limits. *Example:* • *If a user asks the chatbot a question, they should receive a response within a few seconds, regardless of the number of users interacting with the bot.*
	NF3	Performance	Corresponds to how well the chatbots interact and if the correct chatbot responds to the user query. *Example:* • *The user query should be sent to the most relevant chatbots to answer it given a certain threshold is met.*
	NF4	Maintainability	The system should be easy to maintain and update, i.e. finding and fixing issues, updating components, or adapting the system. *Example:* • *If the company decides to add a new feature or improve an existing one, developers should be able to accomplish this without causing downtime or bugs.*
	NF5	Security	All data exchange between the chatbots must be secured to maintain privacy and confidentiality. This can be done through encryption to help prevent data breaches. *Example*: • *If a user shares personal data, such as an email address or phone number, the bot should securely store and handle this data.*
	NF6	Scalability	The interoperability functionality should be scalable to include future chatbots. *Example:* • *If two chatbots have established interoperability, it should be easier to include another chatbot.*
	NF7	Usability	The chatbots connection should be user-friendly and intuitive for users. *Example:* • *Users should find it straightforward to interact with the chatbot, with clear prompts for input and easily understandable responses.*
	NF8	Consistency across devices	Compatibility of chatbots to work on different devices and platforms. *Example:* • *Whether a user is interacting with the chatbot on a desktop or mobile device, or via a website or a messaging app, the user experience should be consistent and error-free.*

B.9.2 Epics / User Stories overview

<u>Step 4. Define Epics</u>

Epics are large bodies of work that can be broken down into a number of smaller tasks, namely user stories. Based on the functional and non-functional requirements outlined earlier, related requirements will be grouped into categories, or "epics". Table 27 below is an example template of what epics can look like in the context of interoperability.

Table 27. Examples of epics

Epic ID	Epic Name	Related requirements
1	Rich and dynamic user engagement	Multilingualism, Context handling, Command execution
2	Advances operational intelligence	Monitoring, Error handling
3	Secure and compliant information handling	Data/knowledge exchange, Security

<u>Step 5. Create User stories</u>

User stories help us understand the user perspective. These are created based on user personas, which are fictional representations of main user types, and are assigned to an epic. First, user personas will be defined, then narrative-based scenarios from the perspective of each persona will be written. These personas and stories will help design tests that resemble real-world use of the product.

<u>User personas</u>

User personas represent fictional characters based on actual users and their behaviours, needs, goals, attitudes and pain points. The idea is to use these personas to guide design/test decisions by providing a realistic representation of the key audience that will use the chatbots. The use of personas helps in comprehending with the user's needs, facilitating the creation of more user-friendly solutions. Profiles need to be considered also on the different features that we would like to include in the chatbots. The personas should take into account what topics your chatbot covers, which languages it supports and why they would use your chatbots. Figure 57 and Figure 58 below show some examples of general user personas as well as one more detailed example of a user persona.

				
User type	Non-EU citizen	Recent graduate	Non-tech-savvy user	Business owner
Description	I'm currently located in India but I'm planning to relocate to Germany for work. I need information about the visa process, tax rules, and healthcare services in Germany. I'm also curious to know about the local culture, housing, and cost of living.	I've recently completed my Masters in accounting. Currently, I'm searching for job opportunities in companies across the EU. I'd like to understand more about the job market trends, the wage standards, and the credentials transfer process in different EU countries.	I'm not very comfortable using advanced technology services. However, I want to understand the latest EU policies and social welfare programs. I also need to know how to avail of state services in my country, Spain. I hope these chatbots can assist me in a simple and intuitive manner.	I run a small tech firm in France, and I am considering expanding my business to other countries, particularly Italy and Spain. I need information on EU business regulations, tax laws, and trade rules in these countries. Also, I would like to know whether there are any government programs that could support my expansion.

Figure 57. Examples of user personas for different chatbots

User targeted

Recent graduate

Description

I've recently completed my Masters in accounting. Currently, I'm searching for job opportunities in companies across the EU. I'd like to understand more about the job market trends, the wage standards, and the credentials transfer process in different EU countries.

Goals

- Wants to kickstart her career in a reputed tech company within the EU.
- Hopes to understand the job market trends across the EU.
- Wants to explore internship opportunities and entry-level positions in different countries.
- Needs to understand the wage standards for entry-level tech roles in different EU countries.
- Needs to know the process and requirements for credentials transfer among different EU countries.

Tech usage

High

Attributes

- Driven and ambitious
- Adaptive and open to new experiences
- Tech-savvy

Figure 58. Detailed example of a persona

<u>User stories</u>

User stories help us understand the user perspective. These are created based on user personas, which are fictional representations of our main user types, and are assigned to an epic. After defining user personas, like we have done in the previous section, narrative-based scenarios can be written from the perspective of each persona. These personas and stories will help design tests that resemble real-world use of the product. Table 28 below shows an example template of user stories which are based on the user personas created in the previous section.

Table 28. Examples of user stories

Epic ID	Epic Name	US ID	User story
1	Rich and dynamic user engagement	1.1	As a Recent Graduate, I need the chatbot to execute commands accurately when I search for job market trends across the EU
		1.2	As a non-tech-savvy User, I need the chatbot to support my native language, Spanish, when I ask about the latest EU policies.
2	Advances operational intelligence	2.1	As a Recent Graduate, I expect the chatbot to monitor my search history and provide personalized results during the conversation.
		2.2	As a Business Owner, if there are errors during the chat, I need the bot to handle them effectively to avoid wasting precious time.
3	Secure and compliant information handling	3.1	As a non-EU citizen, I want my personal data to be handled securely and without any breaches when I ask about visa requirements.
		3.2	As a non-tech-savvy User, I need assurance that my personal details are not shared or misused when I ask about social welfare programs.

