



Specifications for phase 1 (solution specification and phase 2 set-up) for a visa chatbot

VisaChat: Designing an EU chatbot to improve the efficiency of the visa procedure

HOME/2020/ISFB/FW/VISA/0021

Deliverable D2.01: Target architecture and design



Table of Contents

1	Introduction.....	3
1.1	Context.....	3
1.2	Objectives	4
2	Future state definition recap	5
2.1	Business and functional requirements.....	5
2.2	Other key stakeholder considerations	6
3	Technical requirements.....	8
3.1	Hosting and technology.....	8
3.2	Maintenance	8
3.3	Interoperability.....	9
3.4	Security.....	10
3.5	Standards and principles	10
3.6	Summary table	11
4	Vendor comparison	12
4.1	Methodology	12
4.2	Chatbot vendor analysis	13
4.3	Recommendations.....	16
5	Target architecture	17
5.1	Layered architecture representation	17
5.2	Estimation of hardware requirements.....	23
5.3	Summary of target architecture	26
6	Conclusion and next steps	28
Annex A -	Chatbot vendor interviews.....	29

Table of Tables

Table 1: Summary of deliverables	4
Table 2: Overview of technical requirements	11
Table 3: Minimum/recommended requirements for on-premise hosting.....	25
Table 4: Requirements for Rasa Helm Chart (production environment) installation	25

Table of Figures

Figure 1: Vendor comparison - key criteria	12
Figure 2: Vendor comparison - preliminary results	13
Figure 3: Spectrum of chatbot vendors	16
Figure 4: Target architecture - presentation layer.....	18
Figure 5: Target architecture - business layer	19
Figure 6: Target architecture - data layer	20
Figure 7: Target architecture - other components.....	21
Figure 8: Conceptual data model - Answering questions	22
Figure 9: Conceptual data model - Monitoring performance	23
Figure 10: Estimated usage of the chatbot	24
Figure 11: Target architecture - summary.....	27

1 Introduction

The *Target architecture and design* report is the main output of the second task of the first phase of the *VisaChat* project. The overarching objective of this project is to develop an EU cross-border chatbot to improve the efficiency of the visa procedure. Phase 1 of this project lays the foundation for Phases 2 and 3, the project execution and service continuity respectively. Phase 1 defines the exact scope and purpose of the chatbot, what should be developed, who is involved and how the solution can be maintained. In addition, a proof of concept is developed to demonstrate the way of working and the added value of a chatbot. Next, in Phase 2, the chatbot will be developed by a consortium of stakeholders, such as the Member States, eu-LISA and external partners. Finally, Phase 3 deals with hosting and maintaining the developed solution.

The second task of the *VisaChat* Phase 1 concerns the technical requirements and target architecture. Starting from the future state definition (Task 1), this report aims to list the **technical requirements** and **target architecture**. The chatbot technology that can be leveraged for the future solution is also extensively discussed, by means of a **vendor comparison**. While these three parts are presented in separate chapters, there are strong interplays between them.

1.1 Context

The Schengen area is an area consisting of 26 European countries that agreed to withdraw border controls at their mutual borders¹. Nationals from a set of third countries² must be in possession of a visa when crossing the external borders of the Schengen area. These short-stay visas are required for stays not exceeding 90 days in any 180-day period which is typically the case for travels for tourism or business purposes. The issuance of these Schengen visas is organised by means of the Visa Code that applies to all Schengen Member States. However, each Member State can organise the visa issuance process according to its own rules and preferences, as long as these are in compliance with the common practices included in the Visa Code.

Evaluations show that the visa process is time and cost intensive for the Member State's consular posts and the corresponding central administration. One of the reasons is that visa authorities have to answer many requests from visa applicants. Therefore, DG HOME is looking into smart applications to relieve the administration of the Member States.

In this context, the overall purpose of the *VisaChat* project is to deliver to Member States a cognitive application that answers questions from visa applicants in compliance with the Visa Code. The expected benefits of the envisaged visa chatbot are to reduce the number of information requests by phone or emails from applicants handled by consular staff. This will be the case if the applicant receives accurate and interactive support via the chatbot.

The project's phase 1 should define how the future solution should look like and how it should be run. In addition, the value of a chatbot solution will be demonstrated through a proof of concept. Moreover, the set-up of an Artificial Intelligence (AI) Centre of Excellence (CoE) will be outlined. The outcome of this first phase will provide the Commission with a view of developing and deploying a live chatbot (Phase 2) in 2022. Phases 2 and 3 will focus on the actual deployment and maintenance of the developed solution.

¹ For a more complete description of the Schengen area and its policies, please consult https://ec.europa.eu/home-affairs/policies/schengen-borders-and-visa_en :

² This list of third countries can be found in Regulation (EU) 2018/1806 of the European Parliament and of the Council of 14 November 2018 listing the third countries whose nationals must be in possession of visas when crossing the external borders and those whose nationals are exempt from that requirement

1.2 Objectives

The end objective of this report is to design the target architecture. This requires a good view of the components that will constitute the final solution, and the linkages between them. The components depend on the business and functional requirements that have been identified for the chatbot. Additionally, there is a set of technical requirements that should be considered before defining the architecture. While it is possible to develop the chatbot solution from scratch, a more standard approach consists of leveraging technologies from various vendors.

Similar to the other requirements, the technical requirements are defined in co-creation with the European Union Agency for the Operational Management of Large-Scale IT Systems in the Area of Freedom, Security and Justice (eu-LISA)³. An interview with eu-LISA was organised to shape the technical requirements and validate the approach for the vendor comparison. For this vendor comparison, a two-step approach was followed: first, a high-level assessment was performed based on information found online. Then, vendors fulfilling the key requirements were shortlisted for a follow-up interview, during which their solution was discussed in-depth.

Finally, the target architecture is discussed. The typical components of a chatbot solution are discussed first. For some of the components, vendor technology can be leveraged. Then, an estimation is made of the hardware requirements (compute resources, data storage), based on a prediction of the usage. This estimation is compared to the requirements shared by a vendor. All components and requirements are then summarised into one architecture. To further illustrate the architecture, the conceptual data model is presented as well.

The report is structured in six chapters. After this introductory chapter, Chapter 2 recapitulates the main outcomes of the business and functional requirements discussions with the Member States. This serves as input for the technical requirements, discussed in Chapter 3. With a good view on the requirements in mind, the products of different EU-based chatbot vendors can be assessed, in Chapter 4. Chapter 5 then presents the main output of this task, the target architecture. Chapter 6 closes this deliverable by presenting the main conclusions and next steps.

For reference, Table 1 provides an overview of all deliverables in scope of the current project.

Table 1: Summary of deliverables

ID	Deliverable name
D1.01	Future State requirements report
D2.01	Target Architecture and Design report
D2.02	Proof of Concept report
D3.01	Artificial Intelligence Centre of Excellence report
D4.01	Solution Delivery report

³ <https://www.eulisa.europa.eu/>

2 Future state definition recap

Deliverable D1.01 of this project defines the future state for the chatbot from a business and functional perspective. It encompasses what the chatbot should be able to do, and which features can be included to achieve this. The objective of this chapter is to provide a recap of the requirements that have a critical impact on the technical side of the chatbot, since this serves as a starting point for the remainder of this report.

2.1 Business and functional requirements

The business requirements were categorised into four clusters: positioning of the bot, capabilities, support types and interoperability. The functional requirements were presented by explaining the possible conversational flows that users can have with the bot. In this section, the functional requirements are provided next to the business requirements to which they relate.

Positioning of the chatbot

Business requirements: The key requirement for the positioning of the chatbot, is that it should be available both on the Member State websites, DG HOME website and a possible future common EU visa application⁴ platform when it is live. Integration with the External Service Provider (ESP) website and other channels such as WhatsApp were deemed optional. During the interviews with stakeholders, it became clear that some responses to applicant's questions depend on the destination country of the applicant (links to Member State webpages, required documents ...).

Functional requirements: The chatbot should be available both centrally and decentrally (per Member State). The data set should consist as much as possible of questions that are common to all Member States. Only when needed, the answers should depend on the Member State. More generally, there are many other parameters that determine the reply to the user: country of residence, nationality, age, type of passport etc. This will have to be recorded in the chatbot conversation. If the user did not provide this information, the chatbot should first ask this, before answering.

Capabilities

Business requirements: As it is intended for third-country nationals, the chatbot should provide multilingual capabilities. This should at least cover the EU languages, and preferably also the most important Third Country National (TCN) languages, such as Russian and Mandarin. Given that many interactions are currently happening using speech, the stakeholder group saw value in speech-to-text (STT) and text-to-speech (TTS) technologies. Finally, the chatbot should have the capability to automatically learn.

Functional requirements: The languages can be included using a tiered approach. Tier-1 languages are the most important languages, in which the whole data set is translated (supervised approach). For tier-2 languages, the bot would apply machine translation services to the incoming request, search for an answer in the English data set, and then apply machine translation again to provide an answer in the user language (unsupervised approach). The speech technology can be integrated by leveraging dedicated STT and TTS services, and by including a microphone button on the bot's interface. To enable (semi-)automated improvement, it is essential to capture feedback from the users. Analysis of negative feedback should lead to an improved data set, and in combination with positive feedback, could be leveraged to improve the Natural Language Understanding/Processing (NLU/NLP) engine.

Support types

Business requirements: Potential applicants typically face challenges in the steps before the submission of a visa: whether they need a visa, which documents are required, how they can apply etc. These questions are

⁴ The proposal for a future EU application platform is currently under preparation. Co-legislators will have to adopt this proposal for the concept to become a reality.

considered as Frequently Asked Questions (FAQs), which depend on the situation of the applicant, but are not case-specific. These types of questions should definitely be included in the chatbot. On the other hand, questions related to specific applications (e.g. status checks) were deemed less important, since they require the bot to deal with personal data and communicate with back-end systems. The chatbot could also be designed to support the visa application process itself, by allowing users to submit documents or pay the visa fee. For the same reasons, this is preferably not included in the chatbot. The stakeholders indicated that there should be a clear division of tasks between the chatbot (support for questions) and the administration (the actual handling of the application).

Functional requirements: The chatbot should provide two types of support: guided (using buttons) and free-text. The most important topics of the visa application process are offered in the initial menu as a separate support type: visa wizard, support for application submission, status checks and post-issuance support. Note that for the status checks, the default support is a generic response on the duration of the application processing. In the future, or for some Member States, case-specific information can be shared if feasible and desired. The four topics and their follow-up questions can be selected by users by means of buttons. Next to the four support streams, users should also be offered the possibility to ask any question they have in a free-text approach. Here, the bot will have to apply NLU/NLP techniques to find the most appropriate answer.

Interoperability

Business requirements: Assuming corresponding legislation would be adopted, applicants will have the possibility in the future to submit and view their visa application through an EU Online Visa Application Platform. This platform will be a centralised tool for applicants to submit and view their visa application. The chatbot should therefore be closely connected to this system. Both can be connected to the same data sources.

As explained in the previous section, status checks were not deemed very important for the near future. Therefore, the chatbot should only have limited integration with other systems. The Member States were not in favour of interaction with their local systems. The Central Visa Information System (VIS) could be connected, but this only yields limited value, at the cost of increased security risks since VIS contains personal data. Therefore, a specific cost-benefit analysis is required. Systems of the ESPs might contain additional status information (delivery, pick-up ...) for the applications they support, but are less feasible to integrate from a security perspective, since this requires a connection between the visa chatbot and a commercial party database.

Functional requirements: The chatbot should contain the same information as the EU Online Visa Application Platform. For example, the status information for specific applications can follow the same authentication method as is applied for the Digital Visa platform. The other systems do not yield functional requirements.

In addition to the external systems, two tools should be developed and connected to the chatbot: a monitoring dashboard and a content management system. This will allow for streamlined follow-up and improvement of the bot's performance.

2.2 Other key stakeholder considerations

Aside from the business and functional requirements, there were also some other key considerations shared by the Member States that should be taken into account for the design of the solution.

- **EU-based vendors:** If external technology or services are used, these should be EU-based.
- **Data reside within EU:** Even though the chatbot at its initial stage will not process sensitive personal data, the data should be stored in data centres located in the EU, as an additional line of defence to guarantee European Union Data Protection Regulation (EUDPR) and General Data Protection Regulation (GDPR) compliance (see also D1.01, Section 3.6).
- **Security:** Even though security is a technical aspect of the chatbot, the Member States expressed a secure solution as one of the main requirements during the interviews. For this reason, the chatbot

will not be connected to the Member State VIS systems. The Member States also raised concerns on the maintenance and management of the bot, which cannot be outsourced to external parties. In this context, it is advantageous if business owners (Member States, DG HOME) can adapt the chatbot's responses.

- **Open-source:** Open-source code allows for more transparency and reduces the dependency on external services, and should therefore be prioritised.

