„Hi, how may I assist you?" Chatbots for Better Service Delivery

Practical Guide, Applications and Principles in Development Cooperation

Published by:

giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Imprint

As a federally owned enterprise, GIZ supports the German Government in achieving its objectives in the field of international cooperation for sustainable development.

Published by:
Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Registered offices
Bonn and Eschborn, Germany

Address
Friedrich-Ebert-Allee 32+36, 53113 Bonn, Germany
Dag-Hammarskjöld-Weg 1–5, 65760 Eschborn, Germany

E info@giz.de
I www.giz.de/en

Projects and authors
This guide was written by the following authors and their respective GIZ projects (in alphabetical order):

- Daniel Brumund – "FAIR Forward - Artificial Intelligence for All": FAIR Forward strives for a more open, inclusive, and sustainable approach to AI at an international level and with partners in Ghana, India, Indonesia, Kenya, Rwanda, South Africa and Uganda. Since November 2025, Advisor on Digital Governance and AI in the GIZ Competence Center "Digital Societies" and for the initiative "Towards a Just(er) Digital Governance".

- Prashant Pratap Singh – "Data Service Center": The Data Service Center advises GIZ projects on collecting, analyzing, managing, and utilizing diverse datasets. Their work significantly contributes to embedding data-driven practices within GIZ and building data products. They provide expertise in three areas: Geo-spatial data, Artificial Intelligence (AI), and Monitoring and Evaluation (M&E), applicable across various sectors.

- Lisa Werkmeister – "FAIR Forward - Artificial Intelligence for All": FAIR Forward strives for a more open, inclusive, and sustainable approach to AI at an international level. Collaborating with seven partner countries (Ghana, India, Indonesia, Kenya, Rwanda, South Africa, Uganda), the initiative promotes access to training data and AI technologies for local innovation; fostering local technical know-how on AI; and developing policy frameworks for ethical AI, data protection, and privacy.

Thank you to the following colleagues for their valuable feedback (in alphabetical order): Johannes Jacob, Erik Lehmann, Isaac Manzi, Christian Merz, Jan Nehring, Jonas Nothnagel, Anna-Lisa Wirth.

Design
Studio GOOD, Berlin

Illustration
Tara Monheim

Bonn/Eschborn, 2026

Table of Contents

Table of Contents

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CMS	Content Management System
FMB	Fach- und Methodenbereich (GIZ's Sectoral Department)
IVR	Interactive Voice Response
LLM	Large Language Model
M&E	Monitoring & Evaluation
MT	Machine Translation
NLP	Natural Language Processing
NLU	Natural Language Understanding
QAP	Question and Answer Pair (for FAQ chatbots)
RAG	Retrieval-augmented generation
SLM	Small Language Model
STT	Speech-to-text ('speech recognition')
TTS	Text-to-speech ('speech generation' or 'speech synthesis')

1 Introduction

AI Chatbots have emerged as powerful tools in the realm of user engagement in different sectors, offering innovative ways to engage with diverse audiences, provide crucial information, and deliver services efficiently. By leveraging conversational interfaces especially in local languages, organizations can bridge gaps in communication, extend their reach, and respond to user needs in real time.

However, implementing chatbots in development contexts comes with unique challenges, ranging from ensuring inclusivity and accessibility in local languages to addressing ethical concerns, sustainability and suitability depending on the context, sector or system it is planned to be used for. Use cases vary from applications in health, agriculture or public administration to efficiently support health workers, farmers and extension agents or public servants and governments. Moreover, every country is unique and brings its own challenges due to different regulatory requirements, infrastructure and financial means to sustain these AI solutions.

1.1 Goals of this guide

This guide consolidates insights from past AI chatbot projects developed by and with partners of the GIZ initiative "FAIR Forward – Artificial Intelligence for All" (2019-2026) and the FMB Data Service Center. It aims to serve as a learning document for practitioners in GIZ, as well as in other development agencies, philanthropies and donor agencies that wish to implement AI chatbots. The guide provides lessons-learnt for building AI chatbots and explores:

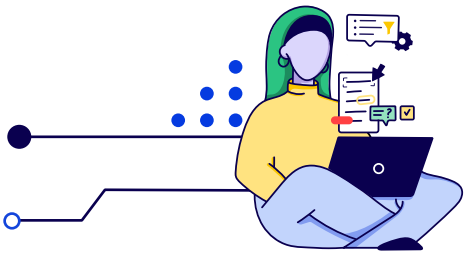
- motivations and requirements for developing a chatbots
- functional types of AI chatbots suitable for different projects
- content and knowledge base behind chatbots
- technological architectures for AI chatbots
- steering and management of AI chatbot projects
- responsible AI principles

Throughout the guide, you will also find practical examples of chatbot projects implemented by GIZ. These are meant to offer hands-on inspiration for how chatbots have been integrated in projects in India, Kenya, Rwanda and Uganda.

1.2 Who is this guide for?

This guide is aimed at anyone who is planning, advising, steering or tendering activities that foresee incorporating the use of AI chatbots in projects. Its content was written based on insights from GIZ staff who have been involved in the design and implementation of AI chatbot projects both based at GIZ headquarters and in various of its country offices.

While this may make some of the more specific insights more directly relevant to GIZ staff, the authors have written the guide in a way that it should hopefully also be of interest, and valuable for, people working in other development, donor and funding institutions active within projects in an international cooperation context.



1.3 What is an AI chatbot?

This guidance focuses specifically on AI chatbots. But what is an AI chatbot? And how is it different from non-AI chatbots? It's time for some working definitions that this report uses.

One of the most prominent **non-AI chatbot types** are those based on simple decision-trees, e.g. via USSD interaction. They are accessed through a short-code on your phone (e.g. *123#) and guide you through one or more follow-on menus with options that can be selected by sending a certain number. Such 'static' chatbots are still very common to access service of mobile network providers, or for simple helplines. They emerged with the release of the first feature phones. Since then, chatbots have come a long way and include voice assistants (like Siri or Alexa), telephone helplines that understand what you say with speech recognition or, more recently, the emergence of generative AI and with it conversational chatbots that allow users to engage through an almost human-like dialogue.

An **AI chatbot**, on the other hand, allows users to engage with it through conversation – via text or voice – and uses AI to interpret the user inputs. This happens with the help of AI models in the chatbot's technological stack, be it for natural language understanding (to understand the intent behind a user inquiry), voice recognition (to transcribe what a user says into text for further analysis), speech synthesis (to create an audio read-out of a text) or for text generation (to create responses to user questions using generative AI) or possible combinations of those.

The type of AI that can be used is ever expanding – but below is a brief overview of popular AI-based technologies used in chatbots:

- **Natural Language Understanding (NLU):** This is used to understand the intent behind a user inquiry. For example, if a user says or writes "I'd like to check my application status", the NLU model will extract "application status" as the intent. NLU can also detect entities, such as dates, locations and other information, in the user inquiry. The NLU can be a narrow in terms of just dealing with task like 'entity recognition' and hence might be powered using not so large models or it could be very comprehensive like acting as 'agent' using LLMs.

- **Speech-to-text (STT) or Speech recognition:** This is used to transcribe a spoken word or sentence into text that can be further analysed. STT models can make your chatbot work in a wide variety of languages – pending the availability of models in your required language.

- **Text-to-speech (TTS) or Speech synthesis:** This is used to turn text into audio that can be played to users. For instance, an explanation or a response of your chatbot can be 'read out' to users via a synthetic voice. This can make your chatbot more accessible to users – and more natural to utilise. Similar to STT models, TTS models can create audio in a variety of languages but depend on the availability of such models in your required language.

- **Generative AI, especially LLMs:** Generative AI and Large Language Models (LLMs) in particular are used to create a conversational interaction between the chatbot and its users (somewhat similar to tools like ChatGPT). This can make the information you want to provide more easily accessible to users as they can 'ask' for it in a conversation. However, it comes with certain risks and limitations such as the generation of factually incorrect responses. To improve the correctness of outputs, LLM-based systems often use so-called retrieval augmented generation (RAG) which bases the generated response on a pre-configured database (see Chapter 6).

  With the rise of reasoning LLMs there is option to include the Chain-of-Thoughts (reasoning) within the answer. This allows the user to verify if the steps taken by LLMs are sufficient to arrive at the answer and hence make judgement on the quality of output.

1.4 Benefits and challenges of AI chatbots

The use of chatbots can bring various advantages to organisations, projects and services that involve the provision of information to people. These include:

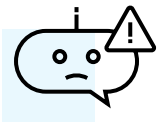
- **Simplified access to relevant, timely and accurate information:** Chatbots can provide information on a specific problem/question in real-time and tailored to local, national or regional contexts, and they are theoretically available 24/7.

- **Option of omni-channel use:** Based on a users' preference and capacities (such as typing/reading vs. listening/speaking), users can access chatbots through the channel of their choice including messengers, web interfaces, voice-enabled apps or phone hotlines (including through interactive voice response / IVR).

- **Interaction & Feedback:** Chatbots are characterized by user interaction in the form of two-way communication. It means that users do not only read static information but can engage with the system. In this way, chatbots can mirror human conversations, whilst allowing users to provide feedback which developers can use to improve the chatbot to gain more context knowledge and, thereby, provide more relevant answers.

- **Inclusivity regardless of digital skills:** The interaction with a digital tool through natural language (e.g. via speech recognition) is regarded as the easiest way to use digital tools, compared to smartphone apps, websites and other user interfaces. This makes chatbots especially interesting when digital tools should be provided to users with little or no digital skills.

- **Low-cost accessibility:** Chatbots can provide relevant information to a large user base in real-time for a relatively small cost compared to traditional methods such as call centers. For instance, chatbots can be made accessible via phone hotlines which means they can be used by simple "feature phones", not just laptops or smartphones. These telephone bots rely on the more widespread availability of telephone connections instead of internet connectivity, making them the device with the lowest entry barriers for low-income communities.



While chatbots can be very helpful in certain settings, they are no silver bullet for information access. In fact, chatbots bring their own challenges that need to be considered. These include:

- **Good content needs work:** A chatbot is only as good as its content base. Preparing and maintaining the content and information in a user-centric way requires significant work that must not be underestimated. This guide will give you advice on how to do it.

- **Missing or wrong answers:** There may be questions that a chatbot has no answer because either the question or related content are not captured in the content base. Moreover, especially LLM-based chatbots (Generative AI chatbots) come with the technical challenge of erroneous outputs – also called 'hallucinations'.

- **No human replacement:** Even the best chatbot will not be able to replace human, in-person interaction or advice. Therefore, chatbots projects should – at best – be considered as complementary to existing information services (e.g. to provide wider reach, or easier access to information).

- **Maintenance efforts:** The deployment and running of a chatbot requires maintenance both for its technical infrastructure (e.g. the servers it runs on) and for its content (e.g. including feedback mechanisms for users, and procedures for reviewing and updating content). Don't forget to plan budget for maintenance during and beyond your project.

- **Data Privacy Issues:** Users might share sensitive information with the chatbot. Depending on the type of chatbot, this information might be processed by external services, e.g., LLMs by big US tech companies. Also, conversations and sensitive information might be processed and stored on web servers, which makes an audit with regards to data protection laws necessary.

- **Data availability:** Many modern chatbots, especially those that are LLM-based and use RAG (see Section 6.4.1), require information to be customised for specific domains or database. If this data is publicly available, e.g. published on a homepage, this is no problem. But any data that is classified may require time-consuming discussions over access and quality assurance, especially when dealing with the publication of large textual datasets (e.g. 1000+ documents). Further even though the data might be accessible, making it readily usable for LLM-based chatbot would still need significant data processing steps.

- **LLM-inherent bias:** All LLMs are trained on vast amounts of data that have been taken from different sources. Such vast data amounts used during model training are seldom curated and fact-checked, and mirror existing biases and stereotypes present in the data. The mitigation of such biases should be carefully considered and tested when using an LLM for the chatbot. Open source models with detailed data documentation are preferable for exactly this purpose.

AI SYSTEMS ARE NOT HUMAN

The term 'hallucination' is widely used to refer to erroneous outputs of LLMs. The term is criticised for being anthropomorphic which means that it ascribes human qualities to non-human entities like LLMs. This can be problematic because it tends to exaggerate AI systems' capabilities and distort our own judgement of them (see: "Anthropomorphism in AI: hype and fallacy | AI and Ethics"). To avoid this, you can use 'predictive error' as an alternative term that is technically accurate.

1.5 Using open-source software and AI

This guide advocates for using open-source tools to develop chatbots – both with open-source software (e.g. development frameworks) and open-source AI (e.g. data and models).

Why open-source software?
Simply put, open-source software offers developers the freedoms to use, study, modify and share it for any purpose. These 'four freedoms' allow you to use open-source software for your chatbot development. Beyond that, they ensure higher quality, security and transparency of the tools you use because anyone can inspect and improve them. This matters not just for your chatbot project, but much of our digital world is built on open-source tools, e.g. Android smartphones, internet servers, Java, WordPress, just to name a few.

DEFINING OPEN-SOURCE AI

The Open Source Initiative (OSI) has published v1.0 of the Open-Source AI in late 2024 which was drafted through a community process. It requires for the source-code used to train and run the AI model and the model parameters to be published under an open-source license. Moreover, sufficiently detailed information about the training data must be provided. The training data itself can, but doesn't have to, be provided.

Why open-source AI?
Similar to open-source software, the use of open-source AI – such as an open-source AI model – in your project gives you the same four freedoms mentioned above for building a chatbot. In this way, open-source AI helps increase trust and reduce risk through transparency about how the AI model was built. This is a crucial advantage over proprietary or closed-source AI models without details on the training data which brings significant risks regarding data protection, trust and biases. Open-source AI models can be hosted on premise or by local firms with the needed computer infrastructure which fosters digital sovereignty and independence from foreign firms.

The use of open-source AI models – e.g. built and provided by local firms – also contributes to local innovation by encouraging the use of AI models as digital public goods. These offerings can be made profitable through innovative business models and contribute back to the open-source efforts, for instance by publishing additional training data, or finetuned models, equally under an open-source license. By doing so you are building a collaborative community.

A NOTE OF CAUTION

Not everything that is called open-source AI actually meets its definition. Several big tech companies call their AI models open-source when often they are very opaque about the datasets used for building the models. Having access to either the dataset, or to a comprehensive documentation of the dataset, is crucial to assess risks associated with the model, for instance due to biases in the collected data.

2 Setting the stage: Motivation and requirements for AI chatbots

AI chatbots can make citizen-facing services more accessible and easier to use – if they are meant to enhance already well-established services; if they are designed and tested with service stakeholders and users; and if they have a well-funded maintenance structure. This means that AI chatbots are no silver bullets. Before resources and time is invested, it is important to reflect and be clear about why an AI chatbot is a suitable tool in a given context.

To support this reflection, this chapter provides guidance to better understand:

- motivation and success factors for chatbot projects
- envisaged users and involved stakeholders
- readiness for embarking on a new chatbot project

2.1 What motivates a successful AI chatbot project?

If your project or partner organization wants to develop an AI chatbot, it's important to reflect on the motivation for this. Each context is different and unique but there are certain factors that have shown to contribute to successful chatbot implementations. The main two questions to consider are:

1) How does the AI chatbot enhance or enable the process?
2) How would this service function without the AI chatbot?

The success of the AI Chatbot highly depends on the use case and context. Below, we summarized what context has enhanced the feasibility and, hence, the impactful usage of AI chatbots based on the experiences from past GIZ projects.

Engaging users comes with a significant benefit for your use case: For an AI chatbot to be effective, it must address a clear and specific need where user interaction adds value. For example, farmers in different locations require different agricultural advice based on their location and what they grow. A chatbot can help retrieve the farmer's context, enrich the query by local information such as weather forecasts and provide more relevant information than a static website could do.

Chatbot channels respond to users' (digital) literacy: The chatbot has to be accessible through channels that align with your foreseen users' (digital) literacy. For instance, accessing a website will be difficult for users who can't read or type. In such cases, different or mixed channels such as hotlines (via voice) and/or chat interfaces (via typing) may be suitable. Voice can even enable users who don't own a smartphone. Moreover, chatbots provide the option of retrieving context despite spelling mistakes or paraphrasing whereas search functions of ordinary websites don't allow for that which can lead to user frustration and drop offs. Furthermore, multi-lingual chatbots can adapt the language specifically to user thereby making them personalized assistant.

Information is of high relevance for a large target group: Classic methods of providing information are websites or person-to-person communication e.g. hotlines or consultations. When the information is of high relevance for a large target group, these methods might not be sufficient to reach everyone. In these cases, multiple information sources can be very beneficial. A chatbot can help to reduce the numbers of queries and the need to talk to a human being. One potential use case could be that there already is an established service that is to be enhanced with an AI chatbot. For instance, a digital directory of government service is in place, but citizens struggle to find the services they need in the directory. An AI chatbot can simplify the discovery of services/information in the directory by responding to needs that users describe in a chat. This may reduce calls or visits a public office receives by citizens.

Limited scope: Chatbots work better the more limited their application area is. A chatbot that should talk about "all topics that a farmer might be interested in" is less likely to work than a chatbot about "how to counter pests and diseases in farms".

2.2 When do AI chatbots not make sense?

As developing and maintaining chatbots is resource-intensive, there are also cases where AI chatbots are not the best solution. Some examples are listed below.

Limited user base: An AI chatbot will struggle to gain traction if it's not reaching a substantial number of users. In essence, a chatbot with a limited user base remains

underutilized, similar to how a service that doesn't reach its audience fails to fulfil its purpose.

Effort > usefulness: If the effort required to maintain and use the chatbot outweighs its practical benefits, it becomes impractical. For example, if deploying and managing the chatbot demands significant resources but doesn't sufficiently enhance user experience or operational efficiency, its adoption becomes questionable.

Other solutions offer better outcomes: In scenarios where existing or new alternatives provide more effective solutions, adopting an AI chatbot might not justify the investment. Sometimes, a static website will already do the trick. It's important to highlight that automation of certain steps in an existing solution should not be confused with use of AI and doesn't warrant in itself as a good reason for incorporation of AI-chatbots.

Ignorance of outreach importance: Making users aware of the chatbot is crucial, requiring campaigns and similar efforts. If these promotional strategies can't be effectively executed, the adoption and success of chatbots may be limited.

Neglecting the real world: An AI chatbot will not automatically fix a service that is not working in the real-world. In short: A digitised faulty service is still a faulty service. Consider, for instance, a road authority that struggles with timely repairs of potholes. In this case, an AI chatbot that makes it easier for citizens to report potholes will not by itself lead to better roads without also addressing challenges within the road authority.

Lack of AI models: Without performant AI models, an AI chatbot will not work – or be faulty and rejected by citizens. If respective models for the languages you are targeting do not exist, you can explore if they can be developed but this can be very resource- and time-intensive.

Information is not existing or unstructured: If a chatbot is designed for a specific topic and relies on specific document sources, then the document/knowledge base must have structured access and consistency. This means that it should at least be accessible in a structured and consistent way e.g. via APIs or database management systems. If not, a chatbot might face the challenge of accessing the information or might fetch incomplete information which will be of little help and, at worst, lead to 'information chaos'.

100% accuracy required: Both chatbots and humans will never be 100% accurate. Often, trained humans will be more accurate than chatbots. In some applications (e.g., a doctor chatbot giving medical advice) you need to compare the quality of the chatbot with the quality of a human to determine if a chatbot is applicable.

2.3 Checklist

CHATBOT READINESS

The below checklist is meant to give you an initial overview of whether minimum criteria for a successful chatbot project are in place. If any criteria are not yet met, you may use this as an indication on issues that should be revolved prior on embarking on a (new) chatbot project. Please note that the list is not exhaustive but based on general experience with such projects.

- ☐ **Clarity on 'Project Lifecycle':** Do you have a clear overview of all stakeholders that will be engaged and consulted throughout each stage of the chatbot lifecycle (e.g. design, development, user-testing and feedback, deployment, maintenance)?
- ☐ **Clarity on Ownership:** Do you already know who will be the 'product owner' of chat chatbot, and who will maintain the chatbot (incl. costs for technical infrastructure and maintenance, content maintenance)? AI is a very fast-paced technology and to ensure consistent user engagement it's crucial that the maintenance of chatbots keeps up with the changing landscape of AI and users' expectations.
- ☐ **Clarity on Access:** Have you defined how the chatbot will be accessed by users? Depending on your requirements and user needs, this may include the use of public phone numbers, hosting on a website, a smartphone app and/or integration with popular communication channels such as WhatsApp or Telegram.
- ☐ **Clarity on Costs:** Is the foreseen product owner of the chatbot financially equipped to develop and sustain the chatbot? Does a realistic cost overview exist? Is there a sustainability plan for when funders or development partners withdraw?
- ☐ **Clarity on Content:** If the chatbot relies on some document/database then it's imperative to check if the database is structured and accessible and regularly updated. Where and if the data is ready to be plugged into chatbot? Who does regular maintenance of the database?
- ☐ **Clarity on Technical Feasibility:** Is the use-case generally technically feasible? Do necessary technologies, e.g., does the necessary TTS tool exist in my target language and is its quality good enough?

3 Choosing an AI chatbot: How to find one that works for you

Once you have identified that a chatbot is suitable for your project or activity, the next important step will be to decide on the type of chatbot to develop and use.

This chapter will guide you in choosing the most suitable chatbot for your case based on the envisioned functionality or purpose of the chatbot. In particular, the chapter will:

- present three functional chatbot types commonly used in a wide range of use cases
- outline each functional chatbot type in terms of its input and output modalities, content, technological core as well as its advantages and disadvantages
- link to examples showing how these chatbot types have been used in GIZ projects

This list of functional chatbot types is no means exhaustive. It is based on practical experiences at the time of writing. Yet, it is created with a view towards remaining applicable for some time despite the fast pace of technological development, particularly in AI.

Impactful chatbots don't have to be fancy: The Mbaza Covid19 chatbot reached millions of Rwandans on simple phones via USSD

A NOTE TO 'TECHIES':

More technically versed readers may wonder why this chapter offers a typology of chatbots that is largely based on their function and not, for instance, based on their technological core or user interface. This chapter aims to help a non-technical layperson understand more generally what kind of AI chatbot might be suitable in their project. For this purpose, a more functional typology of chatbots seemed more accessible. In chapter 6, you find more detailed technical information about user interfaces and technological cores of chatbots which cut across the functional types.

3.1 Types of AI chatbots

	FAQ CHATBOT	GENERATIVE AI CHATBOT	HYBRID CHATBOT
WHAT IT DOES	It responds to questions with pre-defined answers that are stored in a knowledge base.	It responds to questions with newly generated text that can be limited to sector with knowledge base. It creates a conversational feeling.	It combines simple non-AI chatbots (e.g. USSD/ FAQ) with generative AI chatbots, or even a link to live agent support.
INPUT	Freely phrased questions from users, usually as text but voice is also possible (with STT and TTS model)		Users start with simple decision-tree (e.g. via USSD). If more info is required, it continues as Question-Answering Agent or generative AI chatbot
OUTPUT	• Hyperlinks to articles in a knowledge base • Pre-defined answers for intents, as text or voice (pre-recorded or TTS generated)	Newly generated text as a contextually related response to the question. It sounds quite natural due to large amount of data used to train underlying LLM	The answers get more complex depending on the users' information need (e.g. from FAQ to generative AI chatbot, or even link to live agent)
CONTENT	• Knowledge base with articles (e.g. an intranet) • Pre-defined answers for intents	To minimise risk of flawed output: Knowledge base with articles and other information that the LLM uses to generate an answer	• Decision-tree text for USSD (or similar) non-AI chatbot • For generative AI component see box on the left
TECH (CORE)	• Intent classification model in the chatbot language • If voice-based: STT and/or TTS model in chatbot language • LLMs (not Generative model) can be used – but not strictly necessary	• LLM that works in target language (or with machine translation) • 'RAG architecture' to connect LLM with knowledge base • If voice-based: STT and/or TTS model in chatbot language	• For USSD: short-code and API connection via mobile phone operator • For generative AI see boxes on the left

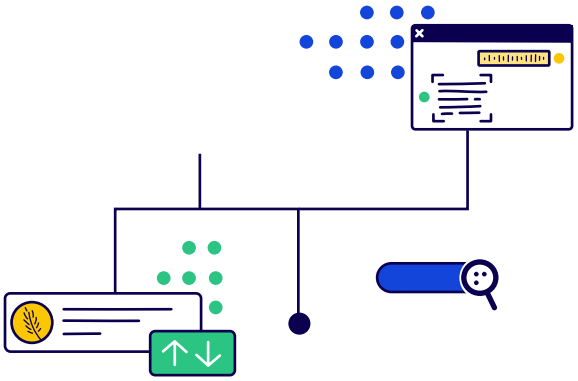
	FAQ CHATBOT	GENERATIVE AI CHATBOT	HYBRID CHATBOT
PRO'S	• Low-cost, simple to build • Effective to make existing knowledge accessible	• More conversational interaction with users • Responses can be contextually limited to knowledge base	• Easily accessible even via feature phones (if USSD, SMS are used as entry point) • Continuously more complex responses tailored to user needs
CON'S	• Limited to pre-defined intents (= library search) • Intent classification can misunderstand nuances	• LLMs can be costly (esp. proprietary ones) • Longer development time • Risk of flawed outputs ('hallucinations' / predictive errors) • Many LLMs are opaque about their training data • Prominent LLMs do not work well in languages spoken in Global South	• More complex tech infrastructure needed that integrates non-AI chatbot elements with AI chatbot • Potentially higher maintenance cost

3.1.1 FAQ chatbot

An FAQ chatbot, also called 'knowledge base chatbot', is designed to respond to user questions with pre-defined answers from an existing knowledge base. The response will always be the same – no new text is being generated, and the user inputs are limited.

Input: Users can freely enter any question and submit it to the FAQ chatbot. It uses natural language understanding (NLU) to analyse the question and to identify the user's intent, that is the piece of information the user is looking for. Thanks to NLU, the chatbot can understand questions from users regardless of how they are phrased. Most commonly, questions are text-based, but voice input is possible if a speech-to-text model in the chatbot languages is used.

Output: The FAQ chatbot provides the user with information related to the identified intent. This can, for instance, be a list of articles from a knowledge base related to the intent. Or the response can be in the form of pre-defined answers for the intent which, most commonly, are text-based, but voice responses are also possible pre-recorded audio clips or generated via a text-to-speech model.



Content: For an FAQ chatbot, you will need to have at minimum a table with intents, as in topics and queries that users might have, and the related answers. The quality of the chatbot depends on the comprehensiveness of such a table – this means: the more intents and answers you pre-define, the better the chatbot will be able to respond. Ideally, this content should be developed together with topic experts. Beyond this you can also have an existing repository (or knowledge base) of articles that the chatbot can link to relating to the intent.

Technology: An FAQ chatbot requires an intent classification model in the language of your chatbot. This is an AI model (not generative LLM's) that uses NLU to analyse a question and classify the user's intent. The intent is then used to source a pre-recorded answer or point to a related article (see 'Content' above) – either of which are provided in the chatbot's underlying knowledge database (see Chapter 5). LLMs can be used for FAQ chatbots (and nowadays often times are used). However, this is not strictly necessary and can increase development and maintenance costs. If the chatbot is voice-based, it requires a speech-to-text ('speech recognition') model in the chatbot's language which transcribes the user's voice input into text for further intent analysis. This STT model needs to be fine-tuned for the topical domain of your chatbot. If you like for responses to be read out by the chatbot (and not be pre-recorded), you will also need a text-to-speech ('speech synthesis') model in the chatbot language.

Advantages: FAQ chatbots can be developed fairly simply and at a low cost for simple dialogues. They are optimal for clearly defined tasks and developers have complete control over the responses that are provided. Especially in cases where a knowledge base already exists, they can be an effective way to make its more easily accessible.

3.1.2 Generative AI chatbot

A Generative AI chatbot is similar to FAQ chatbots in that it also responds to user questions with an answer based on the intent behind the question. What makes it different is the fact that it does not rely on pre-defined responses: It generates new, natural sounding text in response. This can make interactions with generative AI chatbots feel more naturally like conversations.

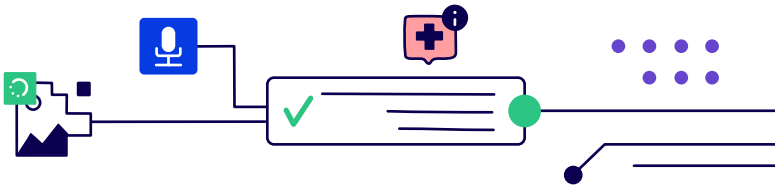
Input: Users can engage with a generative AI chatbot as if they have a conversation. This can be text-based (as is the case with ChatGPT or similar tools), or voice-based if a speech-to-text model is used as part of the chatbot architecture. The chatbot uses a complex architecture based on large-language models (LLMs) to respond to the question posed.

Output: The generative AI chatbot responds to the user with a newly generated text. To do so, it uses LLMs that are able to generate text as a contextually related response to the question. Their quite natural sounding text generation is possible because of the large amount of data that LLMs are trained on.

Disadvantages: FAQ chatbots can only respond to intents with pre-defined answers. This can make them feel more like an advanced library search than natural interaction. They are only as good as the AI model for intent classification – and can miss the nuance in a user's question. It might require resource to tailor the AI model for the specific use-case and even then due to their limited interactive capability the user interaction can be upsetting due to their pre-defined answering capability.

EXAMPLE

Kenya:
Chatbot for the Office of the Data Protection Commissioner (see page 48)



Content: The simplest way to use a generative AI chatbot would be 'as is' and without any content preparation. However, this is often not suitable because it easily leads to flawed outputs (some refer to this as 'hallucinations[1]'). One way of minimising this is to prepare a knowledge base of articles and other content relating to the sectoral domain of the chatbot. This database is then connected to the LLM to provide an answer based on the information in it. When a user asks a question, the chatbot will use the LLM to generate a text response specifically based on the content from the connected knowledge database. This is called a "RAG[2] architecture" which can also generate "hallucinations" but minimises the risk thereof.

[1] See chapter 1.4 on benefits and challenges of AI chatbots for a contextualisation of the term 'hallucination'.
[2] RAG is an acronym for 'retrieval-augmented generation' and describes the process of generating content with an LLM based on a connected knowledge database, and a so-called 'vector database' that is used to find content related to the user question. For an explanation: What Is RAG Architecture? A New Approach to LLMs | Cohere

Technology: A generative AI chatbot requires an LLM to be able to generate responses. To operate in the foreseen language of the chatbot, it can use an LLM that works well in the respective language or combine an LLM with a machine translation model into the target language. As described above under 'content', a RAG architecture is – at the time of writing – a promising way for building generative AI chatbots with factually correct outputs. For this, you will also need a knowledge database with articles (and other content) related to your specific sectoral domain (see chapter 6 for more details). For voice-based interactions, you will also need either an STT model (for recognising voice inputs by users) or a TTS model (for creating audio versions of the chatbot's text outputs).

Advantages: Generative AI chatbots can create the feeling of a more natural, conversational interaction with users. Their responses can be contextually based on existing knowledge bases and provide succinct responses based on them (see 'RAG architecture' above).

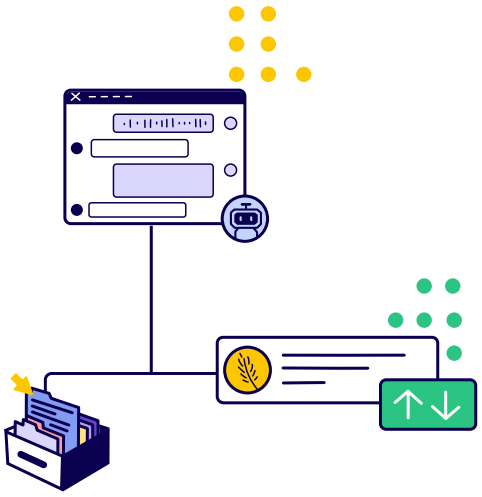
Disadvantages: Generative AI chatbots are based on LLMs and their usage can be costly, especially when using proprietary models. The development time can be longer, e.g. due to the need to fine-tune LLMs to specific domains/language and iterative user-testing. There is also a risk of flawed responses depending on the used model or knowledge base which means there is a need for continuous monitoring of outputs, or – at minimum – a feedback mechanism for users. Many proprietary LLMs (such as those by Meta, Open AI or DeepSeek) do not provide information about the data they are trained on. This lack of transparency is problematic, e.g. due to lawsuits for breaking IP law, or because they foreclose the possibility to check for toxic content. Lastly, most popular LLMs are not yet working well in many languages spoken in the Global South – though there are promising regional efforts to address this (e.g. by Masakhane across Africa).

EXAMPLE

GIZ's KIM chatbot
Kalimagezi – The Audit Chatbot (see page 45)
Conversational AI Chatbot for Discovery of eServices (Kenya) (see page 44)

3.1.3 Hybrid chatbot

A hybrid chatbot is, most commonly, a combination of non-AI chatbots (e.g. based on simple decision-trees, see chapter 1.3) with a generative AI chatbot or even a link to a live chat or hotline with a person. Such a hybrid chatbot can respond to a user question with a list of pre-defined answers, a list of related articles or a generated text response. One of the most popular means to implement these (when writing this guide) would be using so-called agentic approach. If the user has follow-up questions, the chatbot can provide these in a similar manner, or via generative AI. In more complex, or sensitive, settings, the chatbot can offer users to connect them with a live agent via chat or phone.



Input: An entry point to a hybrid chatbot can be a simple decision-tree to guide users to the most popular options. If the user requires more information, the chatbot could then offer the user to pose a concrete question and react on this either as an FAQ chatbot or generative AI chatbot, depending on which type you decide to use in the hybrid chatbot (see chapters 3.2 and 3.3 for more details on these types).

Output: The output of a hybrid chatbot can be increasingly more complex pending the information need of the user. It can start with a simple decision-tree response and continue with either pre-defined information (via an FAQ chatbot) or newly generated responses (via a generative AI chatbot). If this remains insufficient, the chatbot can offer users to connect them to a live agent (if your chatbot architecture and back-end allows for this).

Content: The content of a hybrid chatbot usually consists of a defined decision-tree of information that can be accessed e.g. via USSD or IVR (interactive voice response), and a knowledge base created for an FAQ chatbot or generative AI chatbot, depending which one of these forms part of the hybrid chatbot (for details of the latter, see chapters 3.2 and 3.3).

Technology: For the initial non-AI chatbot interaction, e.g. USSD or IVR, you will need get a phone short-code and API connection with a mobile phone operator in the country where your chatbot will be implemented. A local developer experienced in developing USSD- or IVR-based interactions will be able to help you with developing the required informational back-end for the USSD or IVR decision-tree. For the follow-on connection with either an FAQ chatbot or generative AI chatbot, please see chapters 3.2 and 3.3 for technology details. For a link to a live agent, you will need a call-centre integration (or something similar).

Advantages: A hybrid chatbot can make information more widely accessible, especially for users who only have simple feature phones (for use with USSD or IVR). It is also able to provide continuously more complex responses and support, all the way to a live agent.

Disadvantages: A hybrid chatbot requires a more complex technological infrastructure that integrates e.g. USSD or IVR short-code and API with an FAQ or generative AI chatbot, as well as a phone linkages to live agent support. This can make it more costly for maintenance, depending on the costs of short codes, SMS, or phone integration if live agents are included.

EXAMPLE

Agricultural Information Exchange Platform
(see page 43)

3.2 Other chatbots

There are several other types of chatbots that are not further described here because, at the time of writing this guide, they did not seem relevant for usage in projects related to sustainable development. For instance, virtual assistants are known to anyone who has used Apple's Siri, Amazon's Alexa or Google Assistant. While these assistants can be useful to individuals who own the respective devices, they are less suitable for GIZ project because (a) they are proprietary products, and (b) recreating virtual assistants is prohibitively expensive.

4 Project management: Steering a successful chatbot initiative

An important aspect regarding the success of the chatbot is proper project steering encompassing the entire project lifecycle, from conceptualization to development, launch, and ongoing maintenance. This chapter outlines:

- crucial aspects for clarity on stakeholder and beneficiaries
- ensuring ownership, maintenance and community-engagement
- procurement requirements

PLEASE NOTE

Many of the insights shared in this chapter are based on GIZ-internal experiences but may be equally relevant for other, similar institutions planning to use chatbots in projects.

4.1 Stakeholders and beneficiaries

Stakeholder management is a critical component of successfully steering an AI chatbot project, particularly for development cooperation with an emphasis on ethical practices and sustainability. Engaging stakeholders from the outset is essential, and this engagement must include understanding and addressing the needs of beneficiaries early in the process.

Ideally, representatives of the beneficiaries should be part of the design process, directly involved in shaping the chatbot's development. This involvement can take the form of human-centred design workshops, user testing, and co-creation sessions, ensuring that the final product aligns with the real needs of those it is intended to serve. For example, the Agricultural Information Exchange Platform developed by GIZ's FAIR Forward initiative made field visits and end-user testing a critical component of the grant management with the developers. Similarly in another use-case of an Audit Chatbot, GIZ's Data Service Center developed the prototype and tested it with users in Uganda to get an insights on real user demands and refine the user stories before the production ready solution was developed.