B10. Implementation framework: Templates for Phase B – PoC development

Develop specific trigger mechanisms and conditions based on user interactions: Program triggers to respond to specific user actions or inputs. As previously seen with the Starting up smoothly use case in section B.2.1, three scenarios were seen where a trigger mechanism could switch the conversation to another chatbot, see below:

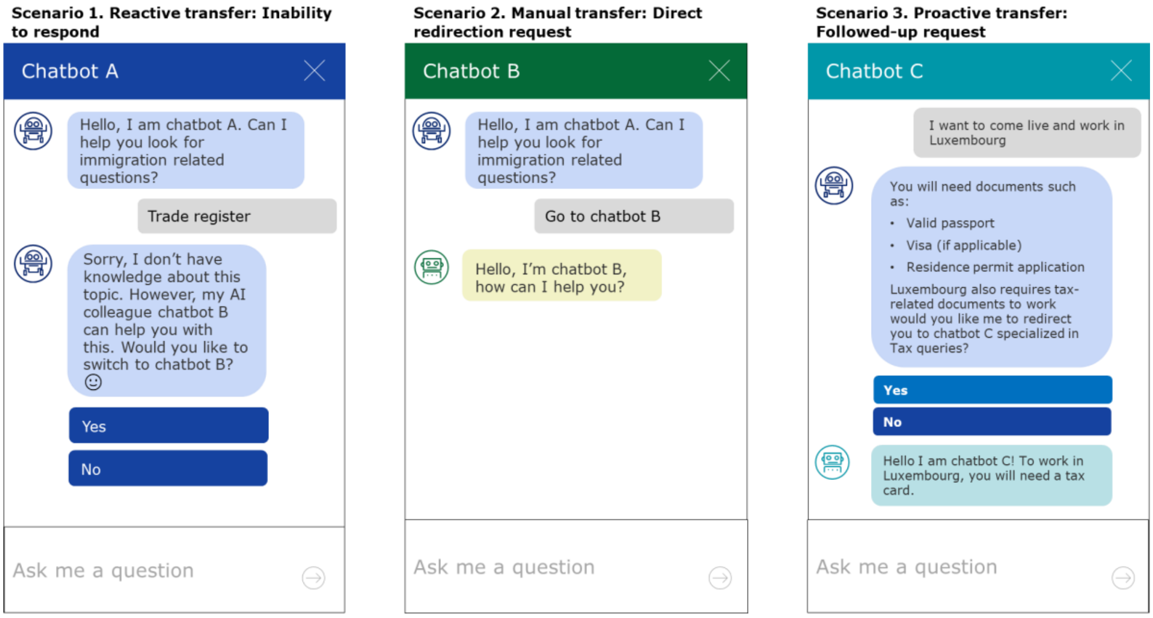


Figure 59. Trigger mechanism to redirect

Scenario 1: The chatbot cannot answer the answer has it is not part of their intent; therefore they propose to switch to a bot which would be able to answer the request.
Scenario 2: The user has knowledge about chatbot A being connected to other chatbots and directly requests to change to another bot.
Scenario 3: The user is given extensive information that extends with another chatbots knowledge and is asked if they would like to switch.
The goal is to design the paths along which requests and responses flow between different systems. This is like defining the rules around when and how messages get passed between both chatbots as seen above. The rules could be based on specific user needs, keywords, or other criteria. To determine which chatbot can answer the question the best, a threshold should be set.

THRESHOLD EXAMPLE: Taking two of the chatbots above, Chatbot A: Immigration, Chatbot C: Tax.

If we take a threshold of a 60% match with the list of keywords as the trigger to determine which chatbot should handle a user query.

- If a user asks, "What documents do I need for immigration to Luxembourg, and what are tax implications?"

The system analyzes the query, and the keyword matching might look like this:
For Chatbot A (Immigration):

- Keywords: ["documents", "immigration", "Luxembourg"]
- Match: 3/3 ---> 100%

For Chatbot C (Tax):

- Keywords: ["tax", "implications"]
- Match: 2/2 ---> 100%

Both chatbots match 100% with their keywords scope, but because the conversation started on Chatbot A's platform (immigration chatbot), it would answer the immigration part of the question first and then ask the user if they would like to continue with Chatbot C for tax-related inquiries, since tax-related keywords were present in the query.
In different scenarios where the match is not 100% for both, the query would be handled by the bot with the highest match, given it crosses the threshold. For instance, if the query was "What is the immigration process?", Chatbot A would handle it as it crosses the 60% threshold while Chatbot C doesn't.

B11. Implementation framework: Templates for Phase C – Testing

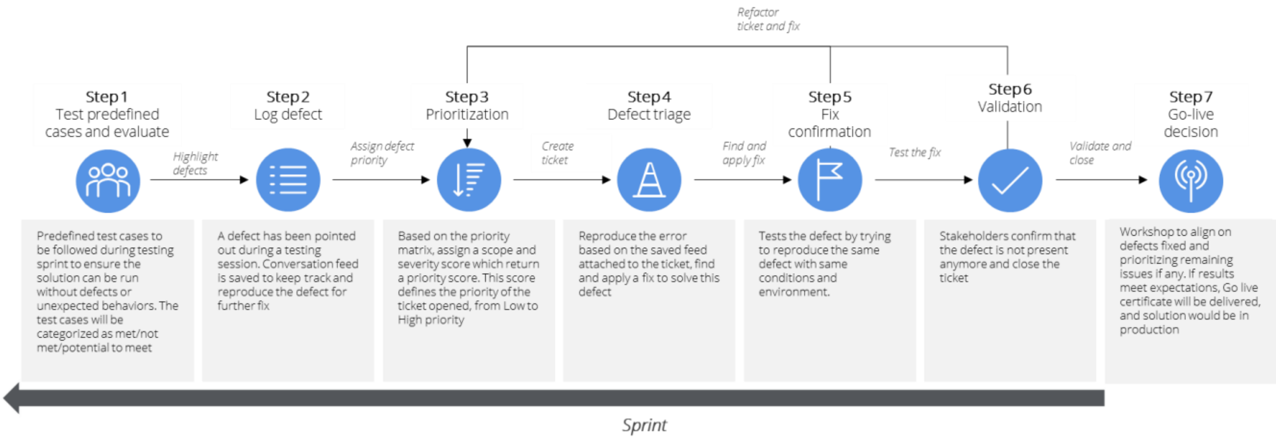


Figure 60. Steps to conduct testing

Zoom in Prioritization (step 3)

To optimize the UAT process, create a prioritization matrix. The matrix will serve as a tool to identify which requirements are of most importance and need to be tested first. This is usually based on the business value, risk, complexity, and impact of each requirement. An example template of such a prioritization matrix can be seen below.

Severity – Does the criteria affect sensitive features of the interoperability?

Scope – Does the criteria affect all users or a minor part of users?

Priority Assessment	
1-2 – High	• To be fixed in sprint 1 • Requires notification to project sponsor & project manager • Does not meet go live criteria if any high priority defect is open
3-4 – Moderate	• To be fixed in sprint 2 • Requires notification to project manager • Does not meet go live criteria if any high moderate defect is open
6-9 – Low	• To be scheduled when time is available or must be scheduled for further enhancements phase • Does meet go live criteria if any low priority defect is open

Table 29. Example of prioritization matrix

		Priority		
	Level	**High**	**Medium**	**Low**
Scope	**Large**	**1** User Story 3.1 & 3.2	**2**	**3** User Story 1.2
	Moderate	**2** User Story 2.2	**4** User Story 2.1	**6**
	Small	**3** User Story 1.1	**6**	**9**

B12. Implementation framework: Templates for Phase D – Deployment & monitoring

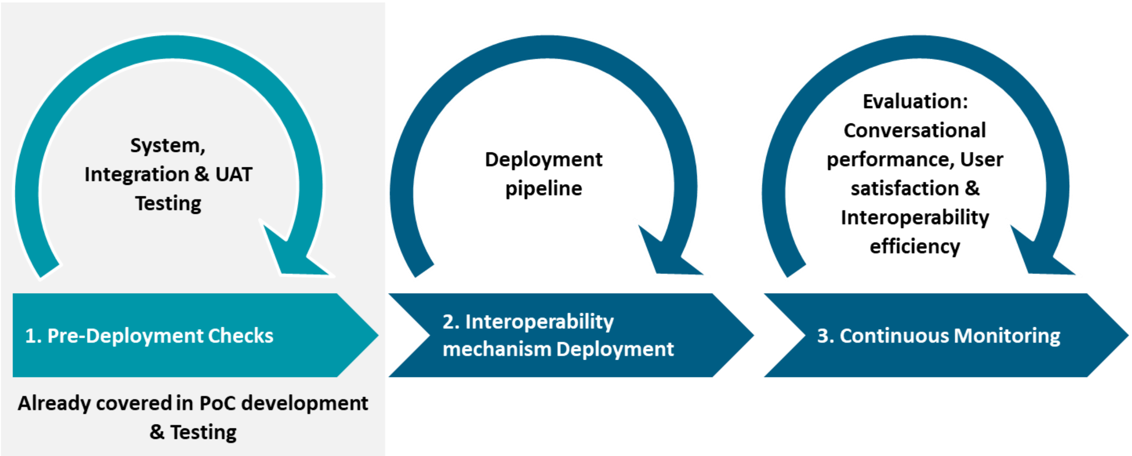


Figure 61. Deployment & Monitoring overview

Define KPIs

Monitoring revolves around observing the interoperability process for any issues that might arise.

Key Performance Indicators (KPIs) are a critical component in assessing the performance and success of the chatbot's interoperability in accordance with the set goals and objectives. These measurable values offer an insight into the effectiveness of the bot's functionality, user interface, interoperability driver, and user satisfaction, aiding in the optimization and improvement of future interactions. Based on the points mentioned above, the following KPI's were identified with examples of measures that should be defined before the development of the solution. They can be classified in three categories, the first one focuses on the conversational efficiency of the bot, the second one centres on measures around user satisfaction and retention and finally the last one emphasis on interoperability efficiency between the bots.

General requirements
Specific requirements
Interoperability requirements

Table 30. Monitoring KPIs

	KPI	Explanation	Example / Measure
1. Conversational efficiency provided by the bot	1. Incoming Connection Requests	A count of received incoming requests could demonstrate the activity or popularity level of the system.	This could be done by logging every data request per day, i.e. 200.
	2. Error Rate	The number of errors experienced per a certain number of interactions or a specific time period.	The error rate can be measured in percentage, i.e. less than 10%.
	3. Resolution Rate	The number of queries successfully answered by the chatbot with the help of other bots, compared to the total number of queries received.	The resolution rate can be measured in percentage, i.e. 85%.
	4. Overload Times	The number of times a chatbot has exceeded its response time limit due to high volume requests.	The overload times can be measured in percentage, i.e. less 10%.

	KPI	Explanation	Example / Measure
	5. Accuracy rate	The extent to which a chatbot correctly understands and appropriately responds to user queries.	On a sample of 100 queries, the chatbot provided correct and relevant responses to 81 queries. It has an accuracy rate of 81%.
	6. Re-query Rate	The frequency at which users have to repeat or rephrase their queries. A lower rate indicates that the bot understood the user's intent accurately the first time.	The re-query rate can be measured in percentage, i.e. less than 10%.
	7. User Interface Errors	Count of errors occurring regarding the interoperability switch. This could be issues related to displaying the returned answers to the user, showing error messages, etc.	The user interface errors can be measured in percentage, i.e. 5%.
2. User satisfaction and retention	8. User satisfaction Rate	This could be measured through a star rating at the end of interactions, asking users to rate their satisfaction on a scale and leave a comment if they want.	For example, on a scale from 1-5, an average rating of 4.5 would indicate 90% user satisfaction.
	9. User Retention Rate	The percentage of users who return for successive interactions after their first use. Higher retention rates generally indicate positive user experiences.	The user retention rate can be measured in percentage, i.e. 85%.
	10. Conversation duration	The time the conversation between the user and bot lasted based on the last answer provided by the bot.	Provided in seconds and minutes.
3. Interoperability efficiency	11. Connection Response Time	How long it takes from the moment a question leaves the chatbot to when an answer from another chatbot is received. The speed is impacted by the interoperability mechanism on which bot should answer the user depending on confidence score.	Ideally, bot connections should take under 10 seconds. Connections beyond 10-15 seconds may be deemed failures, prompting the driver to query another bot or inform the user about the inability to answer.
	12. Message Reformat Rate	The number of times the chatbot had to reformat a response from another bot to ensure compatibility with the original format.	The message reformat rate can be measured in percentage, i.e. 5%.
	13. API Trigger Time	The time taken by the AI driver to trigger an API. Shorter trigger times indicate better system performance. See section 1.4.3.3.	This should be a short as possible, for example a general good response time is 0.1-1 second.
	14. API Failure Rate	The percentage of attempted API triggers that fail. Lower rates are indicative of better system health.	This should be as low as possible, for example 5%.
	15. Metadata Recognition Accuracy	The AI driver's success rate at correctly recognizing and interpreting the metadata of other bots.	The metadata recognition accuracy can be measured in percentage, i.e. 95%.

	KPI	Explanation	Example / Measure
	16. Successful Connection Rate	The ratio of successful connections to another bot (requests answered within the expected time) to the total number of requests.	The successful connection rate can be measured in percentage, i.e. at least 80%.
	17. Fraction of Total Queries Served by External Bots	The proportion of total queries that the bot couldn't answer itself and had to rely on other bots to answer. A lower fraction suggests that the bot is more knowledgeable.	This can be measured by percentage, for example 40%.
	18. Fallback Frequency	How often the system has to default to a 'Fallback' or generic response because it could not retrieve a suitable response from another chatbot.	The fallback frequency can be measured in percentage, i.e. less than 15%.
	19. Outbound Connection Requests	A tally of the number of requests sent to other chatbots helps in measuring the workload on other bots (also gives implicit information on what users tend to ask more).	Chatbot A sent 21 requests to chatbot B (tax) and 3 request to chatbot C (immigration). Your users (chatbot A) tend to ask more tax related questions.

B13. PoC development

Table 31. PoC development: Acceptance criteria

ID	Name	Requirement Description	Measurement
AC1	*Response provision*	*The chatbot(s) should be able to answer user queries.*	*90% intent accuracy in responses*
AC2	*Data / knowledge exchange*	*The chatbot should exchange data with other systems, pulling in necessary information to answer user queries.*	*Programmatic accessible interface (API) provides expected information and the responses from BeaBot is seamlessly integrated in Publio's response*
AC3	*Error handling*	*Should a chatbot not respond, error messages should be issued to the user (e.g., Sorry, I don't understand your question. Could you try rephrasing?).*	*Always gives answer, if error then Publio communicates to the user that an error occurred, and user can get out of the loop*
AC4	*Response time*	*The chatbots should be able to respond to user queries within acceptable time limits (2 seconds on average).*	*2 seconds (95% cases average time below 2 sec)*
AC5	*Performance*	*The correct chatbot responds to the user query. Rerouting error should be below 20%.*	*Below 20% error rerouting (based on the 5 predefined intents [A.05] from BeaBot)*

ID	Name	Requirement Description	Measurement
AC6	*Usability*	*The chatbots connection should be user-friendly and intuitive for users, this includes see measurement*	- *Interoperability should be clear and understandable* - *Consent should be asked before switching chatbot* - *Users can switch between bots (when possible)* - *Interaction between the chatbots should be tracked (e.g., having the transcript of user conversation)*

B14. PoC Technical documentation

B.14.1 Introduction

This project focuses on developing a Proof of Concept (PoC) to demonstrate interoperability between chatbots of public institutions. The initial implementation integrates **Publio**, the chatbot of the Publications Office of the European Union, and **BeaBot**, the chatbot of the Belgian Digital Transformation Office (BOSA). Designed with scalability and practicality in mind, this PoC serves as a learning tool to understand the integration process, validate concepts of interoperability as described in parallel study report, and as a steppingstone for future enhancements in chatbot interoperability across public institutions.

B.14.2 Technical Specifications

Primary Knowledge Directory (PKD)

The Primary Knowledge Directory (PKD) includes the chatbot selection component that has a Bot Catalog and LLM sub-component which helps the host chatbot (in this PoC Publio) to select the most appropriate contributor chatbot that can answer a question.

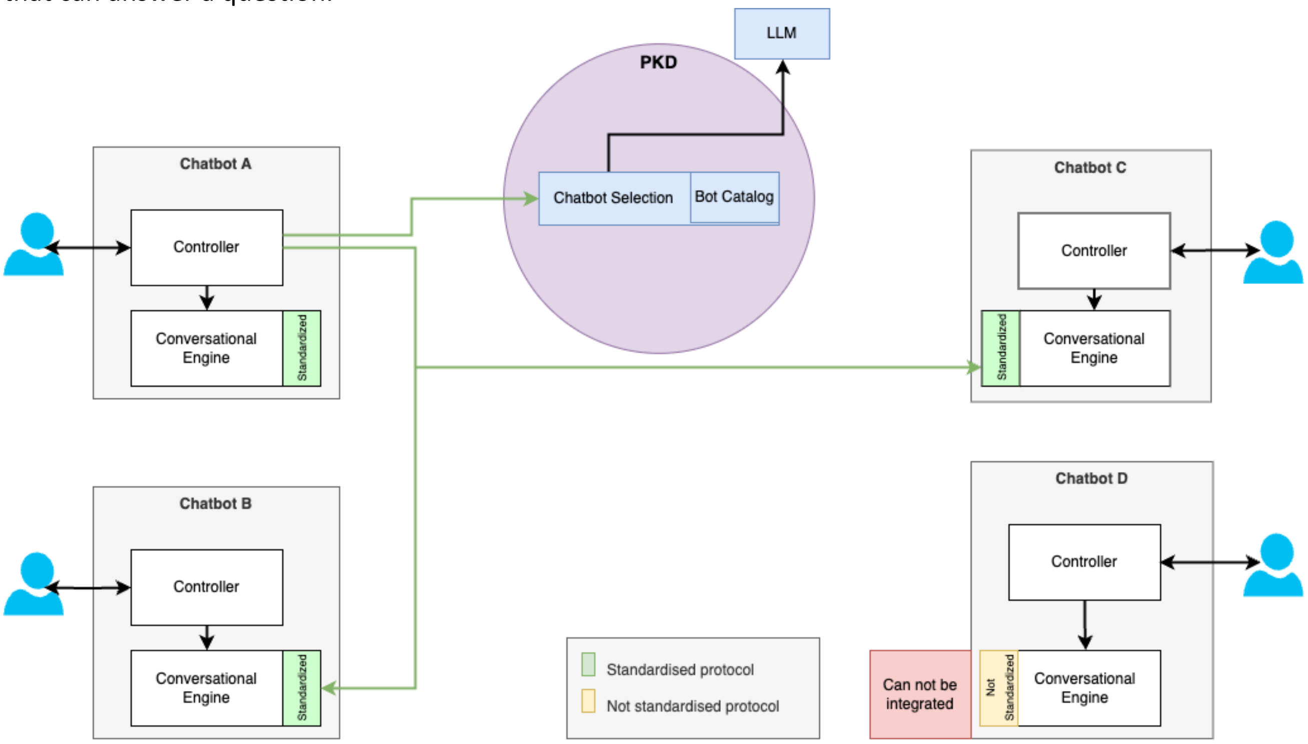


Figure 62. POC interoperability architecture

The PKD exposes a REST API with a “/pkd-service” endpoint for handling user queries and routing decisions.