Interestingly, all these considerations were also raised by eu-LISA during the technical requirements interview (see next section).

3 Technical requirements

This chapter summarises the technical requirements for the visa chatbot. Starting from five key topics (hosting and technology; maintenance; interoperability; security; standards and principles), a questionnaire was created. Then, an interview with eu-LISA was organised. During this interview, the initial understanding of the requirement topic was explained to and validated with eu-LISA. Then, the topics were further discussed with eu-LISA's architects and experts.

The first five subsections below elaborate on each topic. Section 3.6 consolidates this information in a summary table.

3.1 Hosting and technology

During the business requirements interviews, the Member States suggested to opt for a **central hosting**, instead of hosting per Member State. This will reduce the overall workload to develop and maintain the solution, and also ensures consistency across all Schengen-countries. eu-LISA agrees with this approach, and confirms they can be responsible for the chatbot hosting.

The solution can either be hosted on a **public cloud, private cloud or on-premise infrastructure**. The hosting consists of two elements: the server on which the solution runs, and the database to which the solution is connected. This database can contain the question base where the chatbot can find responses, and the logged conversations. In the future, when the chatbot will provide information on individual applications, external systems can be added as additional data source. eu-LISA expressed clear preference for an on-premise hosting, but does not eliminate a hybrid solution, with the chatbot hosted in the cloud, but the data stored on-premise. However if external systems have to be connected, this renders hosting (partly) on cloud difficult. Therefore, the remainder of this document will assume a **fully on-premise solution**, unless the vendor comparison shows this is infeasible. It is assumed that typical infrastructure requirements such as elasticity, scalability and fault tolerance are by default guaranteed by eu-LISA's infrastructure.

Next to the hosting infrastructure, the **technology** used for the chatbot is another crucial choice. In theory, eu-LISA could hire data scientists to develop a solution from scratch. Data engineers and architects subsequently deploy the solution on the hosting infrastructure. To ease and speed up this process, existing technologies can be used. Specialised vendors offer frameworks or even full platforms for clients to develop and design their custom chatbot. This is the subject of Chapter 4. Concerning the choice of technology, eu-LISA is open to all EU-based technologies, since this is a new domain. However, **open-source technologies** should be prioritised, since the techniques should be transparent (the AI models should be explainable) and owned by eu-LISA (no dependence on a vendor).

3.2 Maintenance

After development and deployment of the solution, it should be maintained. The maintenance covers the infrastructure, software and operations.

The **infrastructure maintenance** deals with the compute resources and databases connected to the solution. Since it is assumed that eu-LISA will host the chatbot from an on-premise infrastructure, eu-LISA will be responsible for this technical type of maintenance. The agency currently manages three operational large-scale IT systems⁵, and is developing three other ones. They are hence experienced in maintaining infrastructure.

The **software maintenance** covers the versioning of the algorithms and services integrated in the chatbot solution. The state-of-the-art NLP technologies are rapidly evolving. The first NLP algorithms were rule-based, overtaken during the last decade by machine learning and statistical models. Recently, deep learning-based

⁵ <https://www.eulisa.europa.eu/Activities/Large-Scale-It-Systems>

transformer models emerged as the most promising technique for a variety of NLP tasks.⁶ This implies that over time, the underlying algorithms of the chatbot might evolve. Moreover, patches and bug-fixes will be required to guarantee a performant solution. This will be a shared responsibility between eu-LISA and the selected chatbot vendor, if any. The responsibility of each organisation depends on the type of solution that is implemented. If the chatbot will be fully managed by the vendor, they are in charge of maintaining the software. On the other hand, when a more custom solution is built, the code might be fully owned and maintained by eu-LISA. In general, more ownership by eu-LISA results in more maintenance responsibilities.

For the **operational maintenance**, i.e. following up on the user interactions and the bots performance, two tools are proposed: a monitoring tool and a content management system. A mock-up of both tools is presented in Deliverable D1.01, and this will also be explained in greater detail in D4.01. eu-LISA agrees that there is a need for these tools, although such tools are not yet developed. The monitoring tool should cover technical metrics, to diagnose problems and assess security breaches, as well as business insights such as conversations.

3.3 Interoperability

Interoperability implies that the developed chatbot solution can interact with other IT applications. Of all the large-scale IT systems managed by eu-LISA, there are only two that could potentially enhance the chatbot: the central VIS and the yet to be developed EU online visa application platform. The other systems are not directly related to the processing of short-stay visas.

Central VIS

Integration with the **central VIS** (referred to as ‘central’ to distinguish from the national Visa Information Systems) can be used to provide limited status information: whether an application is submitted and whether the decision was positive (visa issued) or negative (visa denied). When a visa is issued, the status extended, revoked or annulled is also recorded centrally. Member States can perform alphanumeric searches on the central VIS to find persons, their application(s) and any stored biometric data.

When integrating this information in the chatbot, it is crucial to apply the right security mechanisms. Personal data (such as the biometric data) should never leak from the internal systems. A careful assessment should be made of which information should be retrieved from the central VIS (including status information, excluding biometric data), and how external users can access their application only. For example, a view can be created that contains one row per application, with a **unique identifier** (e.g. a numerical code, not referencing the natural person) and the related status information. After submission of the visa, applicants are sent this code, and they can provide this to the chatbot to get the information required. The search activities of the bot in the VIS should also be monitored. The bot should also have restricted access to the VIS, only for the specific case it’s trying to resolve. A feasibility study should determine if this is feasible in a secure way, and if the benefits outweigh the risks.

EU online visa application platform

The main other system with which the chatbot should interact, is the **EU online visa application platform**. To digitalise the current paper-based Schengen visa process and better secure and reinforce trust in the Schengen Area, the European Commission’s proposed new Pact on Migration and Asylum aims to digitalise Schengen visa procedures, including a digital visa and the ability to submit visa applications online. Recently, a prototype was launched to demonstrate the digital platform’s advantages. It is clear that the chatbot should be integrated in the portal, next to the integration on webpages.

Additionally, the chatbot should also benefit from the **online visa application data sources**. The applications should be accessible from the central platform. Applicants are urged to log in or sign up in case they wish to

⁶ Worth mentioning in this context are Bidirectional Encoder Representations from Transformers ([BERT](#)), a versatile model developed by Google and explained in D1.01, and Generative Pre-trained Transformer 3 ([GPT-3](#)), a language model recently acquired by Microsoft that can produce human-like text.

submit their visa application, potentially through **Multi-Factor Authentication** (MFA). After submission, applicants can consult the status of their application or appointment and after issuance, applicants can request a visa extension or revoke their visa.

The chatbot should leverage the same authentication measures as the platform itself. If users then wish to access the status of their application, they can either navigate to the respective webpage, or interact with the chatbot. The data source for both tools is common. Since the architecture for the EU online visa application platform is yet to be developed, **integration with the chatbot can be foreseen from the start**.

3.4 Security

Security deals with the protection of the solution and data. The solution should be safeguarded to attacks and sensitive data should never leak from the application. During interviews with the Member States, security was always indicated as a crucial aspect of the chatbot. Access to the components of the architecture should be restricted as much as possible.

eu-LISA established a '**Build securely by design**' principle, which states that every system managed by the Agency requires the creation and maintenance of a Security Risk Assessment, Security Plan and Business Continuity Plan. The chatbot at first instance only deals with generic information (no personal information), which reduces the security risk. The chatbot is a public endpoint, but there is no connection to the other systems. After integration of other systems, the security mechanisms should be revised with great care to make sure that the internal data cannot leak from the chatbot.

In addition, the '**Privacy by design**' principle states that personal data should only be processed when necessary, in a secure way. This can be achieved through encryption, anonymisation or obfuscation (masking) techniques to ensure that the data, if leaked, does not allow for identification of natural persons. Within the scope of this chatbot, this is only applicable when integrating with the EU Online Visa Application Portal (or the VIS, if deemed valuable). The considerations of that initiative will also apply to the chatbot. In terms of logged data, retention policies should be set up anyway, from an architectural standpoint (see Section 5.2.2).

3.5 Standards and principles

Architectural standards can be specific to an organisation to ensure harmonised solutions and enable interoperability. This also impacts the documentation format of the target architecture. eu-LISA has shared their catalogue of architectural principles, which should be considered for every new solution architecture. This section discusses the principles that are relevant for the chatbot exercise, but not yet discussed in one of the previous sections.

Some important principles are dealing with **data standards**. Each data element should have a single authoritative source, to avoid any unintended changes made to the data. The ownership of the data is closely related to the operational maintenance of the chatbot (see Section 3.2) and will be further established in the report D4.01 on the operating model. A definition of all data elements attached to the chatbot should be provided in a vocabulary.

The **applications** managed by eu-LISA must be composed of different services. For example, the future chatbot can provide support for questions and can answer status questions. Both services can be tested independently, enhancing the flexibility and easing the release management approach. This will be achieved via epics in the backlog of features, also included in report D4.01. In addition, there should be a fully automated process for continuous quality control. This point should be discussed with potential chatbot vendors. It should also be assessed whether their solutions can be created in multiple versions or on multiple environments (development, acceptance, pre-production, production).

In addition to this catalogue, eu-LISA also commented that the architecture should follow a layered representation. This is documented in Section 5.1.

3.6 Summary table

Table 2 summarises the technical requirements and preferred options. This table is crucial for the remainder of the document, since it is used as guide for the vendor discussions (Chapter 4) and serves as input for the target architecture (Chapter 5).

Table 2: Overview of technical requirements

Topic	Technical requirement	Preferred option
<u>Hosting and technology</u>	Hosting central vs. decentral	<ul style="list-style-type: none"> Central hosting (one solution for all MS)
	Hosting on-premise, hybrid or in the cloud	<ul style="list-style-type: none"> <i>Preference</i>: Fully on-premise solution <i>Only if needed</i>: Hybrid solution, with data centre on-premise
	Technology selection	<ul style="list-style-type: none"> <i>Preference</i>: Open-source technology <i>Preference specific technology</i>: --
<u>Maintenance</u>	Infrastructure management	<ul style="list-style-type: none"> <i>If on-premise</i>: Responsibility of eu-LISA
	Software management	<ul style="list-style-type: none"> Shared responsibility between eu-LISA and chatbot vendors
	Operational management	<ul style="list-style-type: none"> Creation of monitoring tool (logging + insights) Creation of content management system Shared responsibility between eu-LISA and Member States
<u>Interoperability</u>	Integration with central VIS	<ul style="list-style-type: none"> Feasibility study required to determine if feasible and desired
	Integration with EU Online Visa Application Portal	<ul style="list-style-type: none"> Chatbot embedded in portal Common authentication practices and status information in portal and chatbot
	Integration with other IT systems	<i>Not foreseen</i>
<u>Security</u>	<i>Build securely by design</i>	<ul style="list-style-type: none"> <i>Initial phase</i>: Limited security risk (no integration with other systems) <i>For integration</i>: Feasibility study required (with focus on security)
	<i>Privacy by design</i>	<ul style="list-style-type: none"> <i>Initial phase</i>: Limited security risk <i>For integration</i>: Re-use considerations from EU Online Visa Application Portal Configure data retention policies (see 5.2.2)
<u>Standards & principles</u>	Data standards	<ul style="list-style-type: none"> Document data ownership processes
	Application standards	<ul style="list-style-type: none"> Service-oriented developments Automated, continuous quality control Availability of multiple environments and/or versions
	Architecture representation	<ul style="list-style-type: none"> Layered approach (see 5.1)

4 Vendor comparison

Chatbots and more advanced digital assistants are an emerging technology that can support many organisations, due to their versatility. While every chatbot use case is unique, the underlying technology and operations can be common for many different use cases. To tap into this potential, a landscape of specialised chatbot-as-a-service vendors has emerged over the recent years. Additionally, major technology vendors such as Amazon, Microsoft, Google and IBM have developed a chatbot service, to complement their cloud solutions and related products.

Despite the common type of technology that all these players offer, there are significant differences between their products. The purpose of this section is to define some minimum requirements to which the solutions should comply, select some promising vendors and finally perform a detailed comparison after interview with the vendors. This methodology is outlined in Section 4.1. The output of the interviews is summarised in Section 4.2. The recommendation of a technology depends on some key factors, such as the in-house capabilities of an organisation. These factors and their implications are the subject of Section 4.3.

4.1 Methodology

During the interviews conducted with the key stakeholders for the first task of this project, a reoccurring requirement was that the data provided to and generated by the chatbot should reside within Europe. Some participants also expressed their objection against non-EU based vendors. While it is possible to set up data centres in Europe using the major technology vendors, they are all based in the US, with the exception of SAP (Germany). Specialised chatbot vendors are numerous both within and outside of the EU. The exercise is hence restricted to SAP and a selection of specialised EU-based vendors. Please note that only a limited selection of vendors is discussed, since this only serves as an initial assessment of the EU-based vendor landscape.

Online resources

To benchmark the vendors, a set of criteria tailored to the identified requirements (cf. Chapters 2 and 3) was established. This set is shown in Figure 1. Three criteria were flagged as mandatory: the language capabilities, the possibility for custom integrations and the inclusion of Natural Language Understanding/Processing (NLU/NLP) models, to have a multilingual smart bot that can retrieve information from various sources.

A first vendor assessment was made by consulting online resources. Some requirements are easily found online for all vendors, but many are not. Therefore, it was decided to rank the vendors on the information found online, and pursue in-depth interviews with a limited selection of vendors.











Criteria:		FOUND ONLINE	MANDATORY CRITERION
	Language Translation: Is there support for language translation? How many and what languages are supported?	✓	★
	Custom Integration: Is it possible to integrate external services?	✓	★
	Hosting: Does the vendor or the client handle the hosting? Is on-premises or private cloud hosting possible?	?	
	NLU/NLP: Does the vendor offer NLU/NLP models?	✓	★
	Speech: Is there support for speech-to-text?	✓	
	Cost: How much does the vendor charge? Is it subscription or load based?	✗	
	Support: What is the vendor support like? What are their responsibilities and availabilities?	?	
	Security & Availability: What are the security measures present? How does the platform respond to failure?	?	
	Analysis: Is there a metrics and logging dashboard for performance analysis?	?	
	Platform: Is the platform open-source? Are there custom code capabilities? Is the platform built using a cloud vendor?	?	

Figure 1: Vendor comparison - key criteria

The outcome of the initial assessment is shown in Figure 2. The included vendors are the ones for which sufficient information could be found online. The five first criteria of Figure 1 are included in this comparison, along with the country and the size of the vendor (number of employees). The hosting criteria require specific attention: some vendors allow only for hosting by them, while others (mainly open-source solutions) only offer hosting by the client. There are some that offer both options. In case hosting by the vendor is an option, this comparison illustrates which platform is used.

	VENDOR	COUNTRY	SIZE	MULTI-LANGUAGE SUPPORT	CUSTOM INTEGRATIONS	HOSTING*		NLU/NLP	SPEECH
						By Vendor	By Client (on premise)		
	Mindsay	France	50-100	✓	✓	✓	AWS	✓	✓
	CM.com	Netherlands	500 +	✓	✓	✓	Own platform	✓	✓
🔍	Boost.ai	Norway	100 +	✓	✓	✓	AWS	✓	✓
🔍	Cognigy	Germany	100 +	✓	✓	✓	Unknown	✓	✓
🔍	SAP (Recast.AI)	Germany (France)	100.000 + (?)	✓	✓	✓	AWS	✓	✓
	Clustaar	France	< 50	✓				✓	✓
🔍	Omnibot	Germany	< 50	✓	✓	✓	Own platform	✓	✓
🔍	Rasa	Germany	150 +	✓	✓			✓	✓
	Tock	France	open-source community	✓				✓	
	Do Your Dream Up (dydu)	France	< 50	✓	✓	✓	Unknown		✓

Figure 2: Vendor comparison - preliminary results

In agreement with eu-LISA, the following selection criteria were applied:

- Eliminate vendors with lacking functionalities
- Consider mainly vendors with sufficient capacity (number of employees), to increase the level of support and reduce the risk of insolvency
- Prioritise solutions that offer at least hosting by client, and preferably hosting by vendor as well.

Four specialised vendors were retained: Boost.ai, Cognigy, Omnibot and Rasa. SAP is also added to the shortlist, since this technology is also explored in the context of the Digital Visa initiative.

Vendor interviews

The ten criteria shown in Figure 1 formed the basis for a detailed interview with the vendors. Upon request of eu-LISA, the open-source and hosting aspects of the solution were emphasised in the questionnaire. The retained vendors were then contacted for an interview session. The questionnaire was shared upfront and is added for reference in Annex A. Omnibot did not reply to the request for information.

4.2 Chatbot vendor analysis

Annex A offers a detailed summary of the interviews. The purpose of this section is to highlight some context, the key functionalities and differentiators of each vendor. The vendors are ordered by size, from large to small.

4.2.1 SAP Conversational AI

SAP acquired Recast.AI, a French company focussed on NLP and chatbots, in 2018. This marked the start of the roll-out of the SAP Conversational AI platform⁷, which is a low-code development environment.

⁷ <https://cai.tools.sap/>, more information on <https://shorturl.at/ILTY4>.

The solution can handle multiple [languages](#), categorised into different tiers. The *advanced level* languages (English, French, German and Spanish) allow for all functionalities. Fifteen other languages are available at *standard level*. This means that some features (sentiment analysis and *gold entity recognition* such as phone numbers, nationalities, duration ...) are not available in these languages. Speech technology can be included through an external service such as Amazon Alexa or Twilio.

SAP positions three key differentiators of their solution compared to the competition:

- Advanced monitoring capabilities
- Advanced NLP capabilities, which is the core of the chatbot
- Smooth integration with other SAP services

The hosting of the solution is always done by SAP, on a Cloud Foundry platform-as-a-service. The solution is always connected to an AWS data centre (in Frankfurt). Other cloud providers (Azure, Alibaba Cloud) will be supported in the future. The chatbot solution is therefore not hosted on the SAP Cloud Platform (SCP). The data can therefore reside on the client's premises, which is a key technical requirement.

Conclusion: while SAP has a performant, multilingual, no-code platform with easy integration of other SAP services, it does not offer hosting by the client and does not allow for data storage in a client's database. This renders SAP Conversational AI a less suitable candidate for the VisaChat project.

4.2.2 Rasa

Rasa is an American-German company founded in 2016 (148 employees). It has an "open core" business model, meaning it offers an open-source stack of machine learning libraries (Rasa Open Source), a free but proprietary code base (Rasa X) and a subscription-based enterprise-grade platform (Rasa Enterprise) for designing and monitoring the solution.⁸

The key differentiator between Rasa and other vendors is indeed the open-source technology, and the flexibility, transparency and ownership that comes with it. It implies that clients have the possibility to customise the solution even to the extent that they can pick the NLP models and tune all the parameters. Moreover, the solution will always be hosted by the client, on a cloud or on-premise environment, as long as Kubernetes⁹ can be run. The data also resides on the client's infrastructure and is fully owned by the client.

Rasa enterprise encompasses all services on top of the NLP core: conversation monitoring, graphical user interface (GUI) for the design of the chatbot. Rasa also provides support, depending on the selected support plan. The enterprise comes with a yearly subscription and the cost depends on the support level and computational nodes deployed.

The flexibility and ownership also come with a cost. Rasa proposes to have teams of four to ten employees to manage a chatbot, including a back-end developer, data scientist, product manager and DevOps engineer.¹⁰ Another potential downside is that speech and translation services are not natively supported. Rasa instead proposes to train a model per language, since this will result in higher accuracy (see also Section 4.1.4 of D1.01). For speech capabilities, any external API-based speech-to-text (STT) service can be integrated.

Conclusion: Rasa has a unique "open core" business model, that encompasses an open-source NLP core of the chatbot. This increases the flexibility, transparency and ownership by the client, which is a good fit with the technical requirements. It however also implies that clients have to free up internal resources for this.

⁸ <https://rasa.com/>

⁹ Kubernetes is an open-source system to automate the deployment and scaling of software applications.

¹⁰ <https://rasa.com/blog/recipes-for-building-conversational-ai-teams/>

4.2.3 Boost.ai

Boost.ai is a Norwegian (Norway is not part of the EU, but is part of the Schengen Area) chatbot company founded in 2016 (115 employees). They offer a low- to no-code platform for the development of conversational AI, which has mainly been implemented at financial services and public sector clients.¹¹

The solution covers most of the dominant European languages, but more can be trained on request. Additional languages or speech-to-text can be brought in via integration of external services. Boost.ai puts its monitoring tools forward as a key differentiator. The tool offers a conversational flow analysis (to see which steps users take and where they become idle), analysis of the model's understanding and new intent suggestion based on what the chatbot failed to answer. Boost.ai also offers task management, which is convenient for solutions with a lot of stakeholders such as the one at hand.

The hosting can either be done by Boost.ai on (European) AWS servers or by the client on their premises. While it is possible to maintain the data on client's premises, Boost.ai recommends to store the data on AWS servers. Only then can the monitoring tool be applied.

They offer two pricing plans: Standard and Enterprise. Standard has a lower monthly fee but comes with a limited number of languages and has a higher cost per conversation compared to the Enterprise plan.

Conclusion: Boost.ai's low-code environment offers extensive monitoring and task management features. The hosting of the solution is flexible, but opting for on-premise hosting disables the monitoring features.

4.2.4 Cognigy

Cognigy was also founded in 2016, with headquarters in Düsseldorf, Germany (100 employees).¹² Their platform allows client to build their own chatbots, with a high degree of customisation possibilities..

Cognigy's platform is technology agnostic, in the sense that it offers clients the possibility to use multiple technologies. For example, Cognigy has a proper NLP engine, but also allows for seamless integration of other engines such as Google Dialogflow, Microsoft LUIS, IBM Watson and Rasa. Moreover, clients could in theory develop their own engine and integrate this in the Cognigy framework.

Similarly, other features such as the front-end, language and speech-to-text can be integrated. For the latter, Cognigy has a Voice Gateway, which enables to easily include this option. For languages, Cognigy's engine supports over 100 languages, with 20 languages for which entity recognition is available. Moreover, external translation services such as DeepL can be seamlessly integrated.

Clients can decide whether they would like to host the solution themselves, or let Cognigy host the solution. Kubernetes should be supported on the client's infrastructure. If this is the case, the chatbot solution can be automatically or manually scaled to match the load. Additional services can be integrated via API calls, for which Cognigy can also provide support. The pricing for Cognigy is subscription-based, with the number of conversations being the main driver. The cost is based on the estimated number of conversations per year, and additional costs are incurred when this number is exceeded.

Conclusion: Cognigy offers a high degree of flexibility and customisability, even for the chatbot's NLP core. The seamless integration of external services and multiple hosting possibilities also add to this. For languages, both a supervised and unsupervised (leveraging machine translation services) approach are supported.

¹¹ <https://www.boost.ai/about>

¹² <https://www.cognigy.com/>

4.3 Recommendations

The first conclusion of this initial assessment of the EU-based vendor landscape, is that it is viable to use EU-based vendors to support the creation of the visa chatbot. This section explains how the final vendor can be selected. Note that these considerations are also applicable to other vendors that were not interviewed.

Since every vendor claims to have a very performant NLP engine, the actual accuracy of the models is a difficult criterion to consider at this stage. However, the comparison shows that the available products differ quite a lot in approach. Therefore, the optimal choice of technology rather depends on the approach followed. From this limited study, a spectrum can be identified on which the interviewed vendors can be positioned.

On one side of the spectrum, solutions such as SAP Conversational AI offer a **fully managed service**. This implies that the hosting is done by the vendor. The underlying models are all integrated in a user-friendly platform, which allows for no-code development. Therefore, the workload to develop a chatbot is greatly reduced. A data set should be imported, and the conversation flow should be designed using drag-and-drop functionalities. At most, some parameters such as confidence thresholds should be tailored. The code is typically not open-source.

On the other side of the spectrum lie the **custom solutions**. The absolute custom solution does not leverage any vendor. Instead, the organisation takes up all tasks that come into play: development and training of the model, design of the chatbot conversation flow, design of a user interface, setting up the database, creating monitoring tools, configuring the hosting infrastructure etc. However, some aspects such as the model development and interface design require extensive work, while vendor services can easily be used for this. For example, Rasa provides a stack of models that can be used, without reinventing the wheel. The enterprise features of Rasa are adding aspects of a managed service, albeit optional.

Cognigy and Boost.ai are placed in between Rasa and SAP Conversational AI on this spectrum. While both solutions offer both hosting by the vendor and by the client, hosting by the client disables the extensive monitoring capabilities of Boost.ai's solution. Moreover, Cognigy is more customisable, since it allows clients to choose the NLP engine (including Rasa and custom engines) and language approach.

The spectrum and positioning of the interviewed vendors is visualised in Figure 3. Note that there is no ideal place on this spectrum, but this rather depends on the use case at hand. Given the technical requirements and key stakeholder considerations, it is assumed that the best fit for the visa chatbot is rather positioned on the spectrum towards custom solutions. For example, hosting by the client and open-source technologies are strong preferences of the stakeholders that are found on this end of the spectrum. The definitive choice of a vendor will depend on a set of parameters, such as:

- **Available resources** at the managing IT organisation (eu-LISA): if there is room to assign an extensive team, technologies such as Rasa might be preferred.
- **Hosting requirements**: after signing a non-disclosure agreement (NDA), the vendor solutions should be analysed to determine which technologies can be leveraged given the available infrastructure.
- **Security**: assessment can be based on the vendor's CAIQ questionnaire ([example](#)).
- **Costs**: the final decision should also depend on the cost.