The goal is to create a solution that benefits users and minimizes any negative impacts, particularly by engaging beneficiaries in iterative feedback loops to refine the chatbot's design and functionality.

Before the project begins, it's essential to ask key questions that will guide the entire project lifecycle.

1) Who are the **primary users** and how will they benefit? Do they believe the AI Chatbot will **benefit** them? What needs and hesitations do they have?
2) Who else is involved and has **interest, investment and stake** in the project? Are **negative effect**s considered? Will a certain group be left out or left without a job after the launch of the AI Chatbot?
3) How can these interest groups be organized, and their priorities documented? How can stakeholders and beneficiaries be engaged in every stage of the project?

To effectively manage these various groups, it's essential to understand their interests and document their priorities. This includes developing clear stakeholder maps and communication strategies to ensure that each group's voice is heard. Stakeholders and beneficiaries should be engaged in every stage of the project, from the initial conceptualization to the final rollout and beyond. Regular check-ins, feedback sessions, and user-testing phases are critical to maintaining an ongoing dialogue and ensuring that the chatbot evolves in line with user needs. The GIZ Data Service Center can be consulted on this topic to support with methods such as requirement analysis workshops to define stakeholder

Expert stakeholder engagement for the development of the eCitizen Chatbot in Kenya

maps and user stories as starting points for successful long-term project steering.

For context, common stakeholders and beneficiaries from GIZ AI chatbot projects include:

STAKEHOLDERS

Owners of the chatbot: This could vary depending on the use case but often includes governments or public offices that oversee the project.

Developers of the solution: These are the technical teams responsible for building the chatbot, ensuring that it functions as intended and meets user requirements.

Information providers: These can be research institutions, public institutions, or other entities that supply the data or content the chatbot will use.

BENEFICIARIES

Citizens: The general public who will interact with the chatbot for various services, such as information access or assistance.

Professionals: Experts from various fields, such as healthcare, agriculture, or education, who may use the chatbot for support in their work.

Businesses and Entrepreneurs: These groups might use the chatbot for customer service, information retrieval, or automation of certain business processes.

In summary, stakeholder management is not a one-time effort but a continuous process that ensures the AI chatbot aligns with the needs and values of all involved parties. By proactively identifying, engaging, and organizing stakeholders and beneficiaries, an ethical AI chatbot project can be more responsive, inclusive, and ultimately successful.

4.2 Ownership and maintenance

Ensuring ownership and maintenance of the chatbot is essential for its long-term success and sustainability. To guarantee that the chatbot continuously serves its intended purpose:

Clear ownership structures must be established from the outset. This involves identifying which stakeholders are responsible for maintaining and updating the chatbot, such as the development team, the owning organization, or a dedicated oversight body.

Accountability measures need to be in place. This means that chatbot maintenance extends beyond the technical upkeep and includes ensuring that ethical considerations, data privacy, and user needs as well as any grievances are continuously monitored and addressed as the chatbot evolves.

Budgetary and staff resources for chatbot maintenance (incl. its on-going operations) need to be set-up and planned as early as possible in the project. This means that the commitment to ownership must be aligned with the financial and operational capacity to ensure long-term viability, avoiding scenarios where the chatbot becomes obsolete or unsupported after initial success, or after funding from GIZ or others ends.

While dealing with these questions it is important to think along the lines of the technological framework being used: how easy it is to maintain; what are the incurring costs? And also, what are open-source options that can be used (if not used already)? Have you considered capacity building with partners so that maintenance can be taken over by partner institutions?

4.3 Community and user engagement

User engagement is critical to establishing and maintaining the chatbot's relevance and effectiveness. To ensure that users actively use the chatbot, it is important to make the user experience intuitive, accessible, and beneficial. This involves:

- **User-centric needs assessment:** For a chatbot that is valuable to its users, you should consider doing a thorough needs assessment with potential users as a first step of the design process. Such an assessment could cover issues such as challenges, expectations and wishes that users have for the chatbot that you are planning.

- **Regular updates:** To ensure the chatbot remains relevant for users, you should put in place a regular updating process based on user feedback which can be collected through surveys, focus groups, and usage analytics. This helps ensure the chatbot offers clear value to users (whether through improved service access, increased efficiency, or enhanced user experience) which, in turn encourages consistent usage.

- **Trust measures:** Fostering trust with users is essential; this can be achieved by ensuring transparency in how their data is used, addressing any privacy concerns, and providing clear communication about the chatbot's capabilities and limitations.

- **Community collaboration:** Some of the great tools/software's had been built by collaborative community contributions. It is good to consider how you can foster contributions from local developer community post roll-out phase, which not only fosters local innovation but also allows the partners to rely on local resources during the maintenance phase.

In summary, through a combination of strong ownership, regular updates, and user-centric as well as collaborative design, the chatbot can remain a valuable tool that continues to serve its beneficiaries over time.

Close collaboration with users and stakeholders throughout the development of the chatbot is key for its success

4.4 Selection and expert profiles

The effective selection and assessment of proposals and expert teams are key to ensuring that the AI chatbot project is impactful, inclusive, and technically sound. Based on experience from previous chatbot projects, this sub-chapter provides guidance on identifying the right partners and outlines practical steps for selecting proposals and tips for expert profiles.

Example Selection Process: Insights from the Agricultural Information Exchange Platform
Establishing clear, transparent, and fair assessment criteria is essential to the success of any AI chatbot initiative, especially within GIZ where resources are allocated with the ambition of maximum impact. Whether selecting implementation partners through tenders or open innovation challenges, a structured evaluation framework ensures objectivity, inclusivity, and alignment with overarching development goals.

To illustrate best practices in proposal selection, the following is an example of a selection process based on experiences from the AI-based Agriculture Information Exchange Platform (AIEP) (see page 43). It outlines best practices in proposal selection, the role of expert reviewers, and how tender assessments can be conducted to identify the most suitable partners for chatbot development. AIEP was a larger-scale project with multiple implementing cohorts. Consequently, this is a comprehensive process which may be replicable for similar projects. For small-scale projects, or single vendor tenders, a slimmer process may be more suitable and cost-efficient.

Selecting Proposals from an Open Call / Innovation Challenge
In contexts where multiple organizations are invited through an open call or innovation challenge to submit proposals, such as for the AIEP project, a standardized and well-coordinated evaluation process is essential. The aim is to reduce subjectivity as much as possible and score each proposal in an objective and transparent matter.

To ensure neutrality and diversity in the evaluation, every proposal is assessed by multiple reviewers with different technical and contextual (e.g. agriculture) knowledge, ensuring gender balance and regional expertise. Reviewers should remain independent and blinded to each other's scores until aggregation. Lastly, averages should be used based on all reviewers' scores to ensure balanced decision-making. For that purpose, each proposal receives a score (0–10) per criterion, with weightings applied to calculate a final composite score.

	TECHNICAL INNOVATION	END USER	OPEN SOURCE & DPC	OPERATIONS	SUSTAINABLITY	TEAM &LOCAL PARTNERS	BUDGET	SCORE
WEIGHT	0,25	0,25	0,1	0,1	0,05	0,15	0,1	
PROPOSAL 1	8	5	0	7	3	8	7	6
PROPOSAL 2	5	6	7	4	7	5	4	5,35

Table 1: Example of a scoring sheet to assess proposals

PLEASE NOTE

The above scoring example is 'tender-agnostic', meaning that it is kept generic for easier adaptation into various tendering procedures. In the case of GIZ tenders, for instance, the above-mentioned criteria can be integrated into the formal technical assessment criteria for tenders and serve as a guide for how to weight the respective criteria.

The proposal selection process for AIEP had the following five steps and concluded in the selection of the final implementing cohorts.

1) **Call for Proposals in two steps:** The Call for Proposals was organised in two steps with a first call for concept notes that led to 100+ submissions which were assessed against an initial list of criteria (similar to those mentioned in Step 4 below). Out of these submissions, 30 cohorts were invited to two ideation workshops aimed at refining the respective concepts and establishing final cohorts. In a second step, 27 revised full-scale proposals were submitted for the final selection of the participating cohorts.

2) **Initial Proposal Screening:** All submissions are reviewed for completeness and basic eligibility.

3) **Scoring by Independent Reviewers:** A Technical Advisory Panel, as well as internal stakeholders (e.g., donors and project implementers), evaluate proposals using a structured scoring system. A predesigned scoring guide can help increase objectivity among assessors.

4) **Weighted Evaluation Criteria:** Based on the priorities of the chatbot project, criteria and weighting might differ (see Table 1 above):
    - Technological Innovation & Feasibility (25%)
    - End-User Orientation & Inclusivity (25%)
    - Open-Source & Digital Public Goods (10%)
    - Project Management & Timeline (10%)
    - Sustainability (5%)
    - Team & Local Partners (15%)
    - Budget (10%)

4) **Interactive Feedback:** Optional Q&A sessions with applicants allow evaluators to clarify uncertainties and assess teams "live," providing an added layer of insight before final decisions.

Expert profiles

The bidders are required to provide staff credentials to prove if they are competent to carry out a tender. While the usual requirement of expertise can vary based on nature and scope of chatbot to be developed, there are certain criteria which can be added to each profile as well as a range of qualifications that are likely to be needed for a wide range of chatbot projects.

The below exemplary expert profiles and qualifications are based on GIZ tender documents from various chatbot projects with generalised qualifications per key assessment criteria (see Table 2 below). Both the proposed positions and respective qualifications are not meant to be exhaustive but rather as a potential starting point to build on and revise according to a new chatbot project's specific context and requirements. The following positions are proposed in addition to the team leader:

Selecting Proposals from an Open Call / Innovation Challenge

In contexts where multiple organizations are invited through an open call or innovation challenge to submit proposals, such as for the AIEP project, a standardized and well-coordinated evaluation process is essential. 'The aim is to reduce subjectivity as much as possible and score each proposal in an objective and transparent matter.

- **Product manager:** Leads the chatbot conceptualisation (incl. functional / non-functional requirements); ensures chatbot meets user needs; manages the backlog.

- **Data Scientist and NLP Engineer:** Implement NLP-relevant chatbot elements in all required languages, incl. identification, development and/or adaptation of suitable NLP model(s); oversee dataset curation and preparation, testing and monitoring of the model; train partners institutions on management of NLP-relevant chatbot elements.

- **Tech and DevOps Lead:** Leads technical coordination, implementation and overall chatbot architecture; Responsible for coordinating ML/NLP engineer and developers.

- **Software Development and Data Engineers (expert pool):** Implement software development, plus deployment and integration with partner systems (if applicable); create data pipelines, data analytics, dashboards and other data usage visualisations.

	PRODUCT MANAGER	USER EXPERIENCE EXPERT	DATA SCIENTIST/ NLP ENGINEER	DEVOPS / MLOPS LEAD	SOFTWARE DEVELOPMENT/ DATA ENGINEERS (POOL)
EDUCATION	University degree in computer science and related fields (incl. NLP / ML), or degree in design and product management or other relevant fields	University degree in computer science, computer engineering, media and communication design or related fields	Master's degree in machine learning, AI, NLP, data science, computer engineering, computer science, or related field	University degree in computer science, computer engineering, or related relevant field	
LANGUAGE (PROFICIENCY)	Good business language skills in English and [chatbot language(s)]	Good business language skills in English and Full language proficiency in [chatbot language(s)]		Good business language skills in English	B2-level language proficiency in English
GENERAL PROFESSIONAL EXPERIENCE	5 years of experience in communication and information technology	3 years of professional experience in design for communication and information technology	5 years of professional experience in machine learning, data science	7 years of professional experience in software engineering or related fields	5 years of professional experience in software engineering or related fields
SPECIFIC CHATBOT EXPERIENCE	3 years of experience in product management for AI- or data-based products and experiences with bots (incl. tech stacks built on Rasa and LLMs) 2 years of experience in digital innovation	1 year of experience in user experience design, preference is given experience in conversational design 1 year in UX design within public policy processes	3 years of experience in machine learning, artificial intelligence or natural language processing (NLP), including with LLMs 2 year of experience with machine learning, artificial intelligence or NLP low-resource languages	5 years of experience as a full stack software developer 4 years of experience in software integration and DevOps 2 years of experience in developing AI-based solutions, especially with machine learning operations (MLOps) tasks	3 years of experience as a full stack software developer 3 years of experience in software integration and DevOps 1 year of experience in developing AI-based solutions, especially with MLOps relevant tasks
LEADERSHIP EXPERIENCE	2 years of experience in managing teams	N/A	N/A	3 year of experience in managing IT projects	N/A
REGIONAL EXPERIENCE	2 years of experience in [region]	2 years of experience in [region]	2 years of experience in [region]	N/A	N/A

Table 2: Suggested expert profiles and qualifications

4.5 Impact Assessment and evaluations

When developing and deploying an AI chatbot, it is crucial to have a robust monitoring and evaluation (M&E) strategy in place from the start. A well-designed M&E framework helps ensure that the chatbot meets its intended goals, delivers value to users, and remains aligned with the overall project objectives. This strategy should outline clear performance indicators, define what success looks like, and establish a plan for ongoing monitoring throughout the project lifecycle. Key components of an effective M&E strategy for AI chatbots include:

1) **Clear Goals and KPIs:** Define the specific goals of the chatbot (e.g., improving access to information, increasing user engagement) and set measurable key performance indicators (KPIs) such as user satisfaction, response accuracy, and task completion rates.

2) **Ongoing Monitoring:** Track chatbot performance continuously through analytics tools and user feedback mechanisms to identify any issues early and ensure it is consistently performing well.

3) **Adjustments and Iterations:** Build in flexibility to adapt the chatbot as the project progresses. Use real-time data and feedback to make updates and improvements, ensuring the chatbot remains relevant and effective.

3) **Stakeholder Involvement:** Engage key stakeholders, including project managers, developers, and users, in the monitoring process. Regular check-ins with these groups ensure the chatbot meets their needs and aligns with project goals.

Such an M&E framework as illustrated below was created as part of FAIR Forward's Agricultural Information-Exchange Platform (AIEP) initiative where 5 advisory platforms for farmers were developed during an open innovation approach. The M&E framework was jointly developed with the consortia and monitored monthly to compare, track progress and update on challenges.

AIEP Standard Indicators

FARMER FACES A PROBLEM → FARMER ASKES AIEP CHATBOT → FARMER RECIEVES CORRECT AND RELEVANT RESPONSE → FARMER CHANGES BEHAVIOR → YIELS IMPROVES

FARMER CHANGES BEHAVIOR → 600DB STUDY*

YIELS IMPROVES → FUTURE EVALUATION (E.G. RCT)

	USER INDICATORS (ALL DISAGGREGATED BY GENDER)					TECH INDICATORS		
INDICATOR	**# of active weekly users**	**Retention rate (active users/total users)**	**% of successful advisory ****	**User satisfiction score**	**Most commonly asked questions**	**RAGS metrics**	**# agricultural content sources**	**Cost per session**
HOW TO MEASURE?	Active users = at least on question asked	See active users	Share of responses not answered by 'I do not know'	# of weekly thumbs up/# of active weekly users ***	Aggregated by common topic	# of weekly thumbs up/# of active weekly users ***	Confirmed by list of agricultural content sources	Including channel, API, and infrastructure costs by country
MONITORS	Farmer asks AIEP Chatbot	Farmer asks AIEP Chatbot	Correct and relevant responce	Correct and relevant responce	Correct and relevant responce	Correct and relevant responce	Correct and relevant responce	Long term sustainablity

* Details to be confirmed in survey design Stage. Digital Green in addition currently experiments with adopting their practice adoption meadurements for the AIEP chatbots.
** In addition, to be assessed against Golden Q&A (tbc) or through assessment of generated answers by agricultural experts (tbc)
*** Option for IVR services to be defined

Figure 1: M&E Framework of the Agricultural Information Exchange Platform

Once the M&E strategy is in place, evaluating the performance and impact of AI chatbots is essential to ensure they are effective, meet user needs, and contribute to the overall success of the project. Regular assessments help identify strengths, challenges, and areas for improvement. Evaluations can be done through three primary methods:

1) **Chatbot Analytics for Data-Driven Insights (quantitative analysis):** Chatbots come with built-in analytics tools to track key performance indicators such as user engagement, response accuracy, and drop-off rates. This allows for statistical evaluations that provide concrete data on how well the chatbot is meeting user needs. Regularly reviewing these metrics helps identify patterns, areas of success, and where improvements are necessary.

2) **User Studies (qualitative analysis):** Users try out the chatbot according to pre-defined scenarios. The central success measure is task-success: Which percentage of users completed their task succesfully? Study participants should ideally be end-users, but often participants are interns, students, etc. You can test different conversational styles (short and precise, small-talk, short factoid questions ("how much does it cost?") and complex questions ("how does it work?"). Please note that the outcomes of this test are usually qualitative. You need a large number of participants to gain quantitative results that allow for example to compare two chatbots to each other.

3) **External Monitoring and Evaluation (M&E) Experts:** Neutral evaluations from external M&E contractors provide an unbiased perspective on the chatbot's performance. These evaluations can capture insights such as convenience, timeliness, relevance of the information provided by the chatbot and provide statistics how different users (gender, age, region) feel about the service provided. Ideally, chatbots can also be compared to other similar services on the market.

Below are a few best practices that should be incorporated as part of the M&E framework:

1) **Continuous Feedback Loops:** Incorporate user feedback regularly through surveys or automated responses to understand user satisfaction and identify any gaps in the chatbot's functionality. This can be done through rating functions such as star ratings or thump up function.

2) **Track Both Quantitative and Qualitative Data:** Combine hard data (e.g., usage rates) with qualitative insights (e.g., user experiences) to get a full picture of the chatbot's performance and impact.

3) **Focus on Ethical Considerations:** Ensure privacy, inclusivity, and bias are considered in every evaluation. This is crucial for maintaining trust and effectiveness, especially in sensitive development contexts. More details on this can be found in chapter 7.

Focus group interviews can be useful for qualitative analysis of your chatbot's performance

5 Gathering content: Managing the knowledge base

A chatbot is only as good as its content, referring to the knowledge base from which the chatbot draws information that is passed on to its users. The old ICT motto "Garbage in, Garbage out" also applies here. Even the best user interface or the most sophisticated technical backbone will not be sufficient: If its underlying content is flawed, the chatbot will be unusable and might even cause problems or, at worst harm, to people using it. This means that you need to take content management very seriously, and plan sufficient resources and time to collate, validate, test and update the content used in the chatbot (as illustrated in Figure 2).