- **Input Parameters:**
  - User query (Text, mandatory)
  - History (Text, optional)
- **Output Parameters:**
  - Selected chatbot identifier (ID, nullable)
    - In the industrialized version this would also include additional details about the bot (address, authentication credentials, name, links, visual identity). For the PoC the details have been hardcoded on the Publio BeaBot integration.
  - Detected intent (Text, nullable)
  - Needs Clarification (Boolean)
  - Clarification question (Text, nullable)

The PKD has been implemented as a JavaScript server hosted as an Azure function. For industrialization it should be moved to a node server hosted in the Kubernetes cluster.

Bot Catalog Overview

The Bot Catalog serves as a centralized repository that stores metadata for each participating chatbot, including:

- Chatbot name - The official name of the chatbot
- Description - A brief summary of the chatbot’s functionality; It can include general rules that apply to all intents, for example *“Should only reply to questions about Belgian public services*”
- Intent descriptions and sample utterances - An outline of the chatbot’s supported intents along with example user inputs
- Access details - Information required to interact with the chatbot (e.g., server name, authentication credentials).
- Visual identity - Icon and colour scheme associated with the chatbot.
- Supported languages - List of languages the chatbot can process
- Link towards “Learn more” page about the chatbot
- Link towards GDPR resources of the chatbot

1. **Notes for industrialization**

- For the PoC, the Bot Catalog is implemented as a JSON file, with a sample version provided in the Annex 6.1 Bot catalog.
- For the PoC some of the fields corresponding to BeaBot details are hardcoded as part of Publio and not part of the catalog data, as only a single bot was available for the PoC: API address, authentication credentials, chatbot name, Link towards “Learn more”, Visual identity. To support multiple chatbot integrations that implement a single standard API, the forementioned details should be moved to the catalog.
- For industrialization, the Bot Catalog should be stored in a database and made editable through a graphical user interface (GUI), accessible to users with administrative permissions.

2. **Notes on Chatbot Scope Definition**

To ensure that the **PKD** can route user queries to the most appropriate chatbot, each chatbot's scope must be clearly defined. During the **PoC**, we defined chatbot scope using a **short description, intent definitions, and sample utterances**. This approach is easy to explain, does not require complex decision-making and leads to precise results. The PKD stores a replica of each contributing chatbot intents.

Intent-Based Scope Definition (Traditional Approach)

In traditional, non-LLM chatbot systems, an **intent** is typically defined by a list of example utterances. These utterances can be **exported** from the contributing bot and **configured** within the PKD. This method was used for the PoC, where we mapped each intent in the contributing bot to a corresponding intent in the PKD.

Key considerations:

- **Selective Intent Configuration:** Not all intents from a contributing bot need to be configured in the PKD. For the PoC, only a subset of **BeaBot's intents** were included, and this selective approach remains a viable option for future implementations.
- **Manual Maintenance Challenge:** Keeping intents and utterances updated requires ongoing effort. To streamline this process, automation could be introduced by integrating with chatbot platforms to support **automated export and import** of intent definitions.

Description-Based Scope Definition (LLM Approach)

With modern **LLM-based chatbots**, alternative methods for defining scope become possible:

- If chatbots have **well-defined, non-overlapping areas of expertise**, a **natural language description** of their scope may be sufficient for the PKD to determine the correct routing—without manually defining intents and utterances.
- This approach is especially effective if the contributing bot is also **LLM-based**, as it would have a broader understanding of user queries within its domain, making explicit intent definition unnecessary.
    - Since BeaBot relies on traditional intent recognition, using LLM-based approach on the PKD would have resulted in a much broader recognition on the PKD, leading to the referral of questions that wouldn't have been supported by BeaBot.

Both methods offer advantages depending on the chatbot architecture. While **intent-based configuration** provides precise control, **LLM-based descriptions** could reduce maintenance effort and improve flexibility in chatbot integration.

Selection logic

The PKD utilizes **Azure OpenAI GPT-4o mini** for decision-making and clarification when necessary.

The prompt call to the LLM includes:

- Prompt instructions and examples
- Last user question
- Conversation history
- Bot catalog.

The diagram below illustrates the selection workflow.

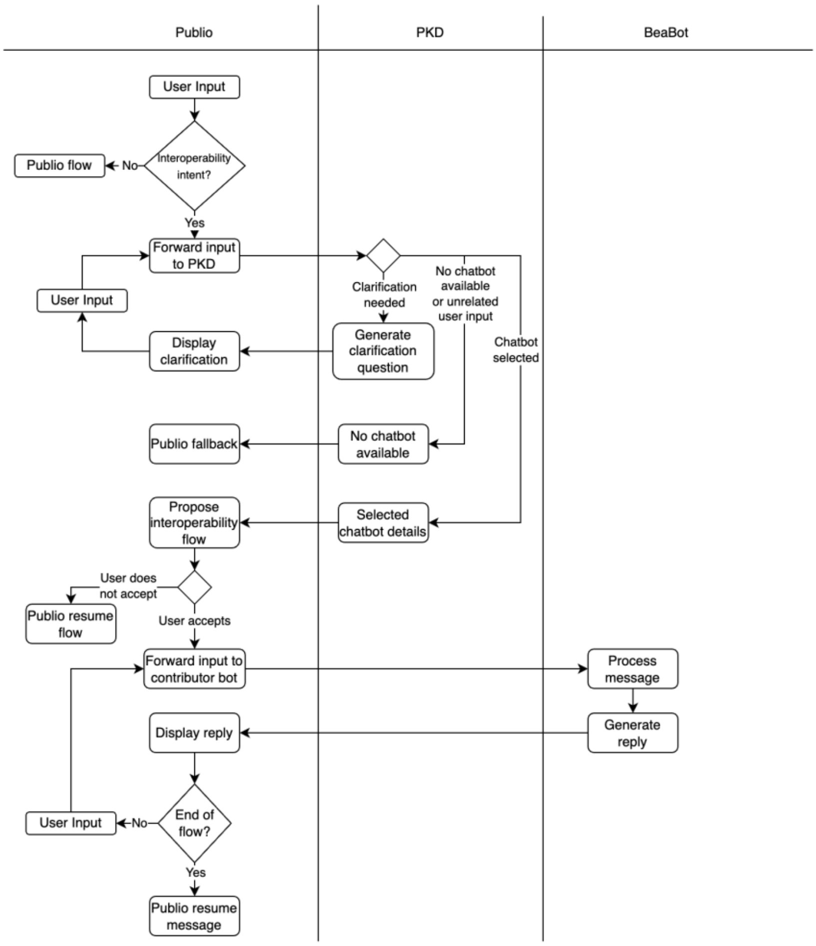


Figure 63. Selection workflow for the PKD

The LLM prompt structure follows these steps:

1. **Initial Chatbot Selection & Clarification Decision**
    - The LLM determines whether:
        - A chatbot can be selected immediately.
        - Further clarification is required.
        - No chatbot can respond to the query.
2. **Clarification Handling**
    - If clarification is needed, the user is prompted with a follow-up question.
    - The clarification exchange continues iteratively, with each user response being appended to the conversation history.
    - The same prompt is called repeatedly with the updated history until a resolution is reached.
    - Messages generated by the LLM are returned by the PKD to Publio and displayed
    - User input is sent by Publio to the PKD in a new call
3. **Handling Unrelated Responses**
    - If the user provides an **unrelated** response the system exits the clarification loop.
    - In this case, the original query is **sent to the host chatbot (e.g., Publio) for interpretation**.
    - The API responds with:

{

"chatbot": null,

"detected_intent": null,

"needs_clarification": false,

"clarifying_question": null

}

4. **Final Query Handling**
   - Once clarification is sufficient, the initial user query (not the entire clarification history) is forwarded to the selected chatbot for processing.
   - The PKD does not participate in the response exchange between the host chatbot (e.g., Publio) and the contributor chatbot (e.g., BeaBot).

Prompt has been added to Annex 6.2. PKD prompt.

The PKD also detects the appropriate intent that is only used for reporting purposes. Other reporting options should be analysed for industrialization.

Selection logic scale up

For the PoC the entire content of the catalog is sent to the LLM.
Also, the examples used in the prompt are anchored in the existing catalog.
This method can work well with 10-50 chatbots depending on the number of intents. (Under 20000 tokens total input).
It can be scaled to even more by adding a vector search as a prefiltering step and only including the most relevant results in the prompt. The vector search would consider all the 3 types of fields: chatbot description, intent description, sample utterances.
An alternative is to split the chatbots catalog entries into multiple batches and run them in parallel.
Since the catalog will no longer be constant, a sample small catalog should be included in the examples prompt section, so the examples will be anchored to a static catalog.

Chatbot Roles & Integration

Chatbots integrated into the system can act as:

- **Hosts** – Primary chatbots that interact directly with users (e.g., Publio).
- **Contributors** – Chatbots that provide responses but are not user-facing (e.g., BeaBot, MockBot).
- **Both** – Some chatbots may function as both a host and a contributor.

In the **PoC implementation**:

- **Publio** serves as the **host chatbot**.
- **BeaBot** and **MockBot** act as **contributor chatbots**.

Publio Integration

Publio acts as the host chatbot in the PoC. Key aspects of its integration include:

- **Query Routing:** Publio sends select queries to the PKD for chatbot selection. When BeaBot is chosen, Publio relays the query and presents the contributor chatbot's response to the user.

  For each transfer to BeaBot Publio is creating a new BeaBot conversation by calling the /chat/newConversation endpoint. No authentication needed. For industrialization authentication and networking-based security measures are recommended.

After getting the conversation id, Publio is using the /chat/message endpoint to relay the user messages. Every call includes the conversation id reference and the user message.

If a PKD clarification question was needed, the clarification message will not be included, just the original one.

At this point control of the conversation is passed to BeaBot until a termination condition is reached.

User messages are sent to BeaBot, and responses shown to the user.

The termination condition was missing from the BeaBot API, therefore we had to hardcode a predefined list of flow termination messages within Publio, including final messages and error messages. The standard API will need to include a termination marker to indicate when the control of the conversation needs to be passed back to the host bot.

BeaBot employs a synchronous REST API pattern. Meaning that for every message send by calling /chat/message, BeaBot returns a response message. This pattern is good enough for most chatbots but has some limitations. **A good practice is to also provide a WebSocket alternative** where chatbots can communicate asynchronously and exchange an asymmetric number of messages (the number of messages received doesn't need to equal the number of messages sent).

Advantages of async protocol vs REST API:

- It can handle long running tasks, for example where a slow legacy system needs to be interrogated. Networking configurations and some chatbot frameworks stop long running REST API requests after a timeout is reached.
- It allows for an asymmetric number of messages. This is critical if the system allows for human operator transfer.
- It allows the chatbot to proactively send messages.

Publio parses the response from BeaBot and is able to display messages, images and suggestion buttons.

BeaBot full API documentation will be submitted as an Annex.

- **Error Handling:** Publio implements fallback mechanisms for two scenarios:
  1. PKD downtime.
     - In case of receiving an error from the PKD, Publio will continue with existing Publio flows, skipping the interoperability mechanism.
  2. Errors from the contributor chatbot, such as unrecognized intents or delivery failures.
     - In case of receiving an error from the contributor bot, Publio will inform the user that there has been an error and resume control of the conversation to Publio.

BeaBot Integration

BeaBot functions as a contributor chatbot with the following key features: BeaBot provides an API accessible over the internet, supporting text, images, and chat button responses.

5 intents have been selected for the PoC out of BeaBot larger pool:
- Finding my national identification number (GENERAL_trouver_nrn)
- Possible internship (GENERAL_stage_chez_bosa)

- International applying to job (SELECTION_belge_fonctionnaire)
- Becoming a civil servant (SELECTION_devenir_fonctionnaire)
- Apply to internal market offers (SELECTION-postuler_marche_interne)

The intent detection on the PKD side is for statistical purposes only. After a chatbot has been selected by the PKD, it will be sent the original user question.

Mock Bot

A mock contributor bot has been included to ensure that the PKD solution is scalable for additional chatbots in an interoperability network comprised of multiple chatbots. For PoC purposes the tests were limited to seeing that intents can be facilitated correctly to such bot and no further conversation was built.