Figure 3: Spectrum of chatbot vendors

5 Target architecture

The architecture of an IT solution can be regarded as the overall design of its computing system. It consists of different components and explains the hardware, software, access methods and protocols used throughout this system. Architectures can either be constructed using generic components, or can be designed for a specific technology. The latter makes the architecture more concrete, but also assumes a technology choice. In agreement with eu-LISA, it has been decided to opt for a technology-agnostic target architecture.

5.1 Layered architecture representation

A typical architecture representation, also used at eu-LISA, is the three-layer architecture. It consists of three layers, which can be physically isolated (if so, they are called ‘tiers’):

- The **presentation layer** is the user-facing layer of the architecture. Users can typically interact with a system through an interface (web page, Graphical User Interface). In the context of a chatbot, this will of course be the conversational chat window. This layer is typically constructed using HTML, CSS, JavaScript or data visualisation tools.
- The **application layer** (also *business layer* or *logic layer*) forms the core of the solution. In abstract terms, any IT solution can be regarded as something that generates an output after a user input. This layer contains the logic on which actions should be triggered by a user input. It is typically developed using Python, Java and the likes.
- The **data layer** provides the data for the output of the IT solution. After receiving a user input, the application layer will communicate to the data layer to retrieve the required piece of information. It is passed back via the application layer to the end user. The data layer is typically a database with either structured or unstructured data.

In this type of set-up, all communication between presentation layer and data layer runs through the application layer. As an example, consider the weather app on smartphones. Through the app’s interface (presentation layer), users can type a specific location for which they want to know the weather. The application layer translates this request into a technical query on the database containing the weather for all locations (data layer), for a fixed amount of days. The results of the query will contain the weather, but in a technical format. The application layer will process this query and render it to a user-friendly output, shown on the user interface.

Aside from these layers, there are also transversal components of the architecture that should be considered. This involves security, monitoring and operational management. Finally, the architecture can also be demonstrated by means of its related conceptual data model.

5.1.1 Presentation layer

The presentation layer consists of user interface(s) and potentially authentication. These components are securely connected with the other layer by means of a connector or event broker. This is shown in Figure 4.

Front-end (user interface)

The chatbot should first and foremost be available as a widget on pages of the Member States and the European Commission. Luckily, chatbots are usually very easy to integrate on websites. If a **vendor technology is leveraged**, and if the websites are developed using HTML, a simple script can be added that can be generated from the chatbot builder platform:

```
<script src="source_code.js"></script>
<script> Chat(" endpoint-id"); </script>
```

Web developers of the Member States should merely ensure that the widget is included in a consistent location (e.g. bottom right). All interviewed vendors confirmed that the lay-out of the widget is customisable.

Styling the chatbot should preferably be done at central level, to show that the chatbot is not only a national product, but covers the whole Schengen area.

In case of **custom developments**, the deployment on websites requires more efforts. First, all interface components should be developed using JavaScript (libraries). For example, the input box, replies, special message boxes, buttons need to be developed. Then, the JavaScript code should take the text from the input box, send it to the business layer, and then provide the response again in the preconfigured interface components. Finally, the JavaScript code can be embedded in the website, similar to the approach outlined above.

Authentication

While the first version of the visa chatbot does not require authentication, this might be required in the future, after go-live of the EU Online Visa Application Portal. As explained in Section 3.3, the authentication method (e.g. MFA) should be common for both the portal and the embedded chatbot.

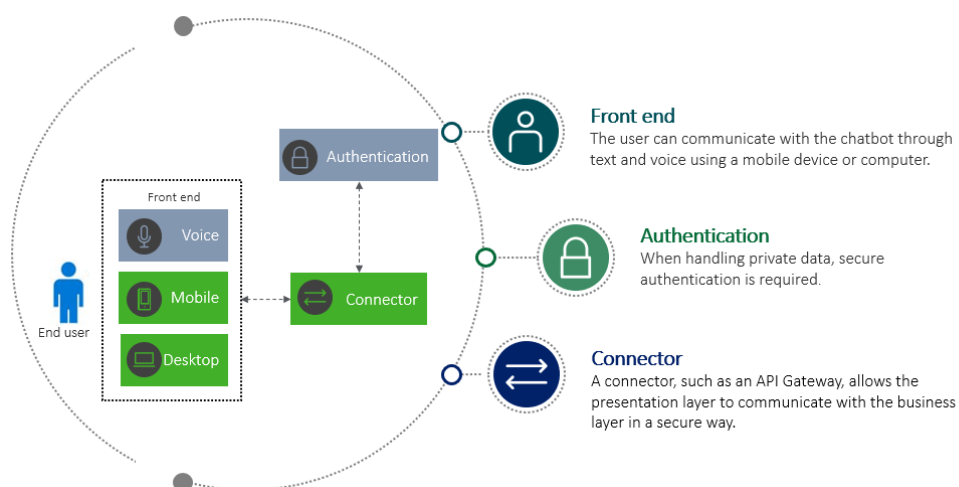


Figure 4: Target architecture - presentation layer

5.1.2 Business layer

The business layer can be considered as the 'brain' of the chatbot, where the calculations take place. It consists of an NLU engine to understand the user's intent (if needed after translation or speech transcription) and a search component that takes a corresponding action. This is visualised in Figure 5.

NLU engine

The artificial intelligence are Natural Language Understanding (NLU) algorithms that try to map the incoming query, which can be free-text, to the existing data set of encoded questions and answers. In its simplest form, the algorithm would search for matching words between the formulations of the encoded questions, and the incoming query. Recently, such techniques are replaced by emerging transformer-based techniques, that also consider the context of the words in the query. Using open-source technology, custom techniques can be developed, although these might require training. Vendors typically offer out-of-the-box NLU algorithms.

Search

After understanding the intent of the user by means of NLU technology, the chatbot will perform an action. Typically, the bot will provide the answer that is encoded in the question base. In some cases, the bot will first ask follow-up questions to the user. It is however also possible that the bot should perform more complex actions, such as retrieving information from a database, referring to a support desk, processing a transaction etc. Leveraging chatbot vendor technology, their platforms typically come with such predefined actions.

Speech and translation services

The NLU engine and data set are developed in at least one language (e.g. English), and can only process written text. If users communicate using speech, or use a language for which no data set is available, the incoming query needs to be processed first. For this, existing speech-to-text (STT) or translation services should be leveraged. It is infeasible and inefficient to develop such services in-house merely for a chatbot use case. Typically, the interaction with external services takes place via APIs. In some cases, additional licenses might be required.

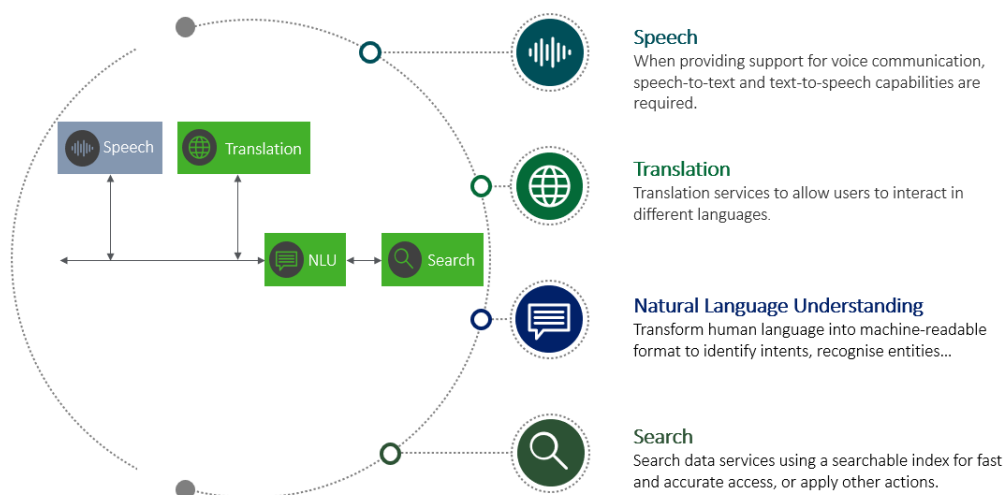


Figure 5: Target architecture - business layer

5.1.3 Data layer

The chatbot can only provide responses that are contained in the data layer, which can consist of various sources. In order of importance, the chatbot should consult the question base that is manually supervised. If needed, for example to provide status information, the bot could consult other systems. In some cases, it might be useful to scrape information from webpages, such as opening hours of embassies (raised during the proof of concept). These sources might require processing by means of an ETL. The whole data layer is represented in Figure 6.

Question base

The principle data source for the chatbot will be a question base that consists of frequently asked questions (FAQs), with different formulations of each FAQ and an encoded answer. Potentially, a question has multiple answers depending on some parameters. The question base is supervised and will be maintained by all Member States, as will be outlined in Deliverable D4.01, the Solution Delivery report. The size of this data source is negligible. The question base should be managed in a Content Management System, where Member States can assess and update the responses that are encoded in the chatbot without having to code.

Interoperability with other systems (in the future)

A second source is that the chatbot could retrieve information from other systems, such as the central Visa Information System or the data sources of the (yet to be developed) Online Visa Application Platform. As outlined in Deliverable D1.01, there should be no integration with Member State systems. During discussions with eu-LISA, interoperability with other systems managed by eu-LISA is feasible if the chatbot is hosted on-premise. This data source is only required if the chatbot should provide status information.

Web information (optional)

The chatbot could search through web pages or documentations for specific information that is too work-intensive to maintain in the question base. This requires search algorithms that are used in e.g. Google's technology, to find information from a large set of documents or web pages. The documents or web pages can be scraped regularly, which again results in limited data size. Examples include opening hours of embassies, addresses, cities and their corresponding countries etc.

ETL

Typically, data should be processed before it can be used in a software application. The process to extract, transform and load data is referred to as 'ETL'. It can be expected that the question base requires little to no processing, while the data from the other systems or scraped data might require more processing.

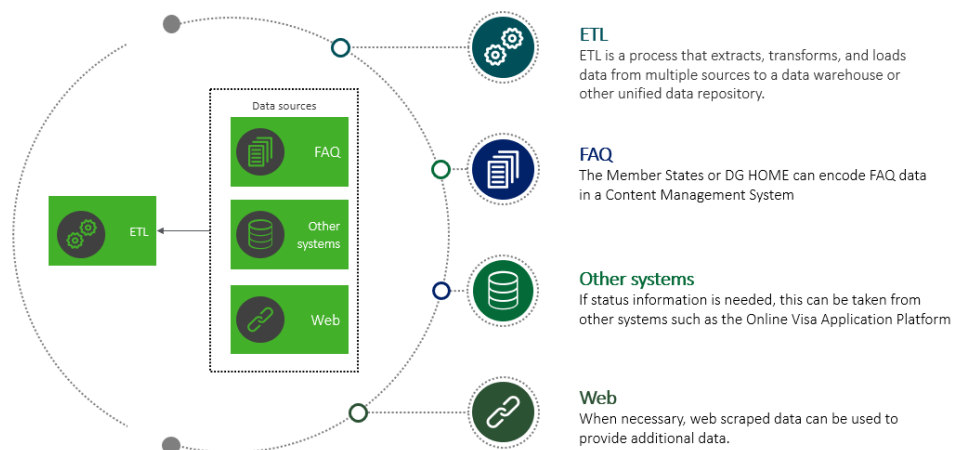


Figure 6: Target architecture - data layer

5.1.4 Other components

The additional components to support the chatbot maintenance are feedback and logging, monitoring and testing and debugging, as shown in Figure 7. Finally, there is the security aspect that is transversal to all components and (importantly) the connections between them.

Feedback and logging

An essential feature of the chatbot is continuous improvement. To achieve this, the conversations between bots and users (and potential feedback) should be logged. This allows for detecting when users stop engaging, which questions are answered inaccurately etc. The log files should be stored in a NoSQL database, since this data is not structured (see also Section 5.1.5.2).

Monitoring

To retrieve actionable insights from the logged conversations, a monitor tool should be created for measuring the chatbot's performance. If the chatbot fails in a particular part of the conversation, it has to be modified. This can either be achieved by changing the data set or by updating the models or chatbot features.

Testing and debugging

If new models or features are developed, they should be tested and debugged. A Continuous Integration / Continuous Delivery (CD/CI) pipeline is a series of steps that must be conducted before delivering a newer version of a software application. The architecture should support DevOps tools such as these pipelines.

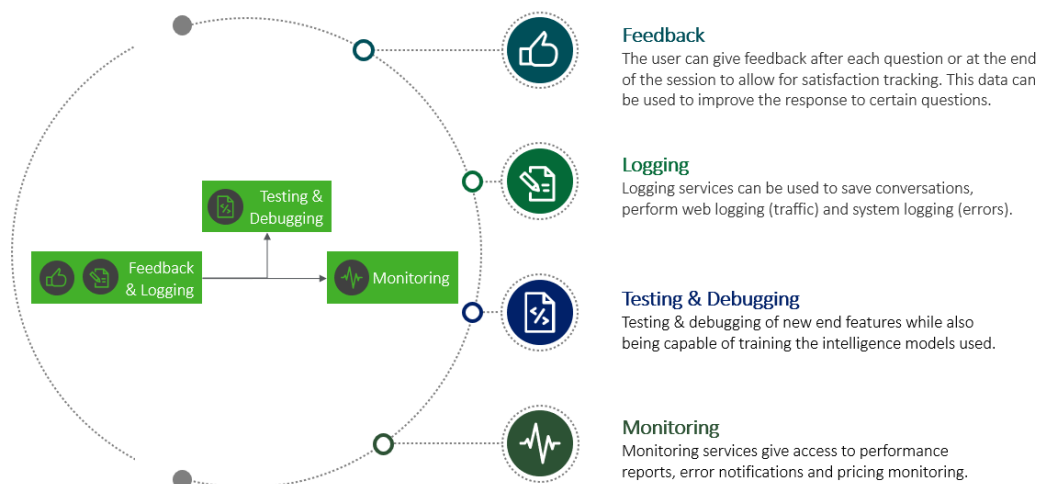


Figure 7: Target architecture - other components

5.1.5 Conceptual data model

The assembled target architecture can also be presented in a conceptual data model. Below subsections provide two data models. The first shows how questions can be answered. A second shows how the data can be stored for monitoring purposes. Note that speech and translation services are not included in these models. It is assumed user input is transcribed or translated *a priori* to a written query in a question base language.

By using a vendor, the data is stored by default in predefined data models. There is no need to set up a database and connect this to a monitoring tool, since this comes out of the box. If desired, all log files can be copied and additional analyses can be made.

5.1.5.1 Conceptual data model and processes— answering questions

The data model and related processes can be structured according to the layers introduced for the architecture. The explanation below is supported by a visual representation in Figure 8.

The core component is a conversation, which is a sequence of (one or) more user queries¹³ and chatbot responses. Every conversation is also linked to a webpage (or alternative channel), which should be tracked for monitoring purposes. The webpage also depends on the Member State parameter, as explained in D1.01. The language is assumed to be fixed throughout one conversation, by means of a language selection menu.

A user can make a query in two ways: by asking a free-text query, or by clicking on a button. This triggers a different logic in the business layer. In case of a free-text query, the NLU engine ('brain' of the chatbot) will calculate the similarity between the query and all intents (1). Then, the intent with the highest similarity will be retrieved (2). The intents, formulations and answers are all available in the question base that resides in the data layer. The corresponding answer depends on the parameters that are active at the moment of the query (3). If needed, the chatbot will first confirm the parameters. Finally, the chatbot will provide the answer as a response to the user (4).

The flow for a button query is more straightforward. The buttons are all linked to a unique intent {1}, {2}. The answer for the intent is provided, again considering the applied parameters {3}. This is then provided as a response to the user {4}.

¹³ In Figure 8, 'Query' refers to a message by the user. It is not to be confused with (SQL) queries on a database.

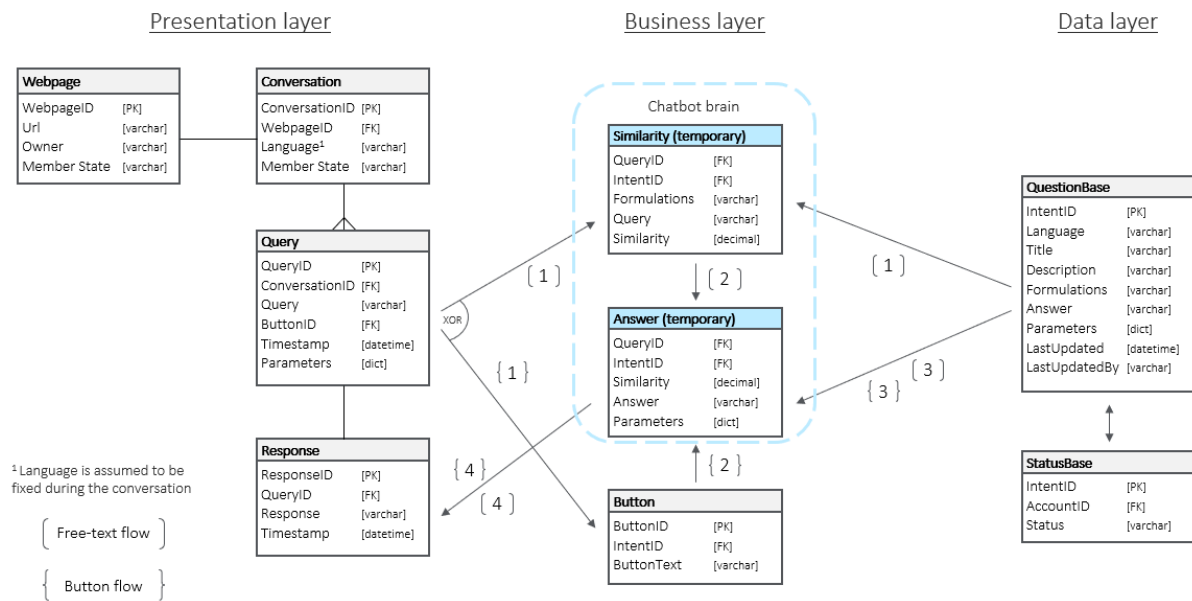


Figure 8: Conceptual data model - Answering questions

Note that the feedback mechanism can also be represented in the below schema. After the chatbot has provided a response, the user can click on a thumbs up or thumbs down button. This triggers a message by the bot, which in the case of negative feedback, should contain the description of the intents with high similarity (but not the intent with highest similarity, since this answer was already provided), so that the user can select one of these intents. The final rating mechanism can also be considered to be a button.

5.1.5.2 Conceptual data model – monitoring performance

Every conversation will generate a log file. This log file contains all information related to the conversation, including the parameters, all the queries and the related answers. These log files can be used for manual inspection of each conversation, which is mainly useful to identify issues with particular conversation flows. In addition, it might be relevant to keep these log files for a certain period to deal with questions pertaining to the advice given by the chatbot. After the retention period, the log files should be removed, from a data storage optimisation and privacy standpoint.

The data from the log files should be stored in a data model that allows for aggregated views on the chatbot performance. A proposed model is shown in Figure 9. Central to the data model is the QueryIntent table, that contains information on every query and the intent to which it was mapped. In this model, three dimension tables are proposed to provide supplementary information: the conversation information, the parameter information and the intent information (question base). Using this model, it is possible to create an overview of many intents were answered successfully for a given Member State, or to track the evolution of overall conversation rating over time. These are just two examples, since the possibilities are nearly endless.

To illustrate that using vendor services speed up the monitoring, Figure 9 contains a partial screenshot from the insights generated by Cognigy's dashboard. This is created on-the-fly, as conversations are taking place. Similar services are offered by all other interviewed vendors. Usually, the services also allow to create additional visualisations using custom tools.

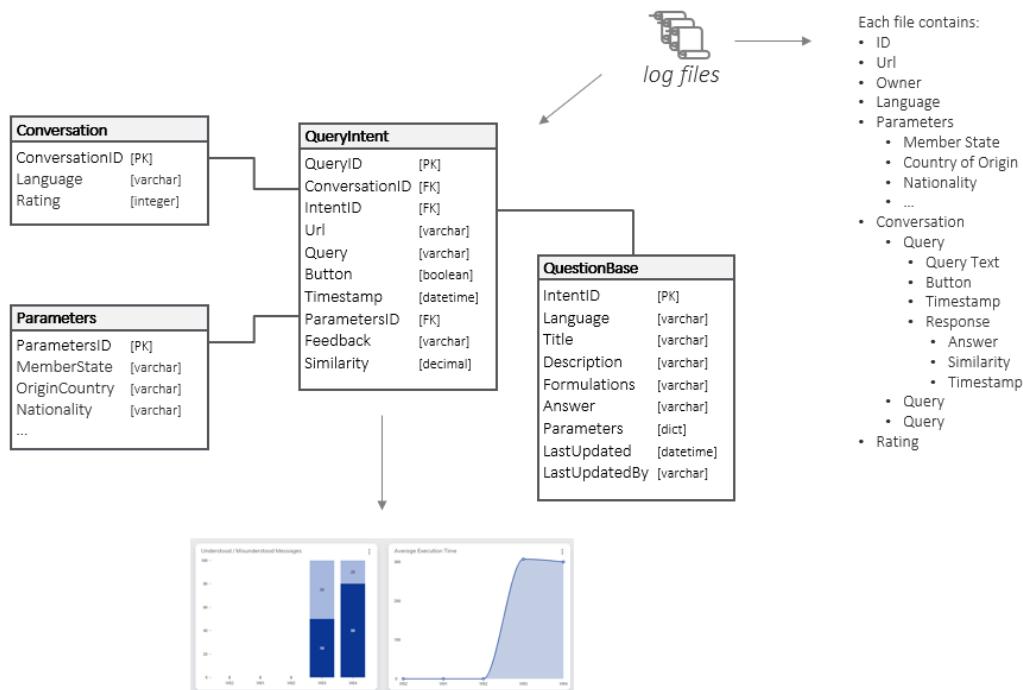


Figure 9: Conceptual data model - Monitoring performance

5.2 Estimation of hardware requirements

Aside from the different components, it is important to know how much capacity should be made available for a performant chatbot, which depends on the expected usage. Sections 5.2.1 to 5.2.3 offer a theoretical reflection on the expected requirements, while Section 5.2.4 looks at a concrete example technology.

5.2.1 Expected usage

The interviews conducted in the context of Task 1 showed that for the Netherlands, the call centre is contacted for questions in roughly 10 % of the applications. Deliverable D1.01 shows that, through extrapolation, approximately 1.7 million questions arise yearly at the combined Schengen Member State administrations. In addition, the External Service Provider (ESP) VSF Global indicated that they receive almost a tenfold of this (of which a large part overlaps with the proposed scope of the visa chatbot).

The engagement rate is calculated as the ratio of users that use the chatbot vs. page visitors. As a rough estimate, chatbots should be able to assist 80 % of its users. In theory, the chatbot should therefore be able to assist 1.36 million users that are now contacting the Member States. If one tenth of the ESP customers engages with the bot¹⁴, this adds another 1.7 million users, leading to 3 million in total. This can however only be guaranteed if all Member States are involved, and if potential users find their way to the chatbot, which cannot be expected. 3 million yearly users is therefore considered as a safe upper limit, for the foreseeable future.

¹⁴ This estimate is kept to a fairly low share of applicants, since ESPs such as VFS are also developing chatbots. If these are performant, users will not engage with the official EC visa chatbot. However, if the ESPs make reference to the official chatbot, the potential for the chatbot greatly increases.

The expected usage based on this thought experiment can therefore be projected to 3 million users yearly, or more than 8 thousand daily, in the future. The usage in the first months and years of the chatbot will be lower. This is visualised in Figure 10.

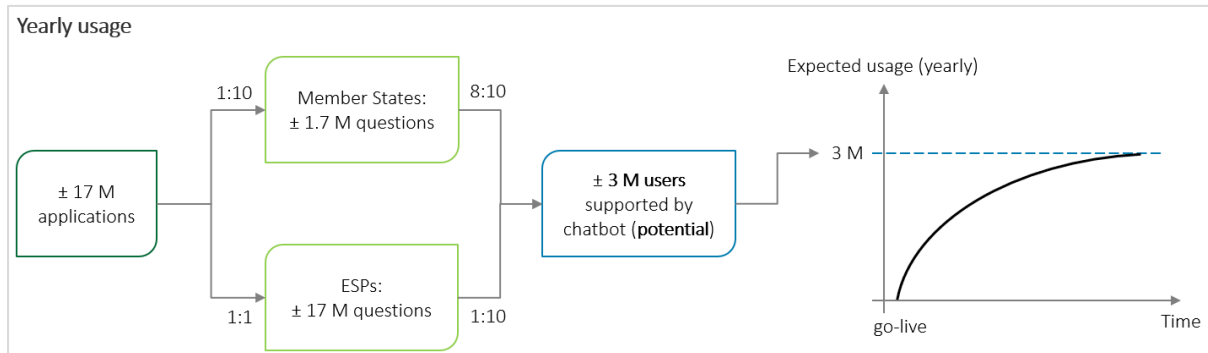


Figure 10: Estimated usage of the chatbot

5.2.2 Data storage requirements

Question base, scraped pages and other data sources

The storage of the question base is negligible. For the proof of concept, the Excel file containing questions, formulations and answers in two languages was only 100 KB. The question base will therefore never exceed the order of MB.

Pages and documents can be scraped and saved in a format that can easily be processed, such as .json files. Typical webpages only yield 5 to 10 KB of text information. With one thousand scraped pages, this results in approximately 10 MB. Documents typically contain more text, but it is unlikely that the total set of scraped pages and documents will exceed 100 MB.

Other data sources are disregarded for this exercise, since it is assumed that this data is already stored. If views need to be created from the data sources, these will yield some space. This needs to be considered when it is clear how the information from e.g. the EU Online Visa Application Portal will be included.

Logged conversations

For performance monitoring, all conversations will be logged. A logged conversation is typically a .json file containing the session ID, timestamp, language and collection of user queries and chatbot answers. Conversation log files are typically of the order of 5 to 10 KB (learned from the proof of concept). This means that on a daily basis (when accounting for 8000 requests), 40 to 80 MB of conversation logs are generated. After a month, approximately 1 to 2.5 GB of conversations are recorded.

However, after some time, it does not make sense to keep all individual conversations. The monitoring tool should show aggregated figures for a long period of time (i.e. multiple years), to show the evolution of the performance. Having access to log files of the past month(s) is however useful, for investigating particular cases where the chatbot fails. Therefore, it is recommended to set up a data retention policy. For example, the conversations could be logged at a detailed level for two months. Afterwards, only the information required to show aggregated figures are retained. This means that there is up to 5 GB of recent, detailed data. In addition, there will be tables or views that contain merely the data required to create the visualisations. Another 5 GB should suffice for this.

5.2.3 Compute requirements

Transformers are deep learning models that are increasingly interesting for NLU and NLP applications. Like all deep learning models, they require significant computing power to be trained, which is preferable done on

Graphics Processing Units (GPUs) instead of Central Processing Units (CPUs). Creating such a model from scratch and training it can require a huge number of GPUs and can last for days (for example, GPT-3 has more than a billion parameters). Therefore, it is recommended to use pre-trained models, either offered by chatbot vendors or available open-source.

If pre-trained models are chosen, the compute requirements are not extensive. If the 8000 daily conversations on average consist of 10 messages, and are spread across 12 hours, there are approximately 1.85 messages per second. Some messages, such as the greeting message or button clicks, can be processed without additional computing power. For the messages where NLU is required, some CPU will be spent on matching the query with existing questions, but the amount depends on the vendor technology. In the proof of concept, processing a request required at most 70 MB of RAM. This means that with a typical amount of 8 GB RAM, 114 queries can be processed simultaneously, which should be sufficient.

5.2.4 Reference: requirements specified by vendor

For reference, the minimum/recommended system requirements for on-premise hosting, as shared by Rasa, are listed in Table 3.¹⁵

Table 3: Minimum/recommended requirements for on-premise hosting

Operating system	<i>Linux distributions:</i> <ul style="list-style-type: none">- Ubuntu 18.04/20.04- Debian 9/10- CentOS 7/8- Red Hat Enterprise Linux 8
vCPUs	Minimum: 2 vCPUs Recommended: 2 – 6 vCPUs
RAM	Minimum: 4 GB RAM Recommended: 8 GB RAM
Disk space	Recommended: 100 GB disk space available
Server ports (open)	<ul style="list-style-type: none">- 22 (SSH) for SSH access- 80 (HTTP) for web application access- 443 (HTTPS) for web application over HTTPS access (optional)

Rasa X or Enterprise can be installed on-premise via Rasa Ephemeral Installer (REI) for local or test installations, or via Rasa Helm charts for high availability and scalability. When opting for Rasa Helm Chart installation, the Kubernetes cluster should support the resources (pods) as listed in Table 4. The *rasa-production* and *rasa-worker* requirements depend on the model size and number of users. Since the installation is meant for highly available and scalable solutions, it is assumed that this can be achieved with the requirements of Table 4.

Table 4: Requirements for Rasa Helm Chart (production environment) installation

Deployment name	Description	CPU	Memory
rasa-x	Rasa X backend, UI and HTTP API	1	1 GiB
event-service		2	1 GiB

¹⁵ The recommended system requirements for on-premise hosting of other chatbot solutions was not found online or shared during the interviews, and is therefore not included in this report. Rasa's recommended requirements, and an installation manual, can be found here: <https://rasa.com/docs/rasa-x/installation-and-setup/install/docker-compose/#docker-compose-manual-install>

rasa-production	Rasa Open Source service running a trained model, used for parsing intent messages and predicting actions in conversations with user over the input channel or Rasa X UI	2	2 GiB
rasa-worker	Rasa Open Source service used for background tasks such as training models	4	4 GiB
nginx	Reverse proxy used to reroute requests to the different services	0.2	200 MiB
app	Custom action server	0.5	200 MiB
duckling	Parse text into structured data via entity recognition	0.5	200 MiB
postgresql	PostgreSQL database service	1	250 MiB
rabbit	Message broker used to transmit conversation events between <i>rasa-production</i> and <i>rasa-x</i> .	0.2	250 MiB
redis	Service acting as a persistence layer for tracker (conversation) locks, and a multi-purpose in-memory cache.	0.2	250 MiB

Note that the above requirements assume one *rasa-production* model (with which the users interact) and one *rasa-worker* (which trains the model). With Rasa enterprise, it is also possible to deploy a development or other environment. A different model can then be developed, trained by the worker, and eventually promoted to production. For development (and other) environments, it can be assumed this will not exceed the requirements for the *rasa-production* pod. However, each version of the model can only have one *rasa-worker*. Therefore, the requirements listed above should at least be provided in threefold, to deploy and publish different models.

Secondly, note that Rasa X encompasses the supporting tools for the chatbot, such as the monitoring tools and content management system. For example, through Rasa X, organisations can assess which intents should be added to the chatbot to improve user experience, based on past conversations. The requirements for these tools are hence assumed to be included in Tables 3 and 4.

Finally, these requirements were described for Rasa, since this was the most documented technology. During the vendor selection step, technical experts should be involved to assess whether the vendor product can be hosted on eu-LISA's premises.

5.2.5 eu-LISA data centres

In terms of operating system, eu-LISA has shared that their data centres use Red Hat OpenShift Container Platform, a platform-as-a-service built around Linux containers orchestrated and managed by Kubernetes on a foundation of Red Hat Enterprise Linux. This is therefore expected to meet the requirements in terms of Operating System. Cognigy and Boost.AI also remarked that the on-premise platform should support Kubernetes as the key (and almost only) requirements.

The other requirements are in general an extension of the requirements outlined in the previous sections. When sticking to the recommendations of Table 3, it is therefore expected that the platform will be able to support the chatbot.

5.3 Summary of target architecture

In summary, the target architecture can be considered in terms of components from the one hand, and in terms of hardware requirements on the other hand. The different layers of Section 5.1 can be summarised into one diagram, shown in Figure 11.

The exact execution of the target architecture heavily depends whether a vendor solution will be leveraged. These solutions come with built-in interfaces, NLU models, actions, monitoring tools, content management systems, hereby covering all aspects of the architecture. With some vendors, it is nonetheless still possible to choose the NLU model that is applied, to maintain ownership over the solution. Building and training a custom model from scratch is not recommended, since this requires a huge amount of compute resources. If this is circumvented, the hardware requirements are in fact very reasonable compared to typical IT systems (as shown in Table 3).

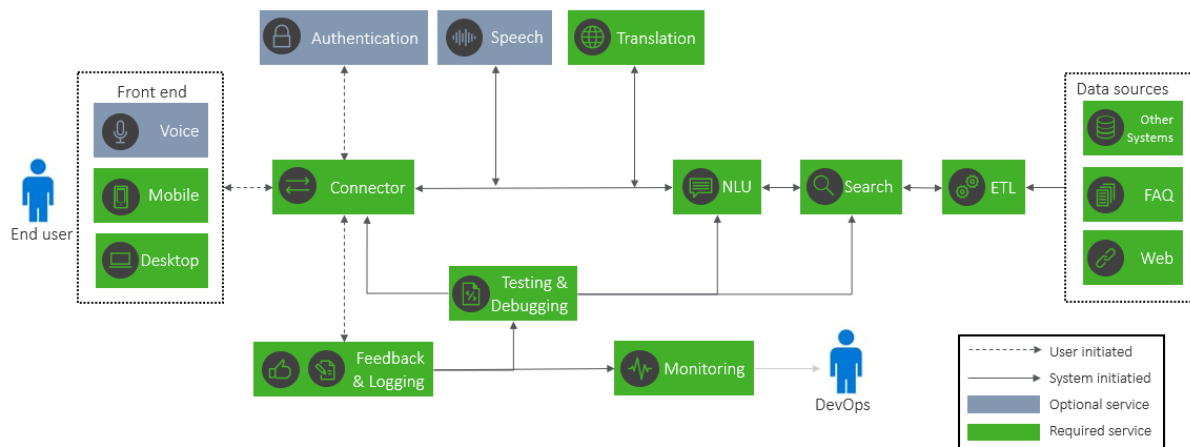


Figure 11: Target architecture - summary

6 Conclusion and next steps

This report elaborated on the target architecture in the context of the visa chatbot project. The starting point for the architecture are the business and functional requirements shared by the Member States, and the technical requirements discussed with eu-LISA. This is summarised in Sections 2.1, 2.2 and 3.6.

Important for the final solution is the technology selection. While it is in theory possible to develop all components of the target architecture from scratch, this is not efficient, since it requires a massive number of compute resources for training the models, and many additional developments for the user interface, content management system and monitoring tool. Therefore, it is recommended to leverage open-source (for the NLU algorithms) or vendor services (for the NLU algorithms and additional services).

This report explains that the EU-based vendor landscape is diverse, containing at least some vendors that meet all the requirements. After a very high-level assessment, four vendors were interviewed to gain insights on the similarities and differences between the vendors. This only serves as an initial assessment, since other vendors might also fulfil all requirements. From the vendor interviews, a clear spectrum of approaches emerged. Some vendors offer their chatbot product as a fully managed service, while others are more leaning towards highly customisable solutions, with more ownership on the client's side. The latter category is believed to be more in line with the technical requirements put forward.

Finally, a vendor-agnostic target architecture was presented in a three-layered representation (presentation layer, business layer, data layer). After introducing this representation, the different components of each layer were described. Many of these components are made readily available by vendors. The estimated usage of the chatbot is leveraged to calculate the hardware requirements. If pre-trained models are used, the requirements are very straightforward to meet. The estimation is backed up by the hardware requirements shared by a vendor.

Before further detailing the target architecture, a vendor should be chosen, or it should be decided to opt for a fully custom solution. After signing a non-disclosure agreement (NDA), the vendors can share the hardware requirements and configuration steps to mount their product on existing on-premise infrastructure. Once this is done, the chatbot can easily be deployed on any webpage. The main attention point will arise when integrating other IT systems, such as the central VIS or the EU Online Visa Application Portal data, to make sure sensitive data cannot leak from the chatbot.

Some of the questions concerning the ownership of data and maintenance will be further addressed in D4.01, which focuses on the operating model and solution delivery. Also, this deliverable will illustrate the backlog of features required for the development of a proof of concept, with a provisional implementation planning

Annex A - Chatbot vendor interviews

Below table shows the questions asked during the chatbot vendor interviews, along with the answers per vendor. Please note that Cognigy provided the answers in writing after the meeting. Any use of 'we', 'us' or 'our' refers to Cognigy. Cognigy's notes are reduced in size, since the answers are often lengthy.