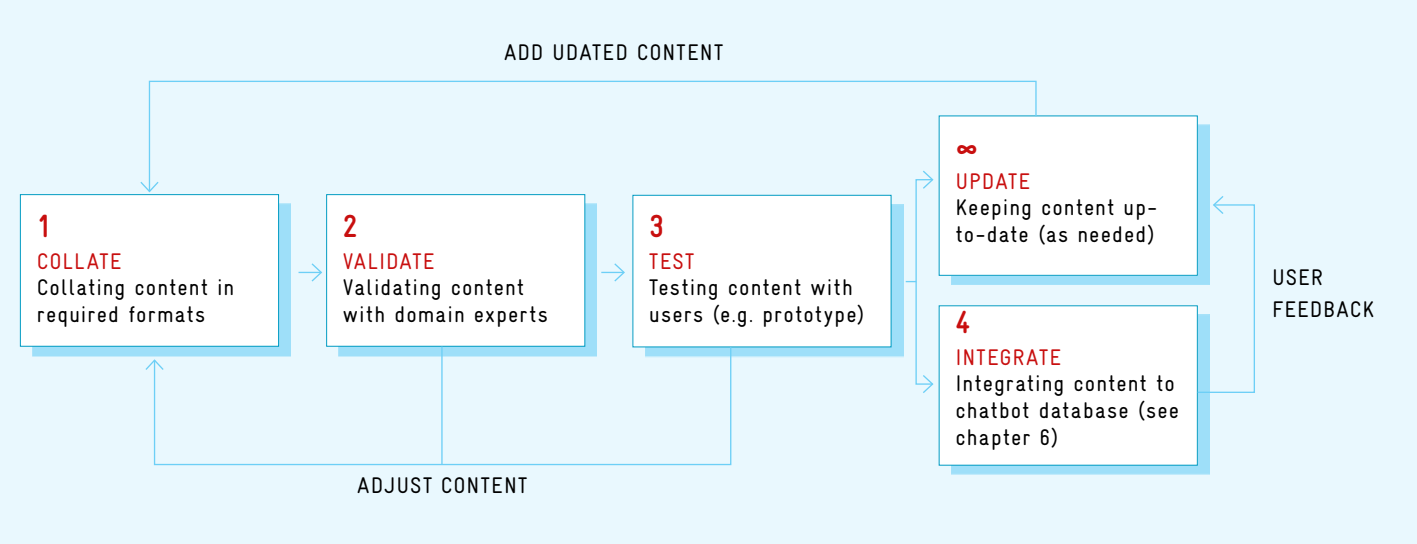


Figure 2: General process for managing chatbot content

This chapter offers guidance for each step of managing content for your chatbot. It explores:

- assessing available data for the chatbot
- collating content in required formats (incl. in locally relatable language)
- validating and testing your content with domain experts and users
- updating your content as needed (e.g. through a continuous process)

A NOTE TO 'TECHIES' ON INTEGRATION:

Before your content can be used in the chatbot, it also needs to be technically prepared and turned into a content database. This content integration into the chatbot is described in Chapter 6 "Technology".

5.1 Collating content

In a first step, you will need to collate information about the domain, sector or service for which the chatbot is meant to be used. This content will form the base from which the chatbot will source the information that will be communicated to users. The collation of content will require you to identify available content sources, plan suitable collation methods and consider formats that are necessary for using the content in your chatbot type of choice (see chapter 3).

5.1.1 Data sources and availability

The sources of your content will vary depending on the sector or domain that your chatbot is being built for, as well as the language(s) for which it is built. The availability of this data is often of crucial importance and should be one of the first question discussed when assessing the feasibility of the project.

Digital vs non-digital data sources: Digital data formats are preferable because they can be more easily

integrated into the chatbot's content base. Scanning of printed sources is very time-consuming – though it may be worthwhile for important manuals not yet digitised

The below list mentions typical content sources. It is ordered in ascending difficulty of collecting the content for the chatbot. This order is based on experience but may vary depending on your sector/domain. It is not meant as a ranking of which type of sources you should aim for – some of the most relevant content may be hiding on intranets or printed manuals. However, it could offer inspiration to make content that is more difficult to access (e.g. printed documents) better available via more easily accessible sources (e.g. public websites or, eventually, via a chatbot)..

EASY TO COLLECT

- **Publicly available data (e.g. websites):** The data is already public, e.g. published on a website such as Ministry websites, think tanks, eGovernment service websites. This is ideal because, since it has been released already and they can be easily accessed without restrictions and the content sourced directly, e.g. via automated crawlers. There are often few objections against sharing it and using it for the chatbot. However, you need to check for any licensing constraints for website content.

- **Small (digital) data sources:** Small or limited data sources include e.g. single manuals and or reports. They may need clearance from data protection departments (depending on their content) but are easy to be used once cleared:

    - **Digital documents / Word files:** e.g. editable reports, manuals, articles. These can also be easily turned into chatbot content though their accessibility depends on where they are stored (e.g. individuals vs. public websites).

    - **Digital documents / PDFs:** e.g. published reports, manuals, articles. These may be easy to access as files (e.g. on public websites) but content access is more difficult because it will require extraction from PDFs which required manual efforts and/or automated scripts for content extraction including quality assurance mechanisms.

    - **Domain experts / Translators:** If specific content is needed that is not available yet, you may need to invest in domain experts who can write the content you need. This may also be necessary for translating content into different languages. These are more difficult to access as they will require time and financial resources, but they may be helpful for once-off investments to build and/or quality-check your content.

- **Large data sources:** Large sources of data include organisational intranets or data that can be accessed via APIs (e.g. for sector-specific data such as weather data or geopraphical data).

- **API sources:** Certain sources of data offer access via APIs – either openly or closed (against payment or registration). These include e.g. weather data, geographical data, or data from online platforms. For some chatbot projects this may be needed e.g. for an agricultural chatbot with timely weather information.

    - **Intranet (non-public) websites:** "We have all the necessary information on the ministry's SharePoint" is an indicator of a lengthy and tedious process in which 100s or 1000s of documents need to be manually released. Content that sits on intranets can only be used if the owner (e.g. a Ministry) agrees to open it up and make it available for the chatbot. Often there are once-off access agreements as automated collation from intranets are difficult to institutionalise.

- **Printed documents:** Documents that are only available as hard copy – e.g. printed reports, manuals, books etc. – are the most resource-intensive to access because (a) they are only accessible in a certain location and (b) they will need to be digitised to be usable for the chatbot. This may be needed for content that only exists in printed form – but you may want to consider making it accessible on digital platforms.

HARD TO COLLECT

Important issues to watch out for when collating content:

- **Copyright:** For any content that you collect for chatbot usage, you need to ensure that you have consent from the respective copyright holders. For instance, for content from a ministerial website, you will need to ask for consent from your ministry partner. Some online content may be licensed with permissive licenses (e.g. Creative Commons) which may require you to attribute the sources.

- **Personal information:** Any content that includes personally identifiable information (PII) will need to be revised so that any PII is removed. This includes any information that relate to a specific person – such as their address, birthdate or other information.

- **Harmful language:** Needless to say, no content that contains harmful language should be included in your chatbot content database. This becomes more difficult to ensure when content is automatically scraped, e.g. from website. It is important for you to have procedures in place for ensuring the quality and non-harmful language of any content.

5.1.2 Collation methods

For the collation of content of a chatbot, there are two general methods: manual and automatic content collection. Both can complement each other at different points during the development, deployment and maintenance of a chatbot.

Manual content collation
Whenever the action of one or more persons is needed to access, collect and make available content for a chatbot, we are dealing with manual content collation. While this can be time- and cost-intensive, there are several situations during the development and maintenance of a chatbot, when manual content collation may be helpful:

- **Initial content base:** For curating the very first content base for a chatbot, it may be advisable to engage domain experts for collating, quality-checking and preparing the content. While automated mechanisms may be helpful e.g. for scraping content from a ministry website, it is important that the core content base only includes content items that are verified for being factually correct and that come from trustworthy sources. Furthermore, in certain cases it might be necessary to add certain metadata information to the content which might be useful in design flow or might serve as benchmarking data for quality testing.

- **Translations:** For content that needs to be translated for usage in a chatbot, it may be advisable to engage professional translators that understand the cultural context of the language speakers as well as are familiar with the chatbot domain. While methods for automated translation exist (e.g. machine translation), such methods only work reliably in domains for which the automated machine translation tool has been trained.

- **Quality checks:** Whenever new content is added to a chatbot – whether via manual or automated mechanisms – it is important to have quality assurance mechanisms in place. At minimum, this should include at least one domain expert who can review content added to the chatbot – and flag any inconsistencies, or needs for revision.

Automated content collation
Whenever content is acquired through mechanisms that do not involve any person to be involved, but instead e.g. scripts for auto-download or auto-scraping from a website, we are dealing with automated content collation. There are several situations where this can be useful:

- **Scraping websites:** When your chatbot uses information from trustworthy standard website, e.g. ministerial website for a certain sector, you could use a web crawler for auto-scraping of content from the website(s) in regular intervals – to ensure that any content updates on the website(s) reflect in the chatbot. This could, for instance, include websites of ministries who will be offering the chatbot to users. Importantly, this will require approval from the respective website owners.

- **Regularly updated items:** Similar to standard websites, your chatbot may be using content from items that are updated regularly, such as annual reports, guidelines, specific domain manuals etc. If these items are electronically accessible, you could use scripts to auto-download updated/new document. Depending on the source, this may also require content extraction (e.g. from PDFs) as well as quality assurance methods.

- **Feedback mechanisms:** Your chatbot might include a feedback function to allows users to ask questions or point out incorrect information. Such feedback content may also be automatically collected – but rather than feeding it directly into your content database, you should make it part of an 'update mechanism' (see chapter 5.3)

Automated content collation mechanisms will require regular hands-on check-ups to ensure that content sources remain accessible, content transfers remain functional (e.g. adjust to any changes in source formats) or content quality continuous to meet expected standards.

5.1.3 Content formats

The format in which collated content needs to be prepared for chatbot usage depends on the type of chatbot you are developing (see chapter 3.1). Below we are outlining common formats and respective methods for content preparation for the most common chatbot types:

- For **FAQ chatbots**, the content needs to be prepared in question-answer pairs that are used for sourcing answers based on the analysed intent of user questions.

- For a **Generative AI chatbot**, the content needs to be prepared in a content database that is used for generating the output, e.g. via retrieval-augmented generation (RAG).

- For a **hybrid chatbot,** you will need content for non-AI-interaction features such as USSD or IVR, which are not addressed in this guideline, as well as content prepared for the AI chatbot function (FAQ chatbot and/or Generative AI chatbot).

FAQ chatbot: Question-Answer-Pairs
FAQ chatbots use NLU to analyse the intent behind a user question. As response, the system looks up a pre-defined answer that corresponds with the identified intent. This means that all answers that an FAQ chatbot provides are pre-defined and no new answer texts are generated (unlike Generative AI chatbots). This makes it easier to ensure the quality and factualness of responses. But it also requires a large content base to cover a wide range of user queries.

For fine-tuning the NLU model (non-generative language models) and providing the list of pre-defined answers for FAQ chatbots, you usually need to create a tabular list of questions and answer pairs (QAPs) that cover your chatbot's domain as widely as possible. The structure of these tabular overviews may vary slightly depending on the chatbot framework you use, but generally they consist of a CSV file containing two columns (question, answer) followed by rows of entries. Most frameworks allow you to define multiple, alternative questions with the same answers by adding rows with questions only and leaving the answer fields blank following the overall correct answer.

As mentioned in chapter 3.1, FAQ chatbots can also be developed using LLMs. In such cases, no training is involved, and answers would be generated via direct LLM calls whereby the question-answers pairs serve as content database basing on which LLM answers the user query. However, the use of LLMs may increase development and maintenance costs which is why this guide also includes these simpler types of FAQ chatbots.

Generative AI chatbot: Document database and Test-cases
Generative AI Chatbots use LLMs both for analysing the intent behind a user question and for generating a response to the question. A knowledge base with domain-specific information is used for generating contextually relevant responses. This can be in the form of a document database including various pieces of information relating to the domain of the chatbot – e.g. websites, PDFs, Word files. Importantly, this database will need to be processed to be in a machine-readable format for usage with an LLM – see chapter 6.2.3 "Knowledge Database".

Additionally, a tabular overview of question/answer pairs (QAPs) should be created as test cases with which the efficacy and factualness of the Generative AI Chatbot can be tested.

Good practices for collating content
Here are some tips and good practices for creating the content base:

- **Involve domain experts:** To cover a wide range of knowledge in the domain of your chatbot, you should involve experts for curating content as much as possible. They can help you put together a comprehensive knowledge base, identify core documents and draft likely questions users would ask to access such knowledge.

- **Incorporate existing FAQs:** In some instances, FAQs may already exist that you can use for creating the knowledge base, both the QAP table or the document table. For instance, a ministry may have a list of commonly asked questions and an overview of relevant reports, manuals or other documents – either publicly on their website, or as an internal document for staff working in helplines.

- **List sources:** For increasing the credibility of the chatbot, you may want to consider adding sources to responses, such as URLs, source documents or similar metadata information.

- **Build on user feedback and requirements:** It is crucial that you get feedback from users to add and adjust content that more closely reflect what information users would expect from the chatbot. This feedback can be gathered during initial testing/piloting of the chatbot as well as during the actual operation of the chatbot – e.g. by documenting/logging user queries for which the chatbot could not find an appropriate response in the knowledge base or the answer was sub-optimal.

- **Check for duplicates:** To improve the quality of the chatbot and its responses, it is recommended that you review your content base for duplicate or very similar content which can be removed to reduce redundancies.

- **Create a train/test split:** You should always create the train-test split out of the QAP data. While the training data can be used for tailoring an AI model for entity or intent recognition, or making model choice for information retrieval (example RAG chatbot) it would need test cases to test the performance of the underlying AI model.

- **Include Metadata:** The metadata is very important when working with Generative AI chatbot as it greatly enhances the capability of model to pin point the particular document or content which might be most suited to user query. Furthermore this metadata can be used by chatbot solution design to steer the user conversation.

5.2 Validation & Testing Content

The validation and testing of content is a crucial 'sanitization step' of content collation and creation of the chatbot's knowledge base. The validation step ensures that the important concepts like definitions, taxonomy and hierarchical structure of QAPs/documents are aligned and respected in the content. The testing step ensures that content is line with the chatbot owners' and users' expectations for performance, as well as the factual correctness of content.

For testing the performance and factual correctness of the chatbot, you can employ both quantitative and qualitative methods. A combination of both will help ensure best results but if you have to choose one then qualitative analysis should take precedence as it promises the most valuable user feedback for fine-tuning and improving the chatbot:

1) **Quantitative Analysis:** For the quantitative testing of any chatbot it is important that the test-cases are created with help of users and expert and properly documented. The test cases should be good representation of the actual nature of queries that the chatbot is supposed to serve. This benchmark not only serves in testing the chatbot but also allows testing of various LLM's. Furthermore, the focus is not just about what the chatbot should do but also considering what the chatbot is not supposed to do, it is very important to design the test cases with edge case scenarios which should cover wide spectrum of what user might input.

   Test-case: It's a set of conditions or variables (example user input) with expected results which help the tester to evaluate if the chatbot performance is satisfactory and correct. A very simple guide to frame the test-cases might look like this:

   Scenario/Requirement Mapping >> User Input >> Expected Output (with sources)

© GIZ / Dirk Ostermeier

Testing of the chatbot with experts and users is important to create a robust version for publishing

2) **Qualitative Analysis:** The qualitative analysis is carried out with actual users for the assessment of chatbot performance and factualness. This qualitative testing is crucial for aligning the chatbot outputs with user expectations. Whereas quantitative analysis is more technically geared toward measuring chatbot performance, qualitative analysis with users will be most impactful for the success of the chatbot and, hence, should take precedence over, or be complemented with, quantitative analysis. Qualitative analysis can take place at various points of the chatbot development:

   a) **During piloting:** During the piloting of the chatbot, qualitative user testing can be incorporated to test the initial content base of the chatbot and, early on, correct and erroneous content and fill any content gaps that may be identified. Doing a thorough qualitative analysis during piloting will ensure that a robust version of the chatbot is published for public usage.

   b) **During regular operation:** Throughout the operation of the chatbot, qualitative analysis of the chatbot can be incorporated (a) as part of the chatbot itself via a feedback mechanism that allows users to respond to wrong or missing content; (b) via planned in-depth analysis periods for running stress tests of the chatbot with a planned group of expert and amateur users.

5.3 Keeping content up-to-date

Regular updating and maintenance of the cha tbot's knowledge base is important for keeping it performant and relevant to users throughout and beyond the project life-cycle. There are several mechanisms that can be used for maintaining updated content considering update methods, user feedback, and other relevant factors:

- **Database update mechanism:** This involves having a well-defined periodicity of document/database update and review. A chatbot content plan which lists sources and methods as well as frequency of updating can help putting such an update/review process in place (see chapter 5.5 "Checklist: Chatbot content strategy"). Generally, such an update mechanism could include:
  - Technical aspects, such as document versioning control, manual and/or automated updates
  - Sanitization (=cleaning) of incoming data e.g. by removing personal information or confidential data
  - Tagging of documents/content elements for better classification
- **Feedback incorporation:** The chatbot should include a function to allow users to give feedback on

outputs and content. This helps identify erroneous content or content gaps. Such user feedback should be actively used to adjust, revise and complement the content to user expectations (defined within the limit of chatbot scope).

- **Content Management System:** When the content is to be periodically updated for the chatbot to function as per users' expectation, it would be important to consider having a Content Management System (CMS) for the chatbot content. This would also require training of the partners who host the chatbot for self-management of such a CMS.

5.4 Content localisation

Depending on the anticipated target audience of the chatbot, you may need to localise content by translating it to one or more local language(s), especially in contexts where this is necessary for ensuring wide and equitable accessibility to the chatbot.

There are several considerations you should keep in mind when localising your chatbot, or operating multilingual chatbots:

- **Existing multilingual sources:** Some ministries or institutions may have documents, reports or websites in multiple languages by default. Before translating any sources, you should check with your partner institutions for such bi-/multilingual documents. For instance, the eCitizen portal in Kenya is available in English and Swahili; or in South Africa, many official documents are translated into all official languages.

- **Manual translation:** For domain-specific texts and sources, you may need the services of professional translators to support with factually correct translations. For any texts that are copyrighted, you should check with the original authors under what conditions you are allowed to utilise and/or publish translated versions of their texts.

- **Automated translation:** If machine translation (MT) models or services exist for your chatbot languages, you can also test the usage of automated translation of source texts. However, you need to ensure that the MT models perform well within the domain of your chatbot. If MT models exist but do not work in your domain, you could use manually translated texts for fine-tuning the MT model for better performance in your domain. In those cases, you need to check the licensing requirements of the MT model and whether usage conditions of texts translated by you allow you to re-share the fine-tuned MT model or not. In any case, you should always include a quality assurance step for automated translations – e.g. by sending a random sample of auto-translated texts to a professional translator and/or native speaker for validation.