It can be triggered by asking for one of BeaBot's intents but specifying Germany instead of Belgium.

Decision Points

- **Host Decision:** Host chatbots, such as Publio, autonomously decide which intents to forward to the PKD for evaluation. For the POC, we defined a list of intents in LUIS that when detected, triggered the interoperability flow in Publio. The intents include the 5 selected BeaBot intents and a "score" intent. The "score" intent was configured as a non-relevant topic to showcase the PKD behaviour when no adequate chatbot is available to answer the question.
- **PKD Decision:** The PKD uses an LLM, in this case Azure Open AI GPT-4o mini, to assess the need for clarification, select the chatbot and intent or generate clarification questions if needed. The LLM prompt includes the instructions, the bot catalog and the user input. Since the decision is take by an LLM, no specific threshold is used. The instruction set also includes examples and can be finetuned to influence the LLM decision. Prompt has been added to Annex 6.2. PKD prompt.
- **Clarifying Questions:** When uncertain, the PKD can generate clarifying questions that the host chatbot presents to the user. For instance, a user asking about "internship opportunities" may be prompted to specify a country.

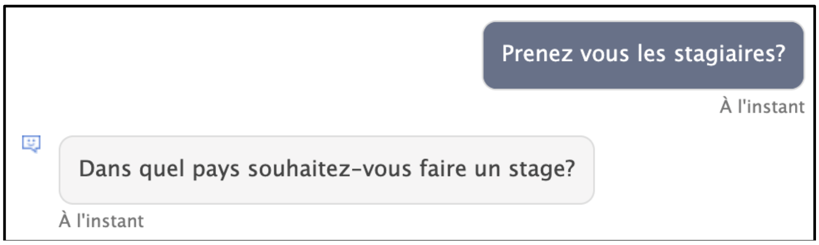


Figure 64. Example of clarifying questions

B.14.3 UX Specifications

- **User Story Table**

Table 32. Overview of the user stories

ID	User story	Intent name	Intent type	Potential trigger phrases		Similarity in intents (both Publio and BeaBot can answer)
				BeaBot	Publio	
US1	Belgian citizen looking for where to find the national identification number on the ID card	Finding my national identification number	Predefined flow *BeaBot*	• Ou se trouve mon numéro de registre national? • Comment trouver mon numéro de registre? • Ou est mon NISS	• N/A	No
US2	Student looking for a potential internship in the public sector	Possible internship	Direct answer *BeaBot*	• Puis-je faire mon stage en Belgique? • Peut on faire une demande pour un stage • Prenez vous les stagiaires?	• Un stage	Yes
US3	International citizen apply to job position	International applying to job	Direct answer *BeaBot*	• Je dois etre belge pour devenir fonctionnaire • Je peux postuler sans être belge? • Je peux m'inscrire au test même si je ne suis pas belge?	• Un emploi	Yes
US4	Candidate wanting to become a civil servant	Becoming a civil servant	Direct answer *BeaBot*	• Comment travailler pour l'Etat? • Comment devenir fonctionnaire • Comment faire pour devenir agent de la fonction publique?	• Un emploi	Yes
US5	Candidate looking for another job in the public sector in Belgium	Apply to internal market offers	Predefined flow *BeaBot*	• Postuler via le marché interne en tant que contractuel • C'est possible pour un contractuel de postuler via le marché interne? • Je suis contractuel je peux postuler pour le marché interne	• N/A	No
US6	Candidate looking for regulatory document	EU Law: Looking for the AI Act EU Publications: Financial report EU WhoisWho: Person	*Publio Search intent*	• N/A	• Réglementation de l'IA • Budget de l'UE 2009 • Jean-Marc Dalimier	No

- **Transparency and visual distinction between chatbots:**
  - BeaBot responses are clearly differentiated by colour and an icon.
  - Users are asked if they agree to be transferred to BeaBot
  - The transfer message includes a reference to learn more about BeaBot
  - To ensure transparency and that it is clear that interoperability is taking place the host bot always mentions where the information is coming from Bea Dit, Bea Demande...

Figure 65. Example of visual disctinction between Publio and BeaBot

- **Flow termination:** Publio detects flow termination or errors in BeaBot and regains control of the conversation.

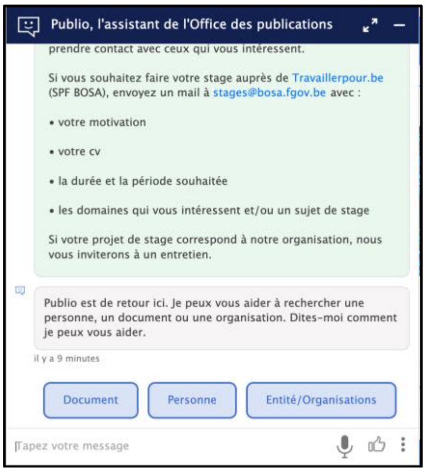


Figure 66. Example of flow termination

- **Restart Functionality:** Users can restart conversations using a button to return to Publio's interface.

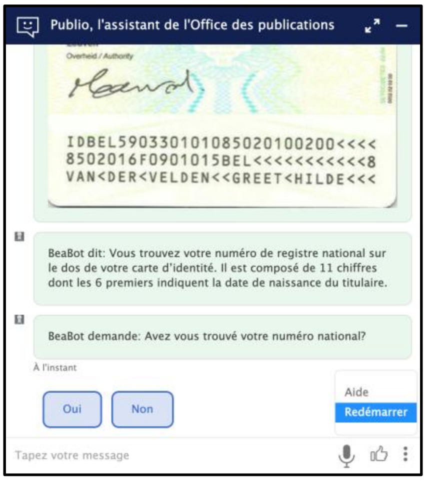


Figure 67. Example of restarting a conversation

B.14.4 Reporting and Feedback

- **Logging:** Logs capture all interactions, including interoperability-specific details such as query routing and chatbot responses.
- **Reports:** Generated reports include:
  - Message and conversation statistics by chatbot
  - Message and conversation statistics by flow for Publio and BeaBot
  - Feedback breakdown by chatbot
  - Unrecognized phrases by chatbot
- **Feedback Management:** Feedback is exclusively logged in Publio, with a breakdown available for queries involving BeaBot.

A sample report has been submitted as an Annex 1 document.

B.14.5 Assumptions and Limitations

- The PoC supports only the French language.
- Publio covers all existing intents, while BeaBot handles five selected intents.
- Feedback interoperability is out of scope. Meaning that feedback will not be shared between Publio and Beabot. The feedback will only be registered within Publio.
- Integration is unidirectional: Publio relays queries to BeaBot but not vice versa.
- Flow routing locks after the initial decision, with mid-flow messages remaining within the active chatbot.

B.14.6 Annex

Bot Catalog

Structure

[
{
"bot_name": <string>, # Name of the chatbot
"bot_description": <string>, # Description of the chatbot
"intents": [
{
"intent": <string>, # The specific intent this bot can handle
"sample_user_utterances": [
<string>, <string>, ... # Examples of user inputs that match this intent
]
},
...
]
},
...
]

Sample content

[
{
"bot_name": "bea_belgium",
"bot_description": "The chatbot of the Belgian Federal Public Service Policy and Support (BOSA)",
"intents": [
{
"intent": "Finding national identification number on the ID card - only for Belgium",
"sample_user_utterances": ["Ou se trouve mon numéro de registre national?","Comment trouver mon numéro de registre?","Ou est mon NISS"]
},
{
"intent": "Possible internship - only for Belgium",
"sample_user_utterances": ["Puis-je faire mon stage en Belgique?","Peut on faire une demande pour un stage","Prenez vous les stagiaires?"]
},
{
"intent": "International applying to job - only in Belgium",

"sample_user_utterances": ["Je dois etre belge pour devenir fonctionnaire","Je peux postuler sans être belge?","Je peux m'inscrire au test même si je ne suis pas belge?"]
},
{
"intent": "Becoming a civil servant - only in Belgium",
"sample_user_utterances": ["Comment travailler pour l'Etat?","Comment devenir fonctionnaire","Comment faire pour devenir agent de la fonction publique?"]
},
{
"intent": "Apply to internal market offers - only in Belgium",
"sample_user_utterances": ["Postuler via le marché interne en tant que contractuel","C'est possible pour un contractuel de postuler via le marché interne?","Je suis contractuel je peux postuler pour le marché interne"]
}
]
},
{
"bot_name":"bea_germany",
"bot_description": "The chatbot of the Germany Federal Public Service Policy and Support (BOSA)",
"intents": [
{
"intent": "Finding national identification number on the ID card - only for Germany",
"sample_user_utterances": ["Ou se trouve mon numéro de registre national?","Comment trouver mon numéro de registre?","Ou est mon NISS"]
},
{
"intent": "Possible internship - only for Germany",
"sample_user_utterances": ["Puis-je faire mon stage en Allemagne?","Peut on faire une demande pour un stage","Prenez vous les stagiaires?"]
},
{
"intent": "International applying to job - only in Germany",
"sample_user_utterances": ["Je dois etre allemand pour devenir fonctionnaire","Je peux postuler sans être allemand?","Je peux m'inscrire au test même si je ne suis pas allemand?"]
},
{
"intent": "Becoming a civil servant - only in Germany",
"sample_user_utterances": ["Comment travailler pour l'Etat?","Comment devenir fonctionnaire","Comment faire pour devenir agent de la fonction publique?"]
},
{
"intent": "Apply to internal market offers - only in Germany",
"sample_user_utterances": ["Postuler via le marché interne en tant que contractuel","C'est possible pour un contractuel de postuler via le marché interne?","Je suis contractuel je peux postuler pour le marché interne"]
}
]
}
]

PKD prompt

`

You are part of a user request redirection service for an interoperability mechanism of a public institution within the European Union.
Your task is to determine if an available chatbot can handle a user request.
The available chatbots and their supported intents are provided in the following catalog:
${bot_catalog}

Rules:
-If no chatbot and intent matches the user question, the resolution should be "unknown".
-If clarification is needed, ask an open-ended question in French.
-Do not attempt to answer any user questions yourself.
-Pay attention to the countries to which the request applies

Response Format
{
"resolution":<"bea_belgium" | "bea_germany" |"unknown">,
"detected_intent": <detected intent,else default to empty string>,
"needs_clarification": <true |false>,
"clarifying_question": <optional question for the user if clarify is true to clarify if any of the bots could handle the intent, else default to empty string>
}

Examples

Example 1:
Conversation history: "User: Puis-je obtenirun stage"
Answer: {
"resolution": "unknown",
"detected_intent": "",
"needs_clarification": true,
"clarifying_question": "Dans quelpays souhaitez-vous postuler pour le stage?"
}

Example 2:
Conversation history: "User: Puis-je obtenirun stage \n Assistant: Dans quel pays souhaitez-vous postuler pour le stage? \nUser: Je souhaite postuler en Belgique"
Answer: {
"resolution": "bea_belgium",
"detected_intent": "Possibleinternship - only for Belgium",
"needs_clarification": false,
"clarifying_question": ""
}

Example 3:
Conversation history: "je cherche le AI Actdans l'UE"
Answer: {
"resolution": "unknown",
"detected_intent": "",

"needs_clarification": false,
"clarifying_question": ""
}

Example 4:
Conversation history: "User: Puis-je obtenirun stage \n Assistant: Dans quel pays souhaitez-vous postuler pour le stage? \nUser: Je souhaite postuler en Roumanie"
Answer: {
"resolution": "unknown",
"detected_intent": "",
"needs_clarification": false,
"clarifying_question": ""
}
Example 5:
Conversation history: "User: stage enAllemagne"
Answer: {
"resolution": "bea_germany",
"detected_intent": "Possibleinternship - only for Germany",
"needs_clarification": false,
"clarifying_question": ""
}

Example 6:
Conversation history: "User: Comment devenirfonctionnaire en France?"
Answer: {
"resolution": "unknown",
"detected_intent": "",
"needs_clarification": false,
"clarifying_question": ""
}
Use case
Conversation history: ${user_message}`

Source code packaging

The PoC consists of the following resources that will be shared as requested:

- PKD – JavaScript project, hosted as Azure function. Includes API functionality, Bot Catalog, PKD prompt.
- Publio interoperability version, including all Publio components (Webchat, Token Server, Bot Framework
- Composer Project – includes LUIS configurations and other Azure services configurations)

  Technical specifications