Question	SAP	Cognigy	Rasa	Boost.AI
<i>Features – General</i>				
The chatbot should include both a guided approach (with buttons) as well as a free-text approach (with free-text). Does your product offer this?	Yes. While building the type of flow, questions can be asked through plain text or through MC/buttons	A chatbot conversation can consist of open questions, where the user can write his inquiry, as well as a guided approach where the user can click buttons, where a user can follow a predefined path within a conversation. https://docs.cognigy.com/ai/flow-nodes/flow-nodes-overview	Yes. There is a GUI to support the creation of buttons in the chatbot.	Yes
Does your product offer the ability to work with parameters (variables that determine the response on a question)? The parameters should only be asked when needed, and they should be saved per conversation.	Yes. Within the design, you determine the specific requirements you need before going to the next step. This can be personalised depending on the requirement. Named Entity Recognition is also present.	Yes. Answers to questions can be used to make a call (but not mandatory) to an third party application, and store the results in the context of the conversation for the entire conversation. Also, when an answer on a question was already give upfront, there is an option to not ask that particular question, as the answer is already in the context of the conversation." https://docs.cognigy.com/ai/flow-nodes/flow-nodes-overview	Yes	Yes, this can be through field input or named entity recognition.
What are the main features that your product offers? What are the key differentiators between your product and the competition?	Key differentiators: <ul style="list-style-type: none"> Advanced monitoring options Strong NLP engine 	Cognigy.AI is the leading enterprise conversational automation platform for building advanced, integrated conversational automation solutions for any chat or voice channel. The platform is an all-inclusive package delivering a highly flexible and customizable enterprise ready solution. Conversation channels such as Facebook and Alexa are connected to conversation flows that are designed and edited in the revolutionary Cognigy.AI flow editor. Built-in flow nodes allow dialogs, rich media and external integrations to be created without code, alongside an integration framework that allows open source custom tools to be developed and added to extend the platform's comprehensive array of existing features. https://docs.cognigy.com/docs/extensions https://docs.cognigy.com/docs/platform-overview	Rasa is open-source conversational AI. This offers: <ul style="list-style-type: none"> Flexibility: plug in different models, frameworks when needed Transparency Ownership Rasa is not aware of other EU vendors that offer open source technology	<ul style="list-style-type: none"> Low to no-code platform Strong NLU advanced free text understanding Conversation flow visualisation Message analysis: what the model recognises. New intent suggestion based on

Question	SAP	Cognigy	Rasa	Boost.AI
				what it couldn't handle previously <ul style="list-style-type: none"> Task management: highlighting conversations to update and assign to personnel/roles
Which technology stack did you use to develop your chatbot?	All the technology is developed in-house in the cloud. No external services are used.	Technology stack: <ul style="list-style-type: none"> React (Typescript) NodeJs (Typescript) Python Golang MongoDB Redis RabbitMQ Kubernetes Policy: Cognigy.AI is built on several micro-services. Around 80% of the services run on the Node runtime and are written in TypeScript. We use numerous open source components here, but adhere to a strict open source monitoring policy. A select set of services that are used for our NLP related services are written in Python and use the spaCy library, which also has an MIT license. A small fraction of our services run on the Go programming language and use no external libraries.	Predefined NLU chunks: spaCy (free, open-source), BERT, other transformers Rasa is a fully fledged open-source platform built on existing frameworks.	Development of the platform has been done bottom-up and all in-house.
Does your chatbot have the capability to communicate in speech (using voice)?	Speech channel using Alexa for example is possible.	Yes. Any TTS and STT services (not provided) can be enabled for the end user to increase accessibility for all users through the Voice Gateway. https://docs.cognigy.com/ai/endpoints/webchat/integrated-demo-page/#enable-speech-to-text-enable-text-to-speech	Yes. Although it is not within the core platform, any speech recognition front end can be integrated e.g. Deepgram, Amazon Transcribe, Twilio,....	Leveraged from a 3rd party chosen by the client. Benchmarking tool present to compare the different solutions.
Do you support transferring the user to a live agent?	Yes. You can set certain failure thresholds (emotion recognition after question failures for example) to enable fallback channels.	Yes. Cognigy.AI supports a handover to multiple different providers, like our own Cognigy Live Agent, Chatwoot, Salesforce, LiveAgent, Ringcentral and others. https://docs.cognigy.com/ai/tools/agent-handover	Yes	Integration with 3rd party solutions.

Question	SAP	Cognigy	Rasa	Boost.AI
	2 fallback channels already supported are Intercom and Singe. Currently no plan to add more fallback channels but since the platform is API driven, it is possible to add a custom fallback channel.			
<i>Features – Monitoring</i>				
Do you record conversations for a posteriori analysis? If so, in which format?	Yes. All-time conversation logs.	Yes. Cognigy.AI contains a built-in project analytics dashboard which displays key chatbot metrics to users within the platform. Conversation data is also available via the Cognigy Odata feed which can be used to import live conversation data to external visualisation tools such as Tableau and Power BI. A native analytics integration is also provided for both Chatbase and Dashbot.io for each endpoint that is deployed. https://docs.cognigy.com/docs/analytics	Core Enterprise feature: conversations are flagged, tagged in order to be reviewed.	Yes. All of the data can be extracted through APIs into CSV files.
Do you store questions made by users to which the chatbot was not able to reply?	Yes	Yes. Cognigy.AI offers a selection of tools to allow conversation designers to handle misunderstood, low-scoring or overlapping intent results. Optional confirmation sentences can be attached to intents to provide a prompt for users when a specified intent scoring threshold is not met. Disambiguation is another feature that allows multiple prompts for closely scored intents to be provided to users. The threshold settings are available for designers to customize for each project in the agent settings menu. https://docs.cognigy.com/docs/machine-learning#thresholds	Yes. In addition, problematic intents are flagged so they can be improved. Rasa follows a 'Conversation driven development' principle.	Yes
Which metrics of the chatbot usage can be tracked? How can they be accessed?	3 monitoring options: <ul style="list-style-type: none"> Pure usage data: basic analytics data such as amount of users etc Simple conversation log Training analytics basic on the dataset such as F1-score, confusion matrixes,...	Cognigy.AI has integrated analytics reporting functionalities, as well as an ODATA interface our customers can rely on to retrieve their raw analytics data. Our integrated analytics dashboard provides a range of key metrics to understand the usage and the performance of your bots. https://docs.cognigy.com/insights/cognigy-insights https://docs.cognigy.com/docs/odata-analytics-endpoint#integrations	Developers of the client have access to all typical data science metrics: analytics using histograms, confusion matrix...	
Do you extract insights from the chatbot's users? If so, how do	Options for dealing with personal data:	Contact Profiles can store information about the end-users of your AI and can be accessed by Flows and Endpoints. Contact Profiles can be used to store information persistently and personalize discussions with users.	Conversation logs are saved.	Personal data can be masked when it's coming

Question	SAP	Cognigy	Rasa	Boost.AI
you collect and store the personal data?	<ul style="list-style-type: none"> Option between storing the logs or not storing the logs. Restricting access to the platform and data. Text analysis: when you detect this kind of information, delete it before sending to chatbot or log. Removal of log files.	<p>You can manage contacts who interact with your AI's in Contact profiles where the contact data and transcript history is available to view. Please note that Contact Profiles can be completely or partially disabled.</p> <p>https://docs.cognigy.com/docs/contact-profiles</p>		in through the GDPR rules.
Features – Integration				
Can the chatbot be integrated in most of the websites or only with specific frameworks? Is it a cross-platform solution (smartphone compatible)?	Yes, the chatbot can be integrated in webpages via JavaScript. In addition, it is possible to integrate the bot in mobile apps, for example through iOS SDK.		Mainstream platforms (Messenger etc.) are supported but generic web sockets are also possible. If additional channels should be included, Rasa is happy to assess this together with their clients.	Omni-channel support
Is it possible to connect external APIs to allow the chatbot to retrieve information from a backend system or database?	Yes		Yes, after triggering an chatbot action, database access can be performed. It is advantageous (from a network security perspective) if the system is available on the same infrastructure as the chatbot, but it is not a requirement. The RASA action server can trigger actions via Python. Connection is typically done by REST APIs and web socket connections.	Yes

Question	SAP	Cognigy	Rasa	Boost.AI
Is it possible to integrate external services within your chatbot, to complement the built-in services? If yes, can it be done by request and implemented by external developers?	Yes. External services can be connected using the API.		Yes, external services can be integrated by the client.	Yes
Features – Security				
Which encryption protocol do you use? Is it possible to change the protocol if required? Which data is encrypted?	The security protocols can be found here .	Data traveling between the customer and Cognigy.AI software is always encrypted using HTTPS. Normal HTTP traffic is not allowed and connections using HTTP get redirected to HTTPS. Furthermore, Cognigy.AI encrypts certain data (such as credentials, etc) - in addition to the encryption on platform/infrastructure level - using AES 256. Access to the Cognigy.AI platform is password protected with infrastructure access to virtual machines secured using SSH-key based authentication.	Rasa applies the following security protocols: <ul style="list-style-type: none"> • Kubernetes best practices are followed • Data at rest can be encrypted When making use of the Enterprise subscription model, Rasa helps to implement required protocols	Encryption based on input/output when hosted by Boost.ai.
Do you perform penetration tests or vulnerability assessments?	<i>See above</i>	Yes. We do penetration and vulnerability tests on a regular basis. Results can be shared after discussion.	Responsibility of the client.	
Do you have any monitoring mechanism to check for suspicious activity?	<i>See above</i>	We're currently not monitoring for atypical usage. Cognigy offers a Log-Page that visualizes in-product logs showing e.g. errors that occurred during conversation execution. Furthermore, Cognigy offers additional docker-stacks that can be deployed next to the actual product. These stacks can be used for system-monitoring, latency measurement and log-aggregation. The stacks Cognigy currently offers are: <ul style="list-style-type: none"> • ELK: A ready-made docker-stack that follows the best practices and can be deployed side-by-side with Cognigy. Offers a central place to visualize all system-logs produced by all containers within the cluster. • monitoring-service: Additional Cognigy service that collects various metrics, such as "lost messages" and "service response-times" • Prometheus / Grafana: A ready-made docker-stack that follows the best practices and can be deployed side-by-side with Cognigy. Visualizes important performance metrics, 	<i>See above</i>	The platform follows AWS protocols.

Question	SAP	Cognigy	Rasa	Boost.AI
		such as CPU, memory, disk-space aggregated for all machines within the cluster Additional dashboards: Some software components within the ordinary Cognigy stack already have dashboards built-in. One of these is e.g. displaying metrics about http response-times for the reverse-proxy that is part of the Cognigy installation.		
What protocols do you follow in the case of a security breach or when personal data is compromised?	<i>See above</i>	We are following the protocol that is defined in our incident response policy.	<i>See above</i>	
How do you ensure compliance with GDPR?	The right to be forgotten can be implemented since logs/messages can be purged through the API. Example: When you close the conversation, the user can be asked whether they want to have their data removed.	Cognigy.AI ships with tooling that allows for a fully GDPR-compliant implementation of Conversational AI. This includes full-transparency in terms of data storage and even logging on a container level (in case of an on-premise environment). Explicit user data related features, like the "Contact Profile", have GDPR opt-in switches that can be used to track user consent. Data capture communication varies per channel (e.g. Facebook Messenger vs Amazon Alexa), and should be customized on an implementation level. https://docs.cognigy.com/ai/resources/manage/contact-profiles	Rasa has supported clients earlier with the implementation of GDPR compliant chatbots. It can assist clients with taking the necessary actions. Removing the conversation when requested by the user is possible but conversation will first be logged and then deleted.	The client chooses the specific rules to comply with GDPR e.g. specifying the exact duration that data can be stored.
AI Cognitive Engine – Accuracy				
Do you offer NLU-based AI or merely a decision tree encoded for all cases?	NLU-based AI model that can make its own conclusions regarding the intent.	Cognigy.AI is shipped with a built-in NLU with industry leading accuracy for intent recognition. https://docs.cognigy.com/ai/nlu/nlu-overview/overview	NLU-based AI (using predefined NLU chunks)	NLU-based
Is the model pre-trained to understand general conversations? Can transfer learning be applied to our specific case?		With Cognigy.AI, your first basic virtual agent is deployed within a few clicks after logging in. This is achieved by offering in-tool templates and user journeys that show new users how to get started. More complex conversational flows and back-end interactions can be added and built from the ground up with deployment to channels within hours of creating agents. https://docs.cognigy.com/ai/resources/agents/journeys	Contextualised learning is possible	The models are pretrained on language data sets.
In which format do you pass information to the chatbot to train it to specific use cases?	CSV file to load in majority of information possible. Afterwards adding new intents can be done manually.	Cognigy.AI's built in NLU applies machine learning to example sentences and uses rule based intents trained within the platform. Custom dictionaries are trained via the lexicon module to extract specific information slots from user inputs. The intent trainer enables monitoring of	Intents are added using YAML files or manually through the GUI (only for Enterprise subscription)	Input using JSON files should be possible.

Question	SAP	Cognigy	Rasa	Boost.AI
		misunderstood phrases to continuously improve the NLU's understanding. https://docs.cognigy.com/ai/nlu/nlu-overview/ml-intents https://docs.cognigy.com/ai/resources/build/lexicons https://docs.cognigy.com/ai/resources/tweak/intent-trainer		
Is it possible to know the accuracy of the model? What about the accuracy of speech recognition?	Insights on the precision, recall, confusion matrix etc. can be gathered.	Yes. With the Cognigy Insights dashboard it is possible to measure how accurate the NLU recognizes user inputs. User input in speech is translated into text with the STT generator and analysed in the very same way as text inputs. https://docs.cognigy.com/insights/cognigy-insights https://docs.cognigy.com/insights/dashboard-nlu-performance	Yes, the analytics platform can give information about model accuracy.	
Do you have any mechanism in place which allows the chatbot to continuously improve its accuracy? If yes, is it an automated process or does it require human intervention?	The level of accuracy required can be customised per intent. Not very relevant for unique cases. Intent improvement requires human input, no auto-training present. The chatbot will never adapt the dataset itself. Every modification is made manually by the developer. However, hints of new intents are given. This is a deliberate choice: auto learning can have unintended consequences (drift, bias). Manual input remains the safest option.	Yes. The Cognigy.AI Intent Trainer allows platform users to review misunderstood inputs to improve the accuracy of the Machine Learning Intents. It is easy to add an input text from a user as an example sentence to a selected ML Intent, or add it to the Reject Intent of your Flow. https://docs.cognigy.com/ai/resources/tweak/intent-trainer	Intent improvement is done by enlarging the data sets. This requires manual input. Alternatively, the model can be altered by comparing pipeline performance and by tuning hyperparameters.	Human intervention by adding intents but aided by intent suggestion
AI Cognitive Engine – Flexibility				
Can the engine deal with spelling and grammar mistakes/errors?	The NLU can deal with grammatical and spelling mistakes to find the correct intent	Yes. The NLU allows for a certain fuzziness and each input is scored against available example sentences. User input is analysed on multiple levels: E.g. on single words, on complete sentences and on a combination of example sentences. https://docs.cognigy.com/ai/nlu/nlu-overview/overview	Yes	Built-in spelling checker
Is there a limited number of questions/actions that can be included?	In the thousands.	No, there is no limit.	No	No

Question	SAP	Cognigy	Rasa	Boost.AI
<i>AI Cognitive Engine – Opacity and code ownership</i>				
Can the client gain insights on the models, or do they operate as a black box? Are you using neural networks, vectorizers, random forests ...?	Basis of the chatbot is built on confidence scores. The confidence requirements can be customised.	Cognigy operates its own NLU, and our NLU team will be happy to have a conversation to give limited insights. Please note however that connectors to Google Dialogflow, Luis and Watson exist to persist using other NLUs with Cognigy, further extending its technology agnosticism.	Rasa Enterprise is a closed-source product. The NLU models are all open-source.	
Who holds the ownership of the developed solution? Is continuity foreseen after ending the license or in case of insolvency of the chatbot vendor?	Solution itself is a SaaS solution. The solution benefits from the internal development by SAP. Ownership by SAP, no possibility to replicate to own environment. SAP will assess the ownership of the logs. If clients stop the engagement with SAP, they can retrieve their data set (intents) and log files.	The use of Cognigy is based on subscription licenses over a contractually determined period of time, with bespoke support and maintenance SLAs. Data is owned by the customer during and beyond subscription duration. Access to an Escrow Account for continued use during a subscription period in case of insolvency can be contractually determined.	Code and data is fully owned by the client.	
<i>AI Cognitive Engine – Language</i>				
How many languages does your chatbot support? Are these only EU languages or in general worldwide languages?	All languages can be used since the language depends solely on the dataset. However, some key features of the bot (golden entities, sentiment analysis) are only available in <u>some languages</u> . There is no translation engine built-in but it is possible to use APIs of external translation services.	Cognigy.AI's on-board NLU is pre-trained with curated data from over 100 languages to support intent recognition and key phrase detection. Any other natural (or artificial) language is supported based on language-agnostic NLU algorithms. For 20 of the most common languages Cognigy.AI provides prebuilt entities that allow automatic processing of inputs like dates, currencies and others specific to a language that is defined in the flow. https://docs.cognigy.com/ai/nlu/language-support	The Rasa assistant can be used on training data in any language. If there are no word embeddings for the desired language, clients can train featurisers from scratch with provided data. Additionally, spaCy offers many pre-trained language models which can be integrated.	All of the dominant European languages except Portuguese and Polish are supported. Additional languages could be trained on request by the client.
In the case of voice support, do you have native speakers for	Depends on the speech channel used.	See above	N/A	Depends on the external service used.

Question	SAP	Cognigy	Rasa	Boost.AI
each supported language? How many languages?				
Does the user need to select one language at the beginning of the conversation or can the language change based on the user's request (language detection)?	Language detection is in place.	No. Depending on the flow design, any user in any supported language can use the bot. https://docs.cognigy.com/ai/flow-nodes/logic/switch-locale/	Language detection can be applied.	Language detection. Different languages can be used in one conversation.
Do you require an external service to perform the translations or is this an internal feature of your chatbot?	External service	Cognigy.AI uses external translation services from for example DeepL, Google translate or Microsoft translate. https://docs.cognigy.com/ai/flow-nodes/other-nodes/set-translation	Internal language models are present. Additionally, external translation can be implemented. However, Rasa is raised about the accuracy of this approach. They are convinced higher accuracy can be reached by training a model per language (nuances, dialects, etc.).	External service for translation when not using language data sets.
AI Cognitive Engine – Customisation				
Is your solution a fixed product or is there room for customisation?	During the meeting, several customisation options were discussed: <ul style="list-style-type: none"> Engine: confidence threshold per intent Conversation flow editor Services: integration of APIs Lay-out: CSS web client (see below) 	<p>Cognigy.AI is an all-inclusive low-code platform package, yet delivering a highly flexible and customizable enterprise-ready solution. This can be done either by the client, or can do it with support of Cognigy.</p> <p>As an example of native customization features provided, please see under Access rights. Access rights can be defined for entities such as projects, flows, lexicons, playbooks down to fine grained flow properties such as Basic or Advanced flow nodes.</p> <p>For each item, the privilege to execute a create, read, update or delete (CRUD) actions can be set in a role based manner.</p> <p>A role a set of privileges to a user assigned to it. Roles assigned to a user define the access control list (ACL) displayed under Access Rights which is updated immediately. https://docs.cognigy.com/ai/tools/user-menu/access-control/</p>	Customisation is possible, even to the extent of the selected algorithms etc.	Customisation possible through integrations.

Question	SAP	Cognigy	Rasa	Boost.AI
Can clients customise themselves or can this only be done by your developers?	Customisation by the client. This is also true for the integration of external APIs. SAP can provide support if needed.	Yes. Cognigy.AI is an all-inclusive platform package delivering a highly flexible and customizable enterprise ready solution. This can be done either by the client, or can do it with support of Cognigy.	Customisation can be done by the client, with support of Rasa (depending on the support plan)	By the client.
In what way is the front end customisable? Is it possible to customise any colours or logos used?	SAP offers a web client. Through the use of CSS code, visuals of the chatbot can be customised.	The frontend of the Cognigy chat can be 100% customized into the corporate branding with logo's, colours, font types etc. https://docs.cognigy.com/ai/endpoints/webchat/webchat	The web widget can be fully customised via code. The result is shown in a GUI.	Standard styling of widget but custom colours and text size possible
Hosting environment				
Where can your chatbot product be hosted? By you and/or on the client premises? If hosted by you, which platform is used? If hosted by client, are there any specifications?	Currently hosted by SAP on the Cloud Foundry platform, attached to the AWS data centre in Frankfurt. Additional cloud providers will be supported in the future (Azure, Alibaba Cloud ...). It is not possible to host the solution on an on-premise platform. There is no link between the SAP Conversational AI platform and SAP Cloud Platform	Cognigy offers SaaS, dedicating hosting and on-premises deployment options with support for Kubernetes as a container orchestrator. This microservice based environment allows the Cognigy.AI platform to be automatically or manually scaled to match the resource requirement of the customer. https://www.cognigy.com/about	Solution is hosted by the client. Either in the cloud or on-premises. Ability to run Kubernetes is required. The hardware specifications can be found on: https://rasa.com/docs/rasa-x/installation-and-setup/install/helm-chart	Boost.ai hosts on AWS servers but hosting by client is possible. When hosted by the client, this has the downside that they don't have direct access so no monitoring available. Dedicated list of requirements regarding the hardware. This can be shared when a NDA is in place.
Is it possible to have the chatbot hosted by you, while the database remains on the client premises?	No, the database is always owned by SAP, on AWS (or others to come in the future)	See above	No (always hosted by client)	Yes, but recommended to have the data on the AWS servers to have access to all of the tools.
How do you foresee flexibility to integrate your solutions with client architectures?	APIs	There are no major hurdles we foresee with integration with 3rd party solutions, as long as they provide an API access. Please note that would encourage the involvement of our Professional Services department to enable and assist either an implementation partner, or the end-customer directly in the integration execution. https://docs.cognigy.com/ai/developer-guides/using-api/	By default, the solution is hosted on client's premises	Setup is supported but client is responsible for the maintenance.

Question	SAP	Cognigy	Rasa	Boost.AI
After deployment, who maintains the solution (from a technical perspective)?	SAP is responsible for maintaining the software behind the solution but only performs an advisory role for the development/improvement of the chatbot.	The solution needs to be maintained and administered by either the customer or the customer's services partner (as managed services). Cognigy itself offers only hosting services, but will not have access to the solution itself.	Rasa is responsible for maintaining the platform behind the Enterprise subscription but the client is responsible for the maintenance of the actual solution. However, support is provided (depending on Support Plan).	Depends on the hosting. When on-premises, the client handles the servers but Boost.ai is responsible for software updates to the platform.
What recovery mechanisms do you have in place in the case your chatbot goes down if you are hosting the solution? Can those mechanisms also be applied on-premises?	AWS cloud recovery mechanisms	Chatbots are a virtual construct that relies on the availability of multiple microservices. Each microservice has 3 or more replicas. Kubernetes ensures that always desired amount of replicas exists. If one replica dies, Kubernetes creates a new one. As Kubernetes is the only supported runtime, this can be replicated on-premises as well.	Responsibility of the client but cooperation between client and Rasa possible to build a recovery plan that covers different types of failures, to bring the environment in line with the existing enterprise policies.	Natural disaster protocols to survive data centre failures by spinning up containers in different data centres, while keeping the data contained in Europe.
In the case the solution is hosted on-premises, is the chatbot availability compromised or can we also implement some recovery mechanisms in case of failure?	N/A	This highly depends on the on-premises infrastructure setup. Cognigy achieves high-availability in cloud setups by using constructs like availability zones and regions.	<i>See above</i>	
Pricing				
Which pricing plans do you offer?	Pay-as-you-go and subscription model (charge per pack of X conversations). No fixed monthly fee applicable.	Subscription based licensing, contract durations from 1 to 3 to 5 years.	Yearly subscription based. Usually starts with a 3-year plan.	Standard and Enterprise pricing. The Standard plan has a lower monthly fee but higher ticker charge and limited languages available.
Which are the primary factors that determine the cost? Do you have a fixed monthly fee, or does it depend on the usage? (Number of actions/questions,	Pricing per conversation, independent of length of conversation. A conversation is between a user and the	Cognigy does not charge per bot (virtual agent), user, flow, channel, language etc, all of which can be considered "unlimited". All of the functionalities of Cognigy.AI are included in the subscriptions, with main criterion being the number of "conversations" which will determine pricing.	Pricing based on: <ul style="list-style-type: none"> Support level Number of nodes you deploy (infrastructure multiplier) 	Fixed annual or monthly fee + ticker charge per conversation when using Boost.ai's cloud platform.

Question	SAP	Cognigy	Rasa	Boost.AI
Number of languages, Chatbot hosting, Features included, Support plan)	chatbot. A conversation ends after inactivity of 15 min. Pricing information is <u>publicly available</u> but discounts possible depending on volume and type of contract. Price range of 230 EUR - 300 EUR per pack of 1000 conversations. This can be lower if the demand is steady and can be predicted/reserved upfront.	<p>A conversation means a Session with a User on a Channel on a day with a maximum of 50 User In-puts per Conversation.</p> <p>Customers commit to a contingent of conversations per year to be automated, this contingent being freely usable during the agreed billing period (i.e. 1 year), overages being charged only once the contingent is exhausted.</p> <p>Cognigy's Voice Gateway is priced separately and may be dependent on other variables.</p> <p>A detailed insight into pricing structure and costs can be given under NDA.</p>	Specific requests can be handled.	
<i>Closing remarks</i>				
Do you see other relevant points/functionalities that distinguish you from the other competitors? In other words, why would we decide to go for your solution instead of other chatbot vendors?	<ul style="list-style-type: none"> Monitoring capabilities. Advanced platform regarding NLP which is at the core of the chatbot. Integration with other SAP services. SAP often lists at the top of independent reviews (e.g. Gartner) 	<p>Cognigy is a named leader in the IDC marketplace 2021. It is without a doubt Cognigy stands out amongst its competitors.</p> <ul style="list-style-type: none"> Easy to use graphical conversation flow editor allows conversations to be built without code, no technical skills required Built-in NLU with industry leading accuracy for intent recognition Open source extensions repository with drag and drop installation (pre-built connectors for Salesforce, ServiceNow, Microsoft Office365 and many more) SaaS, Dedicated hosting or On-Premises deployment options Seamless live agent handover for contact centre integration Rapid project implementation time frames Machine-learning agnostic (built-in connectors for Watson, Dialogflow, LUIS and open NLP pipelines) Trusted by more than 500 brands worldwide. 		