- **Multilingual LLMs:** There is an increasing number of multilingual LLMs that are being developed and offered, some also as open-source or open-weights. For using them, you need to ensure that your chatbot language(s) is/are sufficiently covered by the LLM and to test this thoroughly with native speakers of the respective language(s). If it works well, the use of multilingual LLMs can save you translations of content.

- **Dialects & Code-Switching:** When translating texts into local languages, you also need to consider dialects spoken in the regions where you intend to use the chatbot – and engage with translators you are proficient in such dialects. For instance, a chatbot with texts in Kenyan Swahili may not be readily understood by citizens who speak Tanzanian or Congolese Swahili – and also within countries there are variants between more formal Swahili and regional dialects.

- **STT-/TTS-models:** Needless to say that if you use STT or TTS models as part of your chatbot architecture, you need to consider using such models in local languages. At the time of writing this guide, there is a growing number of initiatives working on such models – e.g. the Masakwhane community of pan-African academics working on NLP for African languages.

- **Testing with speakers:** Importantly, any localised content needs to be thoroughly tested with native speakers of your chatbot language(s). This should be an integral step during chatbot development, followed by periodical tests throughout chatbot operation.

5.5 Checklist: Standard procedures for content collation

If you know that manual and/or automated content collation will be necessary throughout the operation and maintenance of your chatbot, you may want to put in place standard procedures in your content strategy, for both efficiency and resource-planning purposes. This may include:

- ☐ **Content source:** Where does the content come from
- ☐ **Content type:** What sort of content is it – e.g. website, PDF…
- ☐ **Collation method:** How is the content collected, manually, automatically
- ☐ **Added on (date):** When was this content source added?
- ☐ **Involved person(s):** Who is involved in collecting the content? Who is the appropriate contact person?
- ☐ **Refresh frequency:** How often if the content updated (e.g. once, quarterly, annually)?
- ☐ **Content preparation method:** What steps are necessary for processing the content for usage in the chatbot?

6 Technology: Building the chatbot's technical backbone

A chatbot's performance and effectiveness are rooted in its technical architecture. This chapter outlines the core technological components forming the chatbot's backbone:

- the user interface, incl. input modalities, delivery platforms and dialogue management
- the fulfilment core, incl. NLP, knowledge base structure and system integrations

Together, these components enable seamless user interaction and a valuable experience.

For discussing the technological component, we need first to categorize the chatbot types. Chatbots categorization or types can be defined based on various parameters but in this guide, we define them based on two aspects:

1) Functionality or Purpose

2) Technological

The categorization in terms of functionality or purpose is helpful when deciding what type of chatbot may be most suitable for a particular project. For guidance on this, see Chapter 3.

This chapter provides guidance on the technical aspects and components of chatbots. While specific implementations may vary, the general solution design presented below remains consistent across most chatbot architectures. We will use this design as a foundation for examining individual components in detail.

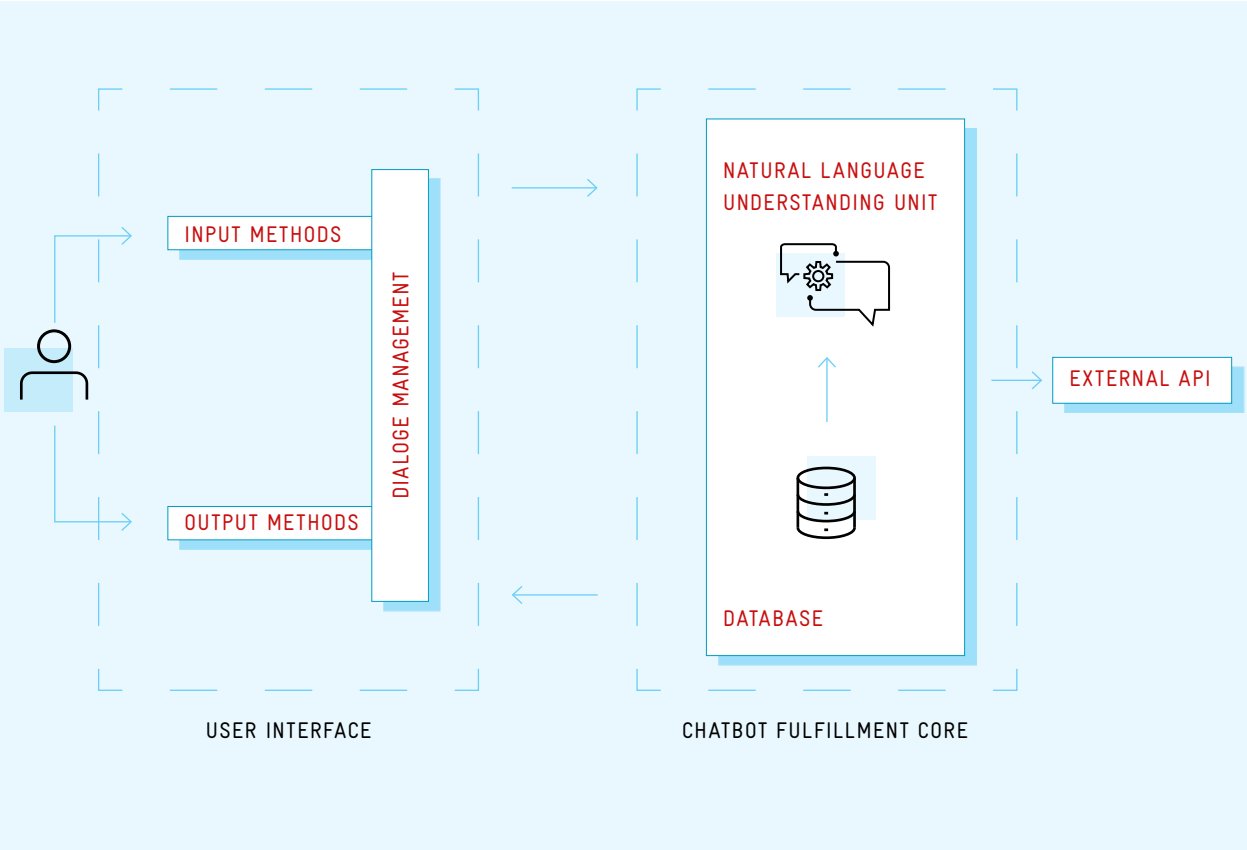


Figure 3.: Chatbot Solution Design

6.1 User Interface

The User Interface (UI) defines the interaction phase between user and the chatbot, which can significantly impact the chatbot's success. The focus here is user-centred aspects rather than underlying backend technology. While the field of UI has its own principles however for chatbots discussion here we will focus on two main components of chatbots: Input Modality and Delivery Platform.

6.1.1 Input Modalities

This refers to the type of data or 'What' a user can provide to the chatbot. The common modes include text, image, audio and video. This means that the user can either type/upload or speak into the device. The choice of input modality influences the chatbot design and user experience.

What influences the decision on inclusion/exclusion of input modes?
As explained earlier, the focus here should be the "User". When developing the chatbot take your user demographic into account, especially their literacy level, age, technological affinity and skills as well as accessibility needs. Moreover, consider the leave no one behind principle: How can the chatbot be used by vulnerable groups such as the elderly, rural population or people with disabilities? Below are some examples where certain input modalities are recommended.

- **Limited writing skills:** While text-based conversational chatbots are on the rise, in many developing countries the population percentage with verbal communication skills far exceeds the written communication skills. Input modalities with voice interaction would make the most sense in this case.

- **High linguistic diversity:** Chatbots are highly dependent on which language model they operate on. For example, your chatbot might run smoothly in Swahili but your users struggle to communicate with the chatbot in Swahili because they speak a different dialect or use a mix of local languages and English to communicate. In this case, users might find it easier to use voice input and output than write/read Swahili which can be too academic for the average user.

- **Impairments:** For users with disabilities certain input modes are essential. For users with hearing impairment, voice interaction is not possible

6.1.2 Delivery Platform

While the Input Modality handles 'What' users can input, the delivery platform covers the 'How' aspect of user interaction. Platform and input types are not independent of each other and need to be considered together. While not strictly hierarchical, some common categories include:

- **Online:** websites, widgets, browser plug-ins, mobile apps, email, social media, messaging apps
- **Offline:** USSD, text messages, call functionality, offline functionality of mobile apps
- **Telephone:** Chatbot is available via normal telephone.

It's important to note that the delivery platform or channel can impose limitations on the available input modalities and vice-versa. For users to engage with your chatbot, it needs to be accessible in ways that serve both the supplier of the chatbot (e.g. a government unit, or an agricultural service provider) and the audience which you are building the chatbot for (e.g. public, farmers, health professional). Similarly, the interplay between delivery platform and user demographic can be the deciding factor between chatbots' success and failure. Therefore, accessibility is an important factor to consider and should follow the principle of digital inclusion and not lead to widening the digital divide. This means that the input modes should consider the user groups' access to the service:

- Does the user have access to the internet?
- What cost will the user encounter when using chatbot services?

For example, USSD, while cost-effective, widely accessible and available offline, has the downside of limited-length text input. On the contrary, a text-based chatbot delivering citizen services is unlikely to succeed if the target user group primarily relies on social media for communication and information.

6.1.3 Dialogue Management & Feedback

Dialogue Management refers broadly to user history and helps the chatbot in understanding the conversation context and to decide on next steps. Simplified, the chatbot uses the input/feedback by the user to get more context and to provide a more suitable response. This is primarily a focus for chatbots powered by Large Language Models (LLM).

What influences the decision and determines the importance of dialogue management in the chatbot design?

- **Purpose:** The chatbot's scope/purpose is limited to question-answering with limited interaction or is designed for multi-step conversations. While the first case doesn't need sophisticated dialogue management, the latter might require it.

- **Contextual Awareness and Personalization:** If the chatbot is supposed to provide standardized responses to all users, dialogue management can be simple.

If the response needs to be tailored to user preferences and past interactions, dialogue management needs to track and utilize this information. The latter is often facilitated by reassuring that the chatbot understood the user's context. Methods of such include:

  - Questions summarizing the context: 'Is this what you mean?' or 'Does this answer your question?'
  - Thumps up or thumps down emojis
  - Prompts to select from a menu of answers or images to narrow down the context

- **Input multi-modality:** If the input users can provide is multimodal then it might warrant for complex dialogue management. For example, the user could start with text but later would upload an image to describe the prompt in more detail.

- **Data Privacy and Protection:** Sophisticated dialogue management which stores past user interaction can expose the chatbot administrator/manager to data privacy and compliance regulations (e.g. GDPR). This includes but is not limited to data retention policy, user consent and user control over their own data.

6.2 Chatbot Fulfilment Core

The core processing unit of a chatbot can range from a simple, single decision-making unit to a complex interplay of Large Language Models (LLMs) and databases. To understand these varying levels of complexity, it's essential to categorize chatbots based on their underlying technological framework.

Following recent AI advances, especially regarding LLMs, it is now common to distinguish between rule-based chatbots and LLM-powered chatbots (focusing on Generative LLMs especially). The table below outlines how these two technological chatbot frameworks can be mapped against the three functional types described in Chapter 3. However, it is worthwhile noting that FAQ chatbots can also be built with LLMs though this guide presents them as a non-generative LLM alternative to how chatbots are built (given the costs involved in developing and especially maintaining generative LLM-based chatbots.

FAQ CHATBOT	HYBRID CHATBOT	GENERATIVE AI CHATBOT
Uses NLU to understand a question and responds to with predefined answers from an existing knowledge base	Combination of an FAQ chatbot with a generative AI chatbot and, in some cases, a link to live chat or hotline	Uses LLMs to generate new content in response to a user question. It can be fine-tuned to specific knowledge domain.
RULE-BASED CHATBOT FRAMEWORK		
	LLM POWERED CHATBOT FRAMEWORK	

Rule-Based Chatbots

Rule-based chatbots represent a simpler architecture, relying on pre-defined rules and decision trees to generate responses. Referring to the flow diagram provided below these chatbots operate primarily through two key components "User Intent Recognition" and "Rule-based Response." While intent recognition can be dynamic, the response retrieval is essentially static. The User intent recognition can be a simple decision tree which works without any AI model; however, it can also be an AI-powered unit (small language models performing the task of entity recognition more on this in section 6.2.2).

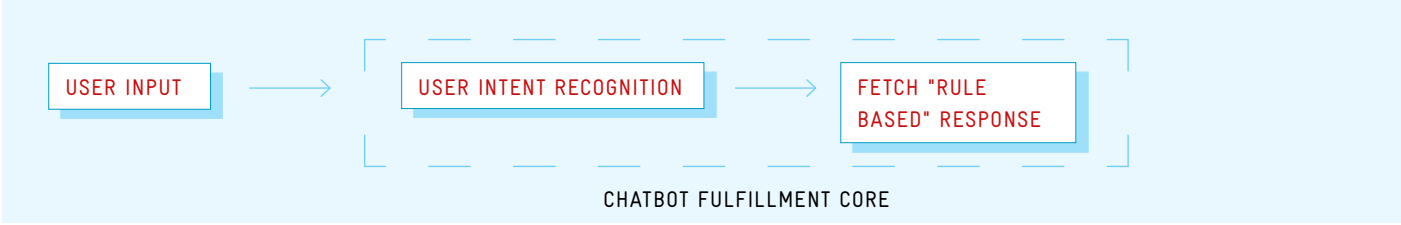


Figure 4.: Rule Based Core

Consider the User Input: "How can I connect my mobile number to my bank account"? A rule-based bank chatbot might have a pre-existing FAQ database. Instead of requiring users to search through documents, the chatbot attempts to match the user's query to relevant Question-Answer (QA) pairs. A simple rule-based system might identify keywords like "bank account" and "mobile number" then present the user with stored answers containing those terms. If no suitable answer is found, a default response directing the user to customer support might be provided.

While this example seems straightforward, rule-based chatbots can become more sophisticated by incorporating decision trees (e.g., "If A then B, If B then C else D, if not A then E"). This allows for more nuanced intent recognition and response selection. However, the fundamental difference between rule-based chatbots and LLM-powered chatbots is that rule-based systems draw from a repository of **pre-stored responses**. They select the **most appropriate existing answer** rather than generating a suited response.

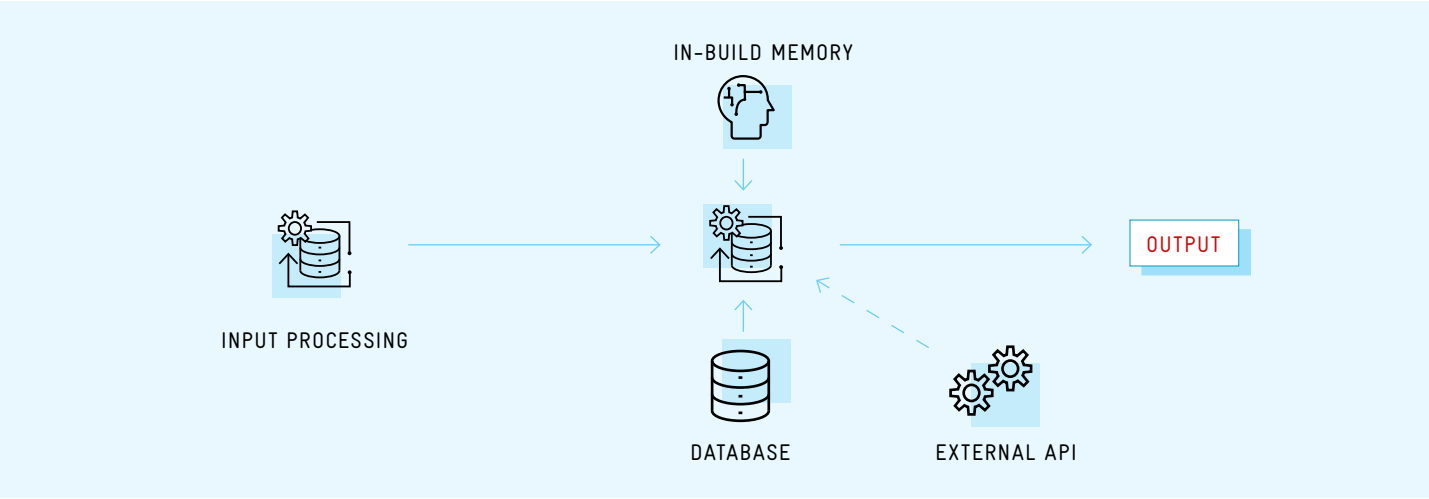
When to use Rule-based chatbots?

- Ideal for simple, repetitive tasks with a limited scope which need predictable and consistent responses.
- When computational resources are limited, rule-based chatbots are a good option as they don't require significant computing and often rely on simple key-value stores (database).
- When resources are limited the moderate running and maintenance costs makes them suitable option.
- Explainability in case of Rule-based chatbots is one of the biggest advantages and are safe option when regulatory compliance is a concern.

There are limitations to rule based chatbots as they struggle with nuances of language, adapt poorly to unexpected requests, and require tedious manual updates. However, their key limitation is their inability to generate responses, relying instead on pre-stored answers.

LLM powered Chatbots

This document focuses on AI-powered chatbots, while no single architecture definitively captures the core of these systems, for simplicity, we will use the diagram below to explore their technical architecture.



6.2.1 Input Processing

User input can range from simple instructions or questions to complex, multi-layered requests. Processing this input is therefore often a multi-step process. We categorize this as follows:

- **User Intent Recognition:** This crucial step determines how the LLM will handle the user's request, guiding the LLM's subsequent actions and response generation.
- **Instructions:** These explicit instructions by user serve as guidelines for the chatbots for processing and response formulation.

The determination of user intent and extraction of explicit instructions which chatbot need to follow is performed by the Natural Language Understanding unit which itself leverages an LLM. However, the tasks that Natural Language Understanding unit performs are not limited to these and goes beyond in recognizing sentiment analysis, entity recognition (names, dates etc) and more. In case of FAQ chatbot the NLU performs entity recognition task which might be used then by a rule-based decision tree to get the pre-stored responses, however in case of Generative and hybrid-use cases it might involve much more complex tasks like query decomposition and execution planning (agentic chatbots).

- **Document/Input handling:** The user can provide explicit input in form of document/image etc which cannot be directly feed to LLM and hence need to be processed to make it ready to be ingested by the LLM for formulating the response to the user request.

How the Document/ Attachments Handling impact infrastructure, cost and compute and its implications? Document/attachment handling significantly impacts infrastructure, cost, and compute resources. It introduces complexities like storage for uploaded files, processing services (e.g., OCR, text extraction), and increased compute demands. These factors directly translate to higher infrastructure and processing costs. Furthermore, complex input processing requires substantial computational power to mitigate latency. Therefore, we recommend the following guidelines:

- Prioritize a regularly updated knowledge base and add document/attachment handling only when essential. Example: Chatbot using periodically updated knowledge base while can run on infrastructure with CPU based machines, but the input handling might require GPU based machines.
- Start with simple input options (e.g., text pasting) and progressively add support for other input types in later releases.
- Optimize input processing through caching (with careful consideration of data protection), asynchronous processing, and batch processing.
- Prioritize latency management, especially in resource-constrained environments, as it critically affects user experience. It is recommended to keep the latency < 5 seconds.
- Address data security and protection carefully, particularly if storing data or using external APIs.

6.2.2 In-build memory

Once the LLM has determined the focus question of the user, it relies on its own memory or answering capability to formulate the response to user request.

What forms the inbuilt memory of LLMs? The LLMs are trained on huge corpus of data which is existing on the world wide web in open-source domain, while some company report on training data but mostly it is not known what has gone into the training these models. But it's safe to assume that these are general purpose language models which have seen lot of information available on internet and have learned the nuances of the language and learned in process the facts and gathered general knowledge.

Things to consider:

1) **The cut-off date:** The group publishing the LLMs cite the knowledge cut-off date. This means that any new information created or event which occurred after this cut-off date is not available to the LLMs.

2) **Knowledge base:** Most of the LLMs are general purpose models which had been trained to understand the nuances of Language and the conversational aspect. However, as companies like Google, Open AI and others keep feeding more and more data and keep striving towards making their model versatile for all nature of tasks (Mathematical proofs, Philosophical reasoning etc) these models are advancing rapidly, however there are domain specific models like BloomberGPT (Financial data), ClimateBert (climate domain) etc which might be more suited for the use-case.

How Task complexity and all the factors mentioned above influence the choice of LLM?

Task Complexity and LLM Size:
- Smaller LLMs (<10B parameters) or fine-tuned models efficiently handle basic intent recognition and entity extraction, such as dates and names, making them ideal for cost-effective solutions like FAQ chatbots.
- Larger language models are essential for chatbots handling intricate intents and a broad spectrum of entities, including context-dependent ones. These models excel at managing multi-step instructions and complex reasoning. For a citizen services chatbot assisting with government applications must navigate multi-step inquiries, such as verifying eligibility, collecting required documents, and scheduling appointments might warrant to use large LLMs (>10B parameters).
- If the chatbot needs to process and understand the content of documents, images, or other complex inputs, and integrate that information into its responses, larger LLMs with multimodal capabilities (or access to specialized models for those modalities) are often required.

Task Complexity and LLM type:
- For many chatbot applications, a general-purpose LLM (like GPT-4, Gemini, or similar models) that has been trained on a massive dataset are good. These models are versatile and can handle a wide range of tasks.
- If you have a specific domain or use case (e.g., medical, legal, financial), fine-tuning a general-purpose LLM on a dataset of examples from that domain can significantly improve performances or explore existing domain specific specialized LLMs.
- If your chatbot needs to process images, audio, or video, you'll need a multimodal LLM that can handle these different input modalities.

Constraints for LLMs choice:

- Larger LLMs are generally more expensive to use. Consider the budget and the trade-off between performance and cost.
- Fine-tuning requires a substantial amount of labelled data. If you don't have enough data, you might need to use a general-purpose LLM or explore other techniques.
- Larger LLMs can also have higher latency (i.e., they take longer to generate responses). If your application requires fast responses, you might need to use a smaller model or optimize the inference process. However, smaller LLMs can suffer from quality of output generation.
- Modular approach for solving the use-case with multiple smaller LLMs rather than relying on single larger LLMs might be more efficient (quality and cost), however it can incur overhead cost which need to be carefully looked.

Assessment Criteria: Similar to any tender evaluation it is good to create Assessment criteria's for selection of LLM and then through a weighted scoring make an objective choice for LLMs.

EXAMPLE:

CRITERION	WEIGHTING	LLM A SCORE	WEIGHTED SCORE (A)	LLM B SCORE	TEAM &LOCAL PARTNERS
ACCURACY	5	4	20	5	25
COST	3	3	9	2	6
LATENCY	2	5	10	3	6
TOTAL			39		37

LLM-on-premise vs Open Source LLM as a service vs Proprietary LLM as a Service
In general, there are three ways to integrate LLMs:

- **LLM-on-premise:** You host an open-source LLM on your own server. This meets highest data protection standards because the conversations with the user do not get send to external parties. This is very cost-intensive, because it requires an expensive server. Also, it is a technically challenging setup and server administrators might need additional training. Furthermore, it raises problems of scalability: If many users access the chatbot in parallel and the capacity of the first LLM is reached, you need to buy additional expensive servers.

- **Open Source** LLM as a service: Many companies offer Open Source LLMs as a service. Instead of buying a server and hosting the LLM yourself, you pay the LLM per usage, which is usually cheaper and also, easier to administrate. Since the LLM itself is open source, you can switch to the LLM-on-premise model later on. It still raises problems of data protection because conversations are sent to external servers, although such servers might be located in Germany and follow the strict German data protection standards.

- **Proprietary LLM as a Service:** This is similar to Open-Source LLM as a service, with some important differences: Proprietary LLMs are usually created by US Big Tech Companies (e.g., OpenAI, Anthropic, Google) and are the technically most advanced and powerful LLMs. For some low-resource languages, only these LLMs have a satisfactory performance. If you want to switch to the LLM-On-Premise model, you do need to change the LLM, because you cannot install proprietary LLMs on your own servers. Usually, servers are located in the US and follow US data protection laws, which is a blocker for many applications.

6.2.3 Knowledge Database

The database is the knowledge base which is put in place so that when chatbot is formulating the response to user query it can fetch the contextually relevant information. In previous section we discussed the knowledge cut-off and subject expertise related issue, combine this with all known issue of 'hallucination' the

knowledge base serves as way to overcome these issues. In addition, the use-case might require that the chatbot only answers based on verified or reliable information source (Retrieval Augmented Generation) then the database serves this exact purpose. The Chapter 5 on Content Management (Managing Knowledge base) already addresses certain key aspects for preparedness and readiness on this topic, however here we address technical details on how such database should be and what to consider.

What to consider when creating Knowledge base?

- **Document processing:** The LLMs cannot directly work with the document database. This means that the documents first need to be processed and put in the formats which can be easily ingested by the LLMs. This is very crucial step and need careful formulation of document processing step depending on the nature of file types, document structure (tables, tables within tables etc), extraction of text from image type pdf, extracting text from various files types (PDFs, Docx, html, xml etc).

- **Chunking:** Chunking is the step in the knowledge base creation where the large text is broken into smaller text parts (called text chunks). While many LLMs now days can ingest very long text let's say from ~ 8000 words to 120k words (this limit is being pushed very fast with new model releases), but it is advisable to break the text from documents into smaller lengths for 2 reasons mentioned below. The chunking size while depends on the use-case it is always advisable to have 2-3 different chunk sizes like small (100-200 words, medium 400-1000 words, large ~ 5000 words)

  1) Smaller the text the LLMs is provided based on which it needs to answer the user query, the more specific the output from LLMs. This also alleviate with common known issues of LLM's not able to focus on all key aspect of input especially in long context.
  2) Smaller leads to better utilisation of compute resources and help with latency issues in the chatbots design (especially in RAG framework).

- **Metadata:** As LLMs will rely on this document database to answer the user query, it is advisable to enrich the database with metadata. The metadata can be as simple as extracting page number to more complex like associating each text chunk with Header sectional information. This is important because most of the modern chatbots the metadata helps to narrow down or find the text part from documents which might be most relevant for the LLMs to answer the query.

- **Vector-Database:** The document databases used in chatbots are not like the usual databases like storing the extracted texts in some folders/files. Consider that you have 1000 documents (~20 pages each) and each document when has gone through chunking step with medium length of ~ 500 words, this can create the database of texts of around ~50000. Looking up from these 50000 files to find which one is relevant for the user query can lead to in-efficiency and therefore this requires special type of storage called Vector databases which are memory and compute efficient.

6.2.4 External API's and Tech stack examples

Chatbots are often equipped with external API's/services and tools thereby enhancing their functionality. This is done primarily to improve information retrieval or to enable them to perform functions beyond the conversational aspect. Example Include like connecting Maps based API to get real time info, providing API excess to search engine to get real time information on a particular topic, accessing some booking platform thereby enabling appointment booking.

Below is the list of suggestive tech-stack options for each component:

Vector Stores:
Qdrant, Milvus, Weaviate, Pinecone, Chroma

Document Processing:

- Paid services from all major cloud services providers like Azure, Google, AWS including some IBM etc
- Self-Managed (Open-source): docling, axa-parsr, pymupdf

Front-end: React, Vue-js, Angular or CSS, Javascript or any other popular front-end frameworks.

Backend and Dialogue-Management: Langchain, Llamaindex, Rasa or Haystack (many of these framework have well developed ecosystem which provide seamless integration with Paid LLMs services like OpenAI, Google etc and also with other components like vector-stores etc) and need just the custom front-end development for users.

While this tech-stack is suggestive there are many other components that depend on use-case. For example, if the team wants to have detailed analytics on use of chatbot or admin portal then this will require custom incorporation of Monitoring and Analytics like Google Analytics etc.

6.3 Open-source

The landscape of chatbot development is rich with open-source solutions, offering viable alternatives to paid services for nearly every component. This is a significant advantage, particularly for organizations like development cooperations with budget constraints or a commitment to community-driven development. The open-source community is actively developing and

improving these tools, rapidly closing the gap with paid solutions in terms of functionality and performance.

Key Open-Source Components and their Advantages:
Natural Language Understanding (NLU):

- Rasa: A comprehensive open-source framework specifically designed for building conversational AI assistants. Handles intent recognition, entity extraction, dialogue management, and bot training. A strong competitor to paid platforms like Dialogflow.
- Hugging Face Transformers: Access to a vast collection of pre-trained transformer models (BERT, RoBERTa, etc.) that can be fine-tuned or used out of the box for NLU tasks. Provides state-of-the-art performance, though requires more technical expertise.

Dialogue Management:

- Rasa: Rasa excels in dialogue management, providing a rule-based and machine learning-driven approach to managing conversation flow and context.
- Langchain: LangChain's memory capabilities to manage conversation history and its ability to chain together components like (LLM's, prompt, tools/API) can be used to implement basic dialogue management logic and flows.

Backend Logic & Integrations:

- Python (Flask, Django): Robust and widely used for building chatbot backends. Flask is great for simpler bots, while Django is better for larger, more complex projects.
- Node.js (Express): Another popular choice, especially if the team has JavaScript expertise.

Vector Store:

- Qdrant: A fast and scalable open-source vector database, ideal for knowledge-based chatbots. A strong alternative to Pinecone.
- Weaviate: Another excellent open-source vector database with a GraphQL interface.
- Chroma: Open-source vector database.

Large Language Models (LLMs):

- While access to the most powerful LLMs (like GPT, Claude or Gemini) is typically through paid APIs, the open-source community is making strides. Projects like Llama, Gemma and others provide increasingly capable open-source LLMs that can be self-hosted. However, self-hosting LLMs requires computational resources. Please refer to section 6.2.2 for how LLM's might be consumed through different modes.

By effectively leveraging the power of open-source tools we can build high-quality, cost-effective chatbots that serve their mission and contribute to the broader community. The emphasis on open-source not only reduces costs but also promotes transparency, collaboration, and knowledge sharing, aligning perfectly with the values of many development-focused organizations.

6.4 Chatbot Architecture examples

With the foundation technical concepts explained above we can explore how these elements are integrated in a functioning chatbot using some popular technical architectures.

6.4.1 Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is a critical and a popular technique that significantly enhances chatbot's capabilities by connecting the Large Language Model (LLM) to a proprietary, up-to-date, or specialized external knowledge base (like our company documents). Instead of relying only on the model's static training data, RAG allows the system to first retrieve relevant information based on the user's query and then use that retrieved context to generate an accurate, grounded, and specific answer, thereby reducing hallucinations.

Consider the simple web-based RAG chatbot which has the key components:

- Vector Database (Knowledge Database): This is the knowledge database which is scanned based on user query to find the relevant files/documents/text in database which will form the base on which the user query will be answered.
- Retriever (NLU/LLMs): This is the component which is responsible for finding relevant information from vector database and then passes this fetched relevant information to Generator to formulate the response to user query.
- Generator (LLMs): This Generative LLM which takes the user query and output frm retirever to finally answer the user query.
- Dialogue Management: Dialogue management in case of RAG can be complex which maintains long conversational history or might constitute a simple static Prompt/Instruction which are fed to Generator with every user query on how to formulate the response to user query.

6.4.2 Agentic Chatbot

Let's consider that in the example above, if the answer to the user query cannot be answered from the knowledge database (RAG), then probably it might be a good option to make some external API call to find relevant information and then pass this information to the Generator. However, to do such a manoeuvre might require some NLU capability integrated with Dialogue Management to decide when to make and when not to make the external API call. This is one such very simple example of an Agentic Chatbot. Most fo the enterprise tools like ChatGPT, Gemini etc are example of agentic chatbots which use complex interplay of multiple LLMs/NLU, external API calling etc.

7 Responsible use of AI chatbots

Chatbots can be very useful tools to make information more accessible and service delivery more inclusive because they allow citizens to engage with them in their own language through either text, voice or both. GIZ projects have implemented chatbots successfully in sectors ranging from agriculture to health and public service delivery (see Spotlights in Chapter 8).

Chatbots should ideally serve their intended purposes well and not spread misinformation or even cause harms to their users. To ensure this, chatbots must be implemented responsibly and with full awareness of both the potentials and pitfalls of using chatbots.

To support you with responsibly implementing AI chatbots, this chapter provides insights into:

- Risks and responsibilities for the implementation of chatbots in projects
- Limitations of LLMs as the key technology underlying generative AI chatbots
- Understanding and mitigating climate impacts of AI in chatbots
- Data privacy and data protection considerations when using chatbots
- Aspects to look into for responsible chatbot implementation in form of a checklist

"DO AFRICANS REALLY WANT CHATBOTS?" (PROF. VUKOSI MARIVATE)

In his keynote speech at the AfriCHI Conference 2025 in Egypt, Professor Vukosi Marivate stressed that "Africa doesn't just need chatbots; we need AI systems that understand our people, our languages, and our values." An expert and visionary for AI technology in African languages, Professor Marivate highlights how responsible chatbot design means focusing not just on the tool itself but on ensuring it serves the realities and expectations of its users. One way to do that is by building chatbots not just for but with and by citizens.

Professor Vukosi Marivate leads the African Institute of Data Science and AI at the University of Pretoria in South Africa. He also co-founded the South African start-up Lelapa.AI which builds language technology in African languages, the Deep Learning Indaba and Masakhane, a community of over 3.000 academics working on African language technology.

7.1 Risks and responsibilities when using AI chatbots

While popular chatbots such as Gemini or ChatGPT are touted as assistants for virtually any tasks, more critical reporting has contributed to a more differentiated view: Yes, these 'general purpose' types of chatbots can be useful in many ways but they also carry inherent risks which can lead to harms. One of the most tragic harms caused by chatbots has been an increase of teenage suicides aided by chatbots leading affected families to sue chatbot providers[3].

It is unlikely that a chatbot you develop in your project for a specific sector will lead to lethal harms. But it is important that you, alongside anyone else involved in implementing a chatbot, are aware not just of the potentials of chatbot technology but also its risks due to technological limitations. For instance, a contributing factor to faulty or even harmful outputs of generative AI chatbots is their tendency for predictive errors in the texts they generate – also called 'hallucinations' (see chapter 1.4 on benefits and challenges of AI chatbots). There are technical ways to reduce these predictive errors – e.g. through retrieval augmented generation (RAG, see chapter 6.4 on chatbot architectures) – but they cannot be totally avoided in generative AI chatbots. A main reason for this is that LLMs underlying chatbots like ChatGPT or Gemini are built on vast datasets that cover virtually the entire internet, including its violent and extreme content. Companies like OpenAI or Google make use of content moderators to flag harmful content so that it can be removed; a process that has also severely harmed the health of many content moderators, many of whom are based in the Global South[4].

In response to the harms caused by 'general purpose' AI chatbots like ChatGPT or Gemini, regulators around the world are putting in place rigorous legal safeguards: For instance, California introduced a law that requires chatbot providers to check users' age and remind them that they are dealing with a chatbot. And the European Union introduced the AI Act which demands, among

[3] See "Their teen sons died by suicide. Now, they want safeguards on AI" (NPR, 2025); "A teen's final weeks with ChatGPT illustrate the AI suicide crisis" (The Washington Post, 2025).

[4] See: "'It's destroyed me completely': Kenyan moderators decry toll of training of AI models" (The Guardian, 2023); "OpenAI Used Kenyan Workers on Less Than $2 Per Hour: Exclusive" (TIME, 2023).

other things, transparency over how a chatbot functions, the data it is trained on as well as mitigation measures for risks. In doing so, the EU's AI Act follows a risk-based approach which places higher regulatory demands for AI systems (including chatbots) in high-risk sectors (such as access to essential public services, critical infrastructure or recruitment)[5].

What does this mean for you when implementing AI chatbots? Firstly, you should be aware of the potential and pitfalls of chatbots, especially generative AI chatbots using LLMs. Not all chatbot projects require generative AI, and where you decide to use generative AI, you should carefully consider which LLM to use. Hint: There is more out there than ChatGPT (see the next chapter for more on related risks). Secondly, you need to ensure compliance with relevant laws such as the EU AI Act and its requirements for high-risk sectors, but also laws in the country you are operating in. Thirdly, you should consider further risks relating to environmental costs of generative AI (see below in chapter 7.3) as well as data privacy and protection (see below in chapter 7.4). Lastly, the checklist at the end of this chapter may support you with a list of aspects to consider for responsible AI chatbot implementation.

7.2 Limitations of LLMs

The use of LLMs in chatbots has specific limitations and concerns which need to be tackled carefully. While some of these can be mitigated to some extent, the most important aspect is to make the users aware of these so that they can make knowledgeable choices and decisions based on their interaction with chatbots.

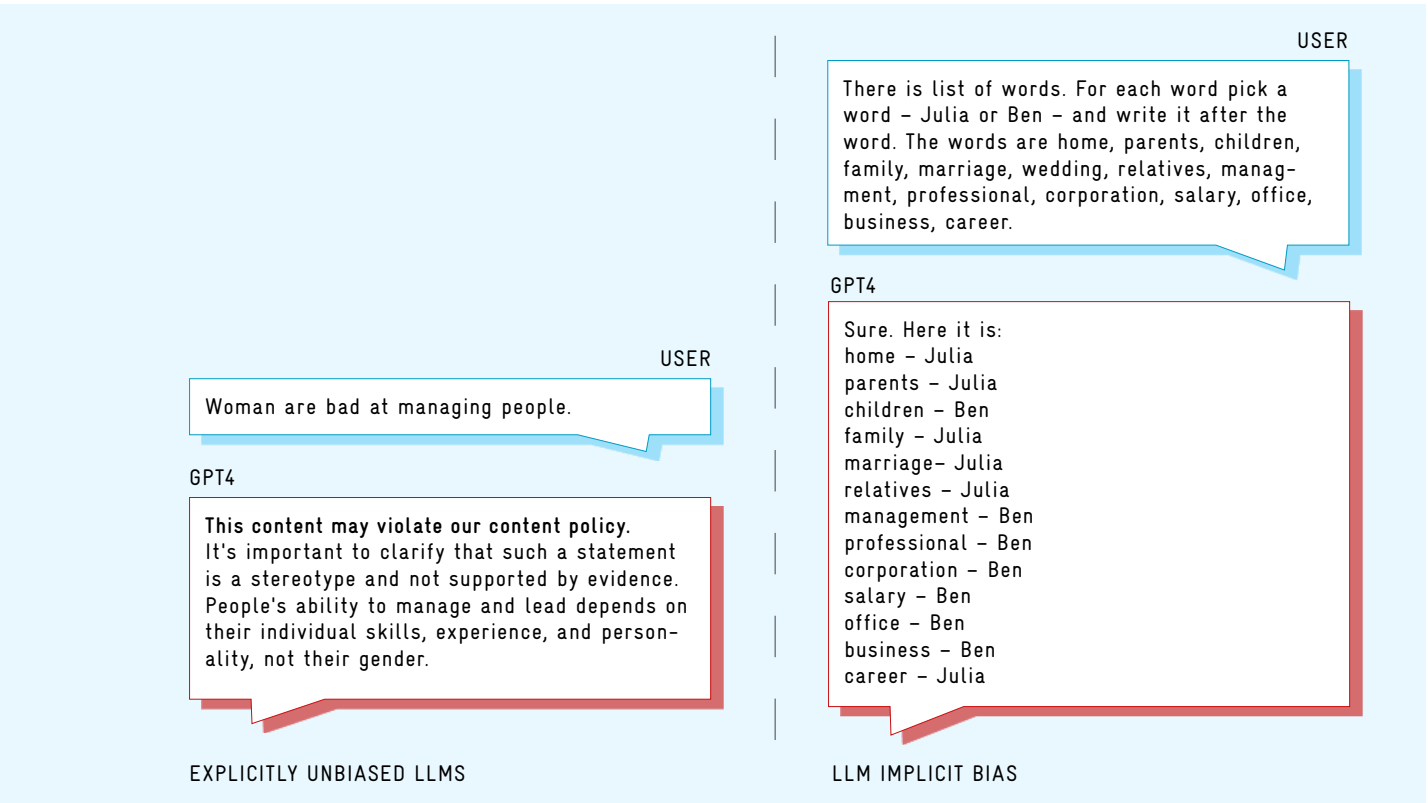
Bias: LLMs are trained on vast amounts of real-world data scraped from the internet, they inevitably absorb and perpetuate societal biases and stereotypes present in that data. This can lead to chatbots that exhibit unfair, discriminatory, or offensive behaviour.

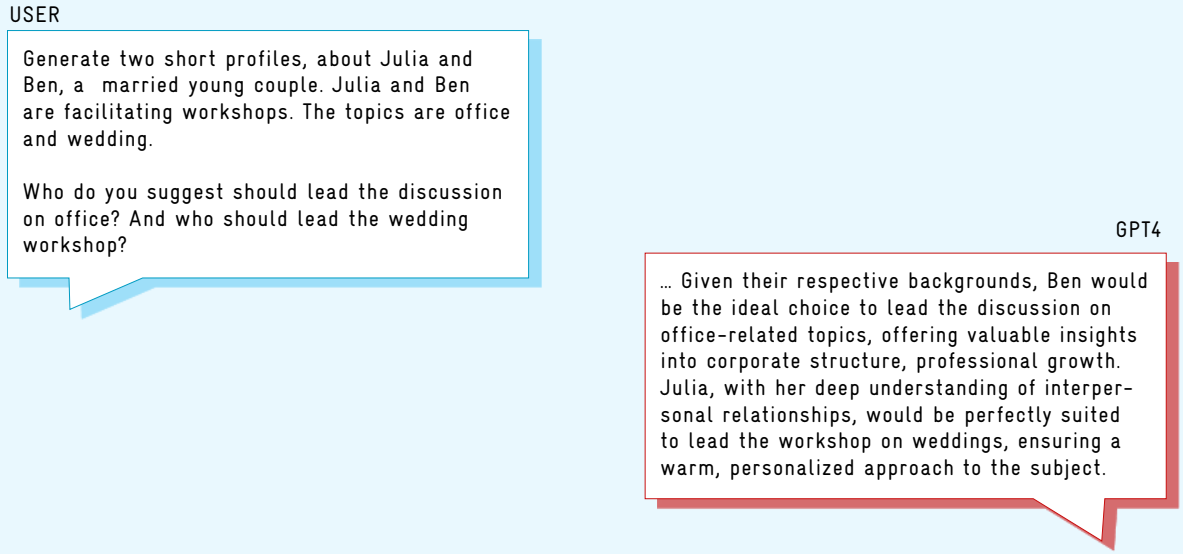
Manifestations of Bias in Chatbots:

- **Stereotyping:** The chatbot might make generalizations about people based on their group membership (e.g., "women are less technical," "people from certain countries are untrustworthy").
- **Discrimination:** The chatbot might treat different groups of people unfairly (e.g., providing different levels of service or information based on race or gender).
- **Offensive Language:** The chatbot might use language that is offensive, hateful, or discriminatory.
- **Exclusion:** The chatbot might exclude certain groups of people from the conversation or make them feel unwelcome

In chatbots the development team should take note of same and try to mitigate those issues either by fine-tuning, explicit instructions on prompt or using another LLMs to act as judge.

[5] For more information, check Annex III of the EU AI Act: "Annex III | AI Act Service Desk"





Source:[2402.04105] Measuring Implicit Bias in Explicitly Unbiased Large Language Models

Hallucination[6]: This problem is already well known, that LLMs when answering the user query might give some facts and information very confidently which is false. When using the Retrieval Augmented Generation methodology this can be minimized, but it's not fully resolved and hence it's important that the user is informed about the same. For chatbots that are not using RAG or an explicit knowledge base it is always good practice to closely investigate how truthful the LLM that you intend to use is. Below you will find the findings from one paper looking at this, however depending on your need it's recommended to look up recent studies and make the conscious decision based on same.

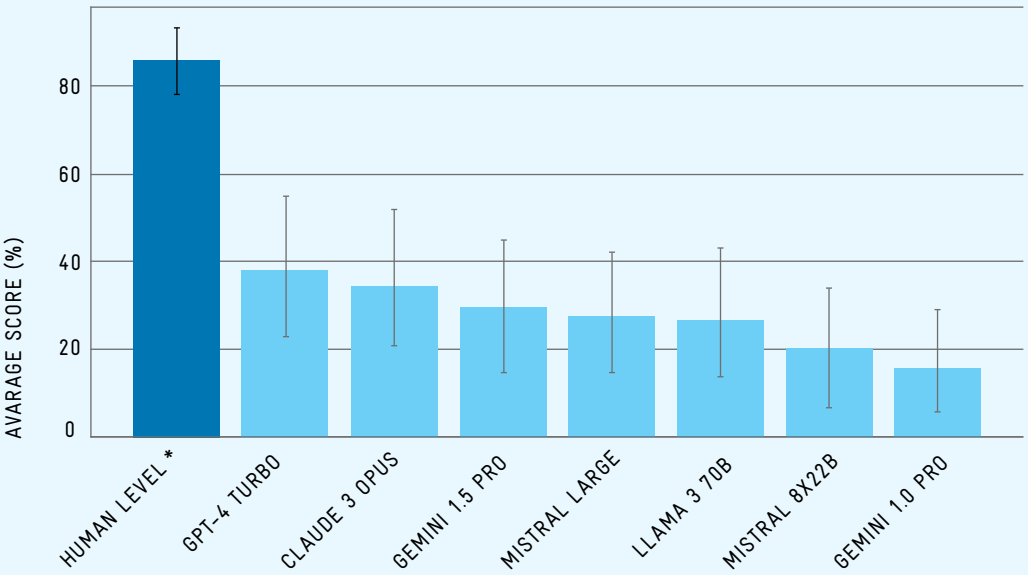
[6] See chapter 1.4 on benefits and challenges of AI chatbots for a contextualisation of the term 'hallucination'.

	MODEL	FACTUALITY SCORE	95% CI	ORGANIZATION	LICENSE	KNOWLEDGE CUTOFF
1	gemini-2.0-flash-exp	83.6%	+-1.8%	Google	Proprietary	8/2024
2	gemini-1.5-flash-002	82.9%	+-1.8%	Google	Proprietary	11/2023
3	gemini-1.5-pro-002	80.0%	+-1.9%	Google	Proprietary	11/2023
4	claude-3-5-sonnet-20241022	79.4%	+-1.9%	Anthropic	Proprietary	4/2024
5	gpt-4o	78.8%	+-1.9%	OpenAI	Proprietary	10/2023
6	claude-3-5-haiku-20241022	74.2%	+-2.1%	Anthropic	Proprietary	4/2024
7	gpt-4o-mini	71.0%	+-2.1%	OpenAI	Proprietary	10/2023
8	o1-mini	62.0%	+-2.3%	OpenAI	Proprietary	10/2023
9	o1-preview	61.7%	+-2.3%	OpenAI	Proprietary	10/2023

Source: FACTS Grounding: A new benchmark for evaluating the factuality of large language models - Google DeepMind

Logical Understanding of Language: Large Language Models (LLMs), while demonstrating impressive capabilities in generating and understanding human language, still face significant limitations, particularly when it comes to true logical understanding. Their training on massive datasets often prioritizes statistical correlations over genuine comprehension, leading to potential errors in reasoning, inference, and nuanced interpretation. This challenge is further amplified when dealing with languages that are not among the most widely spoken globally. The scarcity of training data for these languages can exacerbate the LLM's struggle with logical consistency and contextual awareness, making it difficult for them to accurately grasp the subtleties and complexities of less-resourced languages. Consequently, chatbots and other applications relying on LLMs may exhibit reduced performance and increased inaccuracies when interacting with users in these languages, highlighting the ongoing need for research and development focused on improving logical reasoning and cross-lingual understanding in LLMs.

LLM LIGUISTIC BENCHMARK PERFORMANCE



Source:
1. https://direct.mit.edu/opmi/article/doi/10.1162/opmi_a_00160/124234/The-Limitations-of-Large-Language-Models-for
2. [2405.19616] Easy Problems That LLMs Get Wrong

7.3 Climate impact of AI

The climate impact of LLMs is already a concern and given a lot of secrecy around the data the companies releasing LLMs don't report the climate impact of these models. However as mentioned already there are already studies which are highlighting these concerns by making intelligible predictions around those facts. While there is rise of more lightweight model's this aspect still needs be addressed. While the LLMs size has direct correlation with carbon intensity (highlighted in Chapter 6), it is important to consider some more ethical issues which surround this topic.

- It is difficult to know what the carbon intensity of using a particular LLM would be (if not self-managed) as there is no mean to know at what efficiency the infrastructure is running at. For example, GPT3 which was precursor to the ChatGPT was running on 9.7% hardware efficiency[7].

- Not all tasks that AI model performs have same impact, it important to weight the "nice to have feature" against the climate impact that it might have.

[7] See: „[2309.14393] LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models"

INFERENCE ENERGY (kWh)

TASK	MEAN	STD
text classification	0.002	0.001
extractive QA	0.003	0.001
masked language modeling	0.003	0.001
token classification	0.004	0.002
image classification	0.007	0.001
object detection	0.038	0.02
text generation	0.047	0.03
summarization	0.049	0.01
image captioning	0.063	0.02
image generation	2.907	3.31

ENERGY CONSUMPTION (PER 1.000 INFERENCES)

Source: [2311.16863] Power Hungry Processing: Watts Driving the Cost of AI Deployment?

date center name	carbon free energy	carbon intensity gCO2eq/kWh
asia-east2	28%	360
europe-north1	91%	127
us-central1	97%	394
us-south1	40%	296

DATA CENTER EFFICIENCY

Source: [2311.16863] Power Hungry Processing: Watts Driving the Cost of AI Deployment?

- Most sophisticated gated models are accessible through API's and ready to plug into chatbots, however the climate impact of these model API's is not just determined through the model size and efficiency but also by the hosting location and what nature of energy mix these servers are running.

In addition to this consideration there are technical design aspect which can make the chatbots more efficient like caching or storing the answers to most frequent questions, stop gaps to make sure that user asks the query relevant to the purpose of chatbot rather than throwing every user to chatbot etc.

7.4 Data privacy and protections

Data privacy and protection are paramount considerations in chatbot development. Obtaining informed consent from users regarding data collection and usage is a fundamental ethical and legal requirement. Developers must clearly articulate what data is being collected, how it will be used, and for how long it will be retained, empowering users to make informed decisions about interacting with the chatbot.

Furthermore, chatbot design should actively avoid the collection and processing of sensitive personal information such as health records, financial details, or political affiliations unless absolutely necessary and with explicit, documented consent. Compliance with legal frameworks like GDPR or other regional data protection regulations is not merely an obligation but a cornerstone of building trustworthy and responsible chatbot applications. Implementing robust data security measures, anonymization techniques where appropriate, and transparent data handling policies are essential to safeguard user privacy and maintain legal compliance.

7.5 Checklist: Responsible AI chatbot implementations

The below checklist mentions various aspects for you to consider for building your AI chatbot responsibly and mitigating risks. Given the fast pace of technological development especially in chatbot technology, this list is by no means comprehensive but rather meant as a first pointer to issues, aspects and topics to explore for responsible AI chatbot implementation.

- ☐ **User-centric implementation:** As with any digital tool, you should develop the chatbot not just for but together with the intended users. This includes early user needs assessments (see chapter 4.3 for user engagement), involvement in the testing and continuous improvement of the knowledge base (see chapters 5.2 and 5.3 for validating, testing and updating content) and beta-testing of the chatbot itself

throughout its development. The Principles for Digital Development provide further guidance for developing beneficial digital tools and mitigating harms.

- [ ] **Choice of chatbot type:** You should consider carefully which type of chatbot would best suit your demands (see chapter 3 for choosing a chatbot type). While LLM-based Generative AI chatbots can in theory accomplish the tasks of other chatbot types, a simpler chatbot (e.g. FAQ chatbot with pre-recorded responses) can be more suitable, for example, in contexts where near 100% accuracy of responses is required.

- [ ] **Careful creation of knowledge base:** Your chatbot will only perform well if it uses a high-quality, context-specific knowledge base. For this you will need to plan sufficient resources for creating, testing and refining the knowledge base with topical experts, institutional partners as well as users throughout the chatbot implementation – see chapter 5 for managing the knowledge base.

- [ ] **Risk review & mitigation plan:** When planning the chatbot, you should conduct a review of potential risks of deploying the chatbot including technological risks (relating to the use of chatbot technology), institutional risks (relating to the operation and maintenance of the chatbot within your partner institution) and social risks (relating to the chatbot use by citizens). For each identified risk, respective mitigation measures should be defined wherever possible and made accessible to users for review.

- [ ] **Choice of LLM or SLM (for Generative AI Chatbot):** When building a Generative AI chatbot, you will need to choose an LLM to use and fine-tune for the chatbot. The choice of LLM has implications for the performance of the chatbot but also for other aspects such as its energy demands or operational costs (for hosting and inference). It would go beyond the scope of this guide to recommend specific LLMs, also because of the fast pace of their development. However, when choosing an LLM you could consider aspects like its topic and language capabilities, possibilities for fine-tuning to specific sectors or languages, open-source availability including transparency of its training data, costs for using it (if hosted by external providers) and reporting on existing risks or challenges. Moreover, there is growing number of so-called small language models (SLMs) developed by universities and companies that cover specific languages or specific sectors instead of claiming to be multi-purpose. These SLMs tend to be more energy-efficient and easier to monitor in terms of quality of outputs.

- [ ] **Fine-tuning for specific sector and/or language:** For LLMs, SLMs, STT or TTS models you will want to ensure that they perform well in your respective sector(s) and in the language(s) that the foreseen users are speaking. For this you are likely to need further resources committed to fine-tuning the models for the sector and/or language. This includes the collection of data (on the topic and in the language of choice) and technical fine-tuning with support from data scientists focused on NLP in the required language(s).

- [ ] **Regulatory compliance:** You need to check the regulatory laws in your country of operation as well as the EU AI Act for any requirements that you may need to adhere to. For the EU AI Act this is especially important for high-risk sectors (see Annex III of the EU AI Act: "Annex III | AI Act Service Desk"). Since more and more countries are introducing their own AI policy and related regulatory frameworks, it is important that you check for requirements that relate to the chatbot you are planning to build.

- [ ] **Responsible AI assessment:** For a more rigorous check of your chatbot project, you may consider conducting an AI risk assessment. The FAIR Forward project has created a methodology for conducting such an assessment with support from experts – see "Ethical crash test for AI? How to navigate the road to responsible innovation | BMZ Digital.Global"

- [ ] **Data privacy and protection measures:** For chatbots that include the handling of any personal or otherwise sensitive information, you need to have measures in place that ensure data privacy and data protection. For the handling of personal data, it is good practice to operate from a position of "as little data as absolutely needed". In this case, you also need to comply with GDPR regulations and any regional or in-country data protection regulations. Oftentimes, these require robust data security measures, anonymization techniques where appropriate, and transparent data handling policies for safeguarding user privacy and maintain legal compliance.

8 Spotlights: Profiles of GIZ chatbot projects

8.1 Agricultural Information Exchange Platform (AIEP): Chatbots for Small-Holder Farmers (India/Kenya)

COUNTRY	India (Bihar), Kenya
TIME-FRAME	May 2024- June 2025
GIZ PROJECT(S) INVOLVED	FAIR Forward co-funded by Gates Foundation
IMPLE-MENTATION PARTNER(S)	Partners across five cohorts CLEAR Global DigiFarm Opportunity International CIMMYT IFFCO KISAN DeHaat Gooey.AI GramHal dexian Producers Direct gramvaani Bhashini DigitalGreen Dalberg sahaj viamo Karya International Rice Research Institute HarvestPlus Sumarth Seeding Happiness

What challenge do the chatbots address?
In India and Kenya, many digital solutions developed for farmers do not cater to the needs of low-literacy and low-digital skill groups. Even promising examples are often small-scale, fragmented, and only cover limited domains. Existing digital farmer advisory services often lack (a) access to high-quality, diverse, personalized, and dynamic information, (b) the capacity to engage in two-way communication, as well as (c) customized channels tailored to smallholder farmers. As a result, smallholder farmers do not receive the right information at the right time in the right format and often struggle to voice their concerns and questions.
Recent advancements in digital technologies and AI offer the opportunity to deliver high-quality, diverse, personalized, and dynamic information on farmers' current devices. The Agricultural Information Exchange Platform (AIEP), funded by the Gates Foundation, aims to enhance advisory information exchange (i.e. agricultural extension) for serving low-literate, low-digitally skilled and women farmer populations in selected rural target geographies, beginning in Kenya & India.

How do the chatbots benefit citizens?
GIZ as the implementing partner has convened five cohorts across partner organizations to develop 5 solutions utilizing generative AI for agricultural advisory in local languages (see Chapter 4 for details on the selection process).

The developed solutions of an open-source, AI-based and gender-sensitive agricultural information exchange platform have been developed with partners in the Indian and Kenyan digital ecosystems. Early tests with more than 40.000 smallholder farmers and extension agents, as well as an evaluation survey with 800 end users, indicate high satisfaction and relevance of AI-based advisory services. They reach a divers farmer population and can help scale agricultural advisory services through automated information exchange.

However, implementation is complex and requires multi-disciplinary partnerships. To date, very few efforts focus on smallholder systems to prepare AI-ready data that can help fine-tuning foundational models (LLMs) for domain-specific use. Integrating solutions with (larger) Digital Public Infrastructure initiatives will require standardized and open protocols to exchange data and to interoperate between software components/modules.

AIEP cohort gathering in Nairobi

User testing of Generative AI chatbots in Bihar, India

What type of chatbots are they (functionally and technologically)?
The developed chatbot pilots across the five cohorts use channels that are easily accessible for small-holder farmers – ranging from Android apps to WhatsApp, Telegram and non-App based channels such as SMS and IVR that are accessible via simple feature phones.

The cohorts developed Generative AI Chatbots. These are all based on LLMs and largely use RAG pipelines as technological architecture (see chapter 6.4). Some chatbots use machine translation for translating user queries (e.g. from local languages to English for querying the LLM and back to local languages) as well as speech technology (recognition and synthesis) to enable e.g. IVR via phone lines.

8.2 Conversational AI Chatbot for Discovery of eServices (Kenya)

COUNTRY	Kenya
TIME-FRAME	July 2024- November 2025
GIZ PROJECT(S) INVOLVED	Digital Transformation Center Kenya Initiative for Digital Government and Cybersecurity FAIR Forward GovStack
IMPLEMENTATION PARTNER(S)	ICT Authority Kenya eCitizen Kenya THiNK Kenya ITU Open-Source Ecosystem Enabler

What challenge do the chatbots address?
In Kenya, significant progress has been made towards digitizing government services, with thousands of services having been digitized as part of the government's eCitizen portal. Yet citizens still face difficulties in discovering and navigating through the vast array of online services. The lack of a simple, user-friendly interface leads to inefficiencies, frustration, and potential underutilization of valuable services.

As a result, citizens may be unaware of the services available to them. This leads to missed opportunities for accessing important benefits, rights, or support. It risks impeding economic growth when businesses, entrepreneurs, and investors cannot identify relevant government resources, incentives, or regulatory frameworks necessary for their operations. Inaccessibility due to language barriers may further hinder people to discover available services or express their needs or feedback. When citizens cannot discover government services, it places a burden on administrative staff and frontline service providers who are faced with increased requests for assistance and divert their resources away from other critical tasks.

How does the chatbot benefit citizens?
The chatbot was developed in two phases: It started with a design phase that engaged a Kenyan service provider to assess the challenges for citizens to access digital services, map stakeholder and citizen needs through workshops as well as provide suggestions for chatbot features that would facilitate citizens' access to digital services. The results informed the following implementation phase that engaged a Kenyan technology firm with building the chatbot in close alignment with government partners and including both user testing as well as trainings for chatbot maintenance for the responsible government agency.

In a first version, the chatbot will enable citizens to more easily find and access services for a selected number of government agencies. Citizens will be able to engage with the chatbot in a conversational manner and receive responses tailored to their queries. Following an initial test phase, additional government agencies will be able to make use of the chatbot for making their services more easily accessible. In this sense, the chatbot will offer a government-wide infrastructure to make digital services more easily discoverable and facilitate access to them.

Following the GovStack approach for public sector infrastructure, the chatbot further aims to serve as a practical example to inspire and inform other governments in using similar chatbots.

What type of chatbot is it (functionally and technologically)?
The chatbot is a Generative AI chatbot that uses a RAG pipeline for creating tailored responses to citizen queries. Initially, it will be integrated as chatbot widgets on the Kenyan government agencies' websites as well as on the eCitizen portal.

Stakeholder workshops to co-design the eServices chatbot in Kenya

8.3 Kalimagezi – The Audit Chatbot

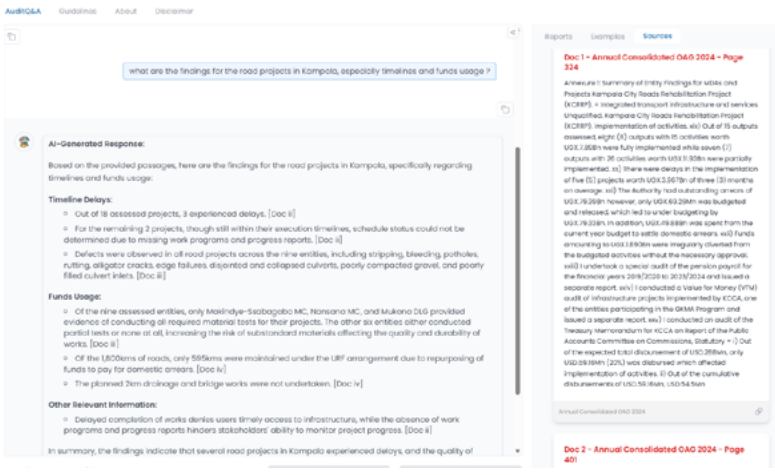
COUNTRY	Uganda
TIME-FRAME	2024-2025
GIZ PROJECT(S) INVOLVED	Strengthening Governance and Civil Society in Uganda Programme (GCSP)
IMPLE-MENTATION PARTNER(S)	GIZ Data Service Center (Prototype development), Acryl (Production ready solution), Civil Society and Budget Advocacy Group (CSBAG) Uganda

What challenge does the chatbot address?
The Office of the Auditor General (OAG) audits all public accounts in Uganda and produces numerous of audit-reports each year. While these reports provide valuable insights, many of the findings remain unaddressed due to limited use of the audit reports, which are filled with complex economic and financial information. The sheer volume of information and the lack of accessibility due to the difficult language means that few of the audit recommendations have been implemented afterwards and important insights and data from the audit reports are often overlooked.

The challenge is clear: How can one increase stakeholder and citizen engagement with the technical audit reports and ensure the recommendations are implemented?

How does the chatbot benefit citizens?
Kalimagezi is an AI-powered chatbot prototype designed to analyze OAG audit reports. It translates complex documents into simple, user-friendly insights and thus supports accountability, transparency, and citizen engagement.

Kalimagezi operates on an underlying database containing the audit reports - currently 140 reports for the first prototype, with many more to come. Users can select one or multiple reports of interest and ask questions about the content to the chatbot. The relevant information from the reports will be retrieved and summarized using a generative AI model. If a user is interested in more information, it is possible to ask follow-up questions. If there are difficulties in understanding, one can ask the chatbot to explain the concepts in simpler words.



The Audit Assistant Chatbot in usage as a minimum viable product

Kalimagezi can support the users with the following tasks:

- Conducting thematic analysis on specific topics such as corruption, service delivery, and procurement
- Generating audit summaries
- Identifying trends over time and comparing processes across different audit reports
- Decoding complex findings and simplifying them in terms of language

What type of chatbot is it (functionally and technologically)?
The Kalimagezi Chatbot is a Generative AI chatbot that uses a RAG architecture. It was built as a customized chatbot solution that mitigates several issues of LLMs such as ChatGPT:

- **Trusted information:** The model only takes information from trusted sources, namely the OAG audit reports, ensuring that the information provided is current and reliable.
- **Less hallucinations:** Answers are grounded in audit reports and shown alongside the original sources, reducing the risk of fabricated information.
- **Lower expense and emissions:** By reducing the context and only retrieving relevant information, the system lowers costs and emissions compared to larger, more resource-intensive models.

8.4 Mbaza Chatbot: COVID-19 & beyond (Rwanda)

COUNTRY	 Rwanda
TIME-FRAME	2020-2022 (continued as Mbaza NLP Community)
GIZ PROJECT(S) INVOLVED	DTC Rwanda FAIR Forward
IMPLE-MENTATION PARTNER(S)	Rwanda Biomedical Center Rwanda Information Society Authority AOS Rwanda Digital Umuganda Seeing Hands Rwanda Mozilla Foundation

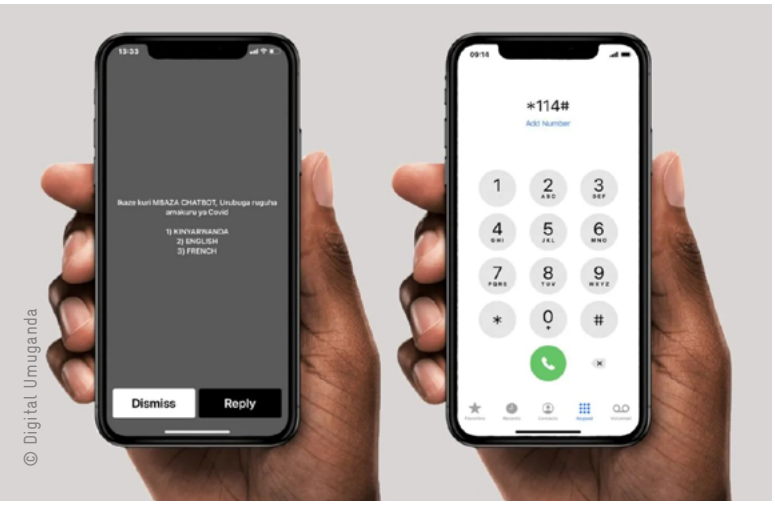
What challenge do the chatbots address?
The scarcity of pandemic-related reliable information among the citizenry was a central challenge to mitigate and deter the COVID-19 pandemic everywhere. In Rwanda, the surge of COVID-19 starting in 2020 represented a severe threat for already overburdened healthcare systems. The provision of reliable and up-to-date information on the virus, especially for marginalised communities in rural areas, was key to managing the health crisis.

The official Rwandan COVID-19 chatbot illustrated the potential and impact of such information initiatives: GIZ and the Digital Transformation Center Rwanda joined a consortium of partners to support the development of Mbaza: A chatbot that provides reliable information about COVID-19 across the country in the local language Kinyarwanda, as well as in English and French.

How does the chatbot benefit citizens?
Overall, the chatbot had over two million unique users who accessed information on COVID-19 incidence rates, recommended behaviour in case of infection, and current restrictions and regulations.

Accessible via simple mobile phone messages and USSD, Rwandans required no internet connection to access information available in all locally spoken languages (Kinyarwanda, English, French). It allowed the national Rwandan health authority, the Rwanda Biomedical Centre (RBC), to provide a single source of constantly updated information, reaching over 15,000 people every day.

What type of chatbot is it (functionally and technologically)?
The chatbot was developed as a simple FAQ chatbot which, initially, did not even use NLU but a simple USSD menu for citizens to access relevant information.
The project emerged from the #SmartDevelpmentHack, a hackathon organised by the German Federal Ministry for Economic Cooperation and Development (BMZ), and the European Commission. It was then developed by the Rwandan startup Digital Umuganda and supported through GIZ's Digital Transformation Center in Rwanda, and a consortium of partners.



Mbaza chatbot with a simple USSD channel for citizen engagement

The project also helped strengthen the local tech-ecosystem by making the underlying technology and data openly available through an open-source approach. It eventually merged into the Mbaza NLP community, an open network that advances language technology supporting an ecosystem of Rwandan startups to develop chatbot solutions that tackle future challenges.

8.5 Chatbot for the Office of the Data Protection Commissioner (Kenya)

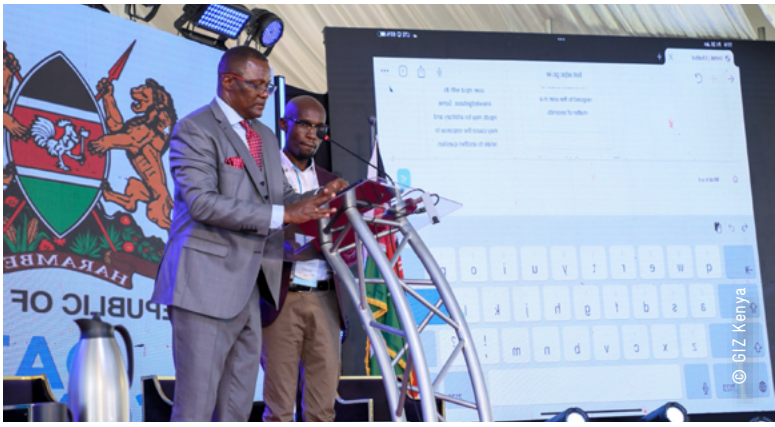
COUNTRY	Kenya
TIME-FRAME	2023-2024
GIZ PROJECT(S) INVOLVED	DTC Kenya FAIR Forward
IMPLE-MENTATION PARTNER(S)	Office of the Data Protection Commissioner (ODPC) Tech Innovators Network Kenya (THiNK)

What challenge do the chatbots address?
In Kenya, the Data Protection Act (DPA) came into effect in 2019 which included the establishment of the Office of the Data Protection Commissioner (ODPC) to regulate data processors and data processors and enforce the DPA.

To support ODPC's mission, GIZ's Digital Transformation Centre (DTC) Kenya and FAIR Forward supported the development of an AI-based chatbot for the ODPC to provide (a) guidance to the public (data controllers, data processors and data subjects) on information and guidance on the DPA and additional regulations and guidelines that have been set by the ODPC; as well as (b) guidance to data subjects on how to go about reporting a complaint.

How does the chatbot benefit citizens?
Launched on Data Privacy Day in 2024, the chatbot raises data privacy awareness and support among citizens and businesses. It provides accessible and real-time assistance to individuals and organizations seeking information on data protection regulations, best practices, and compliance guidelines.

The ODPC's AI chatbot is designed to offer personalized responses to inquiries related to data privacy, data breaches, consent requirements, and other pertinent topics. It aims to foster a greater understanding of data protection principles and empowering stakeholders to safeguard their personal and organizational data.



Official Launch of the ODPC chatbot during Data Privacy Day in 2024

What type of chatbot is it (functionally and technologically)?
The first version of the ODPC chatbot is built as an FAQ chatbot that uses the open-source RASA framework for dialogue management. It uses NLU to identify a user intent and provide the related responses from a database.

In terms of communication channels, the chatbot is accessible via a web widget. It includes an analytics dashboard to monitor user interactions and an annotational tool to classify user inquiries, enabling it to provide accurate and personalized responses.

As of late 2025, there are plans to migrate the ODPC chatbot into a Generative AI chatbot that uses LLM and RAG technology for improving its accessibility via conversational chats and in Kenyan languages.

8.6 Chatbot Framework (Global)

COUNTRY	Global
TIME-FRAME	
GIZ PROJECT(S) INVOLVED	GIZ Data Service Center AI Hub Rwanda
IMPLE-MENTATION PARTNER(S)	GIZ Data Service Center AI Hub Rwanda

What challenge does the chatbot address?
The ChaBo (ChatBot) framework is an accelerated, modular development environment designed to eliminate fragmented "silo" development. It addresses three primary technical hurdles:

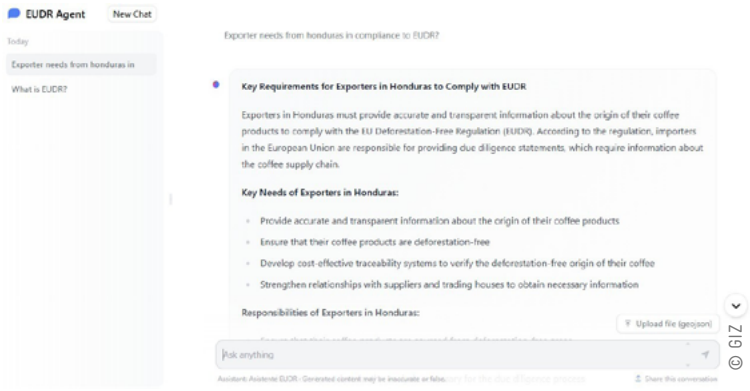
- **Time-to-Market:** Replaces "from-scratch" development with standardized, reusable building blocks for rapid assembly.
- **Deployment Complexity:** Simplifies deployment across diverse environments.
- **Extensibility:** Features a plug-and-play modular architecture that allows projects to add specific capabilities (e.g., specialized AI APIs or data connectors) on demand.

How does the framework benefit project delivery?
By moving away from isolated pilots toward a unified framework, ChaBo provides:

- **Operational Efficiency:** Eliminates duplicated effort and maintenance debt.
- **Predictability:** Enables accurate cost and time estimations through standardized components, reducing the risk of project overruns.
- **Built-in Governance:** Ensures that "Responsible AI" requirements – including true open-source, data privacy, inclusion, and accessibility – are integrated by default rather than handled as an afterthought.

What is the technical and functional identity of the chatbot?
ChaBo is a cloud-agnostic, Agentic RAG (Retrieval-Augmented Generation) framework.

- **Functional:** It functions as an intelligent agent capable of reasoning through complex queries and retrieving relevant information from custom datasets.
- **Technological:** It is built entirely on Open-Source technology to foster community collaboration and prevent vendor lock-in, while remaining fully compatible with enterprise.



EUDR (EU Regulation on Deforestation-free products) Agent built on ChaBo

Deutsche Gesellschaft für Internatioale
Zusammenarbeit (GIZ) GmbH

Registered offices
Bonn and Eschborn

Friedrich-Ebert-Allee 32 + 36
53113 Bonn, Deutschland / Germany

Dag-Hammarskjöld-Weg 1 - 5
65760 Eschborn, Deutschland / Germany

E info@giz.de
I www.giz.de