DeepL

Subscribe to DeepL Pro to translate larger documents.
Visit www.DeepL.com/pro for more information.

Conference of Independent Data Protection Authorities

of the Federal Government and the States

Guidance on the specific data protection issues of generative AI systems using the RAG method

Version 1.0

Status:
October 2025

Contents

1. Introduction 3

2. Definition of the RAG system 5

2.1. Detailed description of a RAG system 7

2.2. Limitations of the RAG method 8

3. Generative language models with and without RAG and implications for data protection assessment 9

3.1. Accuracy 9

3.2. Transparency 10

3.3. Integrity and confidentiality 11

3.4. Purpose limitation 12

3.5. Data minimization and storage limitation 13

3.6. Lawfulness 13

3.7. Handling of data subject rights 14

4. Conclusion 14

5. Glossary of key terms 15

Abstract[1]

This paper examines the feasibility of implementing the principles of the GDPR in AI systems with retrieval augmented generation (RAG), known as RAG systems. Based on a typical application scenario, the technical components of an RAG system are presented. It shows that the RAG method can have positive effects on the accuracy and traceability of an AI system's output, for example in terms of ensuring context-related and verifiable content, and that the confidentiality and integrity of additionally integrated personal data can be improved in an RAG system. At the same time, new data protection challenges arise. Furthermore, in many cases, the method makes it possible to use an LLM on-premise, which requires less extensive training data, meaning that less personal data can be extracted from the AI model. Although the data protection assessment of the training of the LLM used as such remains unaffected, the implementation of the RAG method in an AI system can ultimately lead to a differentiated data protection assessment.

1. Introduction

This paper addresses specific issues related to RAG systems, focusing on systems with embeddings and vector databases. Such systems have established themselves as the technical basis for many RAG systems, as they enable semantics-oriented searches in large knowledge bases. Other technical implementations of RAG systems, such as those based on API queries or classic relational database access, are not covered in this paper. In particular, a RAG system does not change how a generative AI model was trained. An unlawfully trained AI model remains an unlawfully trained AI model even in a RAG system, and its lawful use is subject to high requirements (see EDSA Opinion 28/2024, in particular Scenario 2, para. 124 ff.).

At the same time, it should be noted that both LLM and RAG can cover numerous use cases, and the statements in this paper must therefore be formulated in general terms. The data protection assessment of the specific application must be carried out on a case-by-case basis.

The focus below is on the specific implications of using the RAG system on the data protection assessment of a generative AI system based on an LLM. The data protection assessment of the RAG subsystem is the decisive basis for this. Unless otherwise stated, a use case scenario is assumed in which the entire RAG system is operated on-premises by a controller. Further aspects may arise from the distribution of system components to different operators in the form of order processing or joint responsibility.

[1] A glossary of key terms can be found at the end of the document.

To date, AI systems that use a large language model (LLM) such as generative pre-trained transformers (GPT) as their AI model have attracted particular attention. This technology owes its popularity in part to the fact that it enables natural language texts to be analyzed and syntactically correct texts to be generated. However, the use of LLMs poses numerous risks to data protection. These arise, among other things, from training with data that typically includes personal data and special categories of data, as well as from the phenomenon of hallucination, which compromises the accuracy of the (possibly personal) output. In addition, LLMs are usually trained with data up to a certain date. Input prompts that contain questions about events that took place after this date, for example, therefore regularly lead to incorrect answers ( ). Attacks on a language model must also be taken into account,([2]) e.g., membership inference attacks or the extraction of training data from the AI model (model inversion attacks). In addition, those responsible must ensure, among other things, that the rights of the data subjects are protected in accordance with Chapter 3 of the GDPR, e.g., the right to rectification under Art. 16 GDPR and the right to erasure under Art. 17 GDPR. This paper describes how a RAG system can affect the data protection-compliant use of a generative AI system.

Several approaches are possible for optimizing an LLM for a specific application context, i.e., to utilize knowledge from a domain that was previously unknown or incomplete for the ·One approach is to use retraining or fine-tuning of the language model; another is to apply the RAG method. Put simply, the RAG method combines an LLM with a database to make information that is (only) available to the responsible party and was not part of the LLM's training data accessible and usable via an AI system.

The extent to which an RAG system influences the data protection assessment of a generative AI system based on an LLM must be examined, particularly with regard to the issues addressed in EDSA Opinion 28/2024. These relate in particular to the legal basis for training with personal data from web scraping, reuse in the event of unlawful training, and the exercise of the aforementioned data subject rights.

This paper analyzes the influence of the application of the RAG method on the data protection assessment of generative AI systems and identifies additional challenges, but also simplifications, associated with the extension of the language model with an RAG

---

[2] See Rigaki/Garcia, A Survey of Privacy Attacks in Machine Learning, 2020.

[3] See DSK, OH Artificial Intelligence and Data Protection, Version 1.0, May 2024, in particular margin note 26 ff.

[4] Lewis et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, arXiv:2005.11401v4, April 12, 2021; Daniel Jurafsky and James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition, January 12, 2025, in particular Chapter 14, https://web.stanford.edu/~jurafsky/slp3; Yunfan Gao et al., Retrieval-Augmented Generation for Large Language Models: A Survey, especially Chapter II.D, https://arxiv.org/pdf/2312.10997.

subsystem. The RAG method can be used in various contexts, such as natural language, multimodal data formats, or in combination with non-generative AI (e.g., in classification or recommendation systems). Only RAG systems with generative language models are considered here as examples.

2. Definition of RAG system[5]

The following section uses a selected scenario to illustrate the basic functionality of a RAG system. In this scenario, the reference documents also contain personal data that should be accessible via the RAG system. Different possible variants of the scenario are only discussed where they are helpful for a basic understanding.

In RAG systems, user input (input prompt) is supplemented by the RAG subsystem with text from reference documents before being sent to an LLM. The aim is to ensure that the generative AI system, such as an LLM, incorporates this relevant information and focuses on it in the response to be generated. In the selected scenario, the factual knowledge of a response should come entirely from the reference documents, with the LLM serving only to generate the language. ([6])

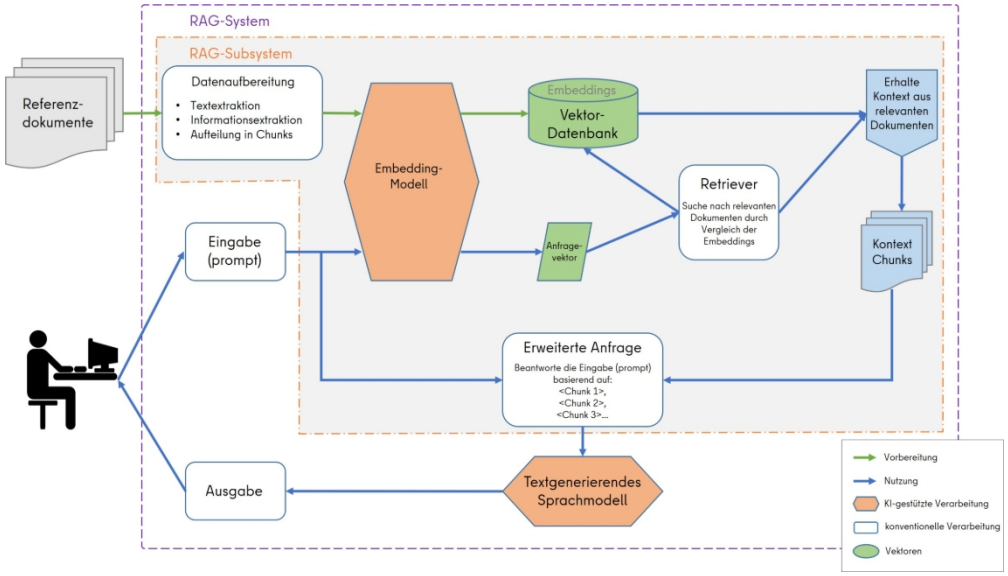

Fig. 1: Answering questions with a RAG system

As shown in **Fig. 1**, the RAG method supplements the LLM with a search function (retriever) and a vector database. In the scenario used here, before using the

[5] See also: Lewis et al. 2020, https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf; Gao et al. 2023, https://simg.baai.ac.cn/paperfile/25a43194-c74c-4cd3-b60f-0a1f27f8b8af.pdf; Chen et al. 2024, https://ojs.aaai.org/index.php/AAAI/article/view/29728.

[6] Currently, this ideal case is not always achieved technically. Even factual knowledge of the LLM can unintentionally become part of the output, which may require additional measures.

RAG systems process the reference documents. In addition to other steps, the process also involves dividing the reference documents into shorter paragraphs (chunks). These chunks were converted by the embedding model into vectors that represent the respective chunk. These vectors were stored together with the chunks in the vector database.

When using the RAG system, the input is also converted into a query vector using the same embedding model. The query vector is then passed on to the retriever. The retriever searches the vector database for vectors of the chunks that are as close as possible to the query vector in order to find chunks that are relevant to the query. A small distance is associated here with a high degree of semantic proximity.([7])

In the next step, the input is supplemented with the search results from the retriever. The generative AI then generates the output from the supplemented input. Depending on the design of the AI system, supplementary information, e.g., on the reference documents, is displayed in addition to the output. A RAG system as defined in this paper is thus a system that essentially consists of a retriever, an embedding model, a vector database, and a generative language model.

The retriever (search function) and the embedding model mentioned above are crucial for the typical behavior of a RAG system. The retriever and the embedding model provide a search function that can include AI algorithms such as transformers and vector-based search mechanisms. The search for relevant chunks is possible because the embedding model converts similar texts into vectors that are closer to each other than the vectors of texts that are less similar. This is made possible by conservative methods or by AI models which, like the LLM, are based on so-called transformer models and therefore, just like the LLM, can have the same data protection issues during training and application.

It is important to note that many RAG systems do not use a purely syntactic search, but rather a search for semantic proximity. This means that text passages that are relevant in terms of content are taken into account in the search, rather than hits that are similar in terms of words. The RAG system thus generates an output to a user's input based on additional relevant information from a dedicated data source. The retriever essentially serves to identify the relevant data in the processed data by means of a vector comparison.

2.1. Detailed description of a RAG system

The following section describes in more detail selected components and procedures that were briefly mentioned in the previous section.

[7] AI models approximate the semantics of natural language by analyzing very large amounts of data and indirectly deriving text meanings from it. There is no genuine understanding of the text content.

Data preparation

First, the data source, especially reference documents, is prepared. In our scenario, this is done by dividing reference documents into individual text sections (so-called chunks). This is an important step that influences the quality of the subsequent responses. Different approaches can be chosen, two of which are outlined below as examples.

In the simplest case, sections of equal size are generated, e.g., with a fixed number of characters. This approach can destroy semantic relationships in the reference texts and thus lead to qualitative disadvantages in the subsequent embedding. Another example is chunking with LLM. By using generative AI models, some of which are smaller, the reference documents can be divided based on their content and semantic meaning. This approach better preserves the context of the content within a chunk and thus also ensures better quality when creating the vectors. Since this type of division has a positive effect on subsequent searches (retrieval), the quality of the RAG system's output is also improved. As already mentioned in the introduction, the latter example must be evaluated in terms of training and application from a data protection perspective in the same way as other LLMs.

Embeddings

The embedding model converts the respective text passages (and also the subsequent input) into tokens or token sequences (e.g., words, character strings, characters) and then into a vector representation. It is crucial that the vectors generated by the embedding model represent the meaning (based on words from the training data that frequently occur in common contexts) of the respective information. These vectors can be stored in a database together with a link to the text passages or the text passages themselves. Here, too, there are a number of different approaches, the use of which depends on the concept and objectives of the RAG system.

Recognizing relevant data (retriever, RAG subsystem)

The retriever's task is to link the input with text passages that are as close as possible in terms of content. To do this, the embedding model also generates a vector from the input. The retriever compares this with the vectors in the database. The text passages with the most similar vectors (greatest proximity) are used to enrich the input. In this process, the proximity of the vectors to each other is understood as a measure of the relevance of the chunk.

The retriever's extended query cannot be compared to the training or retraining[8] of an LLM. The extended query is recreated from the data source each time. Unlike the training data, the extended query does not change the LLM. The information in the vector database of the present scenario can significantly influence the output of the LLM. In contrast to the information learned in the LLM, however, it can be easily updated, deleted, and disclosed, which facilitates the implementation of data protection requirements regarding personal data from the reference documents.

2.2. Limitations of the RAG method

The embedding of chunks and the enrichment of the LLM query with semantically related text passages can only directly influence the input prompt for an LLM. The trained knowledge of the LLM cannot be changed in this way (unlike with fine-tuning or retraining). However, the design and content of the expanded query do influence the output of the LLM. This is precisely the purpose of RAG systems: the LLM should only contribute its language capabilities in the selected scenario.

The idea of automated enrichment of the input prompt is subject to limitations. The input into an LLM cannot be arbitrarily long. This means that the text passages to which the input prompt is added are limited in length. The limit depends on the language model used. However, the actual length of the documents often no longer matters if they can be divided into meaningful chunks, making even very large documents manageable. Furthermore, enrichment only takes place in the semantically adjacent embedding space. Complex if-then chains of thought across long text passages may no longer be semantically adjacent, may not be contained in the same chunk, and are therefore added incompletely to the expanded query. This results in system-related weaknesses in the recognition of relevant information, which logically belong together but do not have sufficient semantic proximity.[9]

[8] It is not possible to say in general terms to what extent the quality of the results can be further improved by retraining (fine-tuning) the AI model. According to the Fraunhofer Institute IESE, "fine-tuning should not be used to train an LLM," but it can help to take a certain response style into account, pay attention to specific details, or better handle special search result formats (Fraunhofer IESE, May 13, 2024, https://www.iese.fraunhofer.de/blog/retrieval-augmented-generation-rag/). Another study observes that RAG and fine-tuning can independently lead to improved results (Balaguer et al., RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture, https://doi.org/10.48550/ arXiv.2401.08406).

[9] Attempts are being made to solve this problem using AI agent systems, which may send multiple requests to the retriever iteratively.

3. Generative language models with and without RAG and implications for data protection assessment

In line with the principles for the processing of personal data under Article 5 of the GDPR and the rights of data subjects under Chapter 3 of the GDPR, this section looks at generative AI systems with a language model and how the use of the RAG method affects the data protection assessment of the AI system. It should be noted that an RAG system does not change the properties of its LLM component, but merely influences and supplements it with additional context.

3.1. Accuracy

AI systems with an LLM must comply with the data protection principle of accuracy of personal data. Generative AI models reach their limits in particular when it comes to using rare knowledge in the training data (specific knowledge of a controller or long-tail [knowledge])or using current data that was not available in the training data. This can lead to the output of inaccurate personal data. In a RAG system, the context of the LLM input is specifically adapted. If certain conditions are met, this can have a positive effect on the accuracy of the generated output. However, it cannot be completely ruled out that the LLM may still output inaccurate personal data. In contrast to the information contained in an LLM, a major advantage is that the data in the reference documents can be deleted or updated if it is incorrect or outdated. Compared to an LLM operated on its own, the retriever, as a component of a RAG system, can reduce the generation of inaccurate data by enriching the information in the input prompt. RAG systems therefore offer a measure to avoid inaccurate personal data in the generated output, provided that the relevant information is available in the reference documents.

AI hallucinations occur when AI models generate information that sounds plausible but is not supported by their training data.

In our scenario, the answers generated by the RAG system are ideally based on the information from the reference documents. Therefore, the RAG method can reduce hallucinations and decrease the output of inaccurate personal data within the meaning of Art. 5 (1) (d) GDPR. However, challenges remain, in particular:

- Quality of reference documents: The reliability of RAG systems depends heavily on the quality, timeliness, and completeness of the reference documents used.

[10] Kandpal et al., Large Language Models Struggle to Learn Long-Tail Knowledge, https://doi.org/10.48550 /arXiv.2211.08411.

Incomplete or outdated data leads to incorrect outputs. The responsible body must therefore regularly check whether the reference documents used meet these requirements.

- Quality of data preparation: When implementing the system, it should be noted that, with regard to parameterization and data preparation, disruptive data such as headers, footers, and page numbers should be removed before converting a document into text sections or chunks, and that the document should be converted into continuous text. The overlap and size of the text sections also influence the quality of the data that can be used for enrichment. Care should be taken to ensure that individual chunks contain meaningful sections that are as complete as possible. If personal data is not required, it should be removed or anonymized.
- Quality of embedding: When selecting an embedding model, specific requirements should be taken into account, e.g., that it has been trained with German-language texts if German-language reference documents are used. Otherwise, less relevant chunks may be assigned to the input prompt, for example.
- Context fidelity of the LLM used: Depending on the design of the RAG system and training of the LLM, the calculations in the LLM may tend to reproduce the knowledge from the training data and discard the content from the reference documents, especially in the case of contradictory content.

  To counter this, depending on the chosen design, it is conceivable that further technical measures will need to be taken. One option to consider is a system prompt that instructs the RAG system to respond exclusively using the referenced sources.

A RAG system can also be operated with external data sources or use these additionally. In this case, a temporary vector database is created and filled with chunks and embeddings. When integrating external data sources, such as third-party document databases, the legality of use, suitability, and accuracy of the data and the results obtained must be sufficiently checked and ensured. These requirements are not regularly met when integrating a web search. Although the integration of external data can make the generated texts appear up-to-date or specific, it can affect their accuracy and may require the prioritization of internal and external data sources in the RAG subsystem.

3.2. Transparency

If documentation of the sources used for the response (references to chunks or documents) is available, this enables the traceability and transparency of the LLM's input, which was generated based on the content of the reference documents. However,

However, it is not possible to increase data protection transparency with regard to the LLM used.

The same applies to the embeddings stored in the vector database. It is impossible to determine why the respective chunks are assigned the corresponding embeddings, nor what their exact meaning is.

Consequently, transparency in a RAG system is limited to making statements about the expanded query to the LLM component used. Statements about how the outputs of the AI models and thus also of the RAG system are generated are very difficult to make and are therefore non-transparent.

When selecting reference documents, it is important to ensure that information requirements are implemented to guarantee the principle of transparency. Documents from unknown or untrustworthy sources should therefore not be used.

3.3. Integrity and confidentiality

With regard to confidentiality, data protection requirements for the database can be met within the RAG subsystem using established measures. For example, proven technical and organizational measures ( ) and security measures ( ) such as client separation/functional separation[11] and the rights and roles concept[12] can be applied in an RAG subsystem. The rights and roles concept refers here to access restrictions to areas of the vector database and the reference documents. In comparison, the language model (LLM) itself cannot control which information certain users are allowed to access and which information they are not allowed to access. If the data protection requirements are met, personal data with a higher need for protection, e.g., data according to Articles 9 and 10 of the GDPR, can therefore also be processed, as this data does not remain permanently in the LLM, provided that no targeted training or retraining of the LLM with this data takes place, but is stored separately in the reference documents and the vector database.[13]

Since the information from the RAG subsystem is typically used to generate a response when using an RAG system, an LLM that contains less factual information can be used. This opens up a wider range of LLMs for use and, under certain circumstances, allows the RAG system to be operated on-premise. This avoids the transfer of personal data to online operators of large language models.

---

[11] WP 203, several passages. For example, at the departmental level of an organization, since different files (groups of files) are worked on.

[12] For example, based on individuals who are assigned to different organizational units.

[13] The data is always processed in the AI model. See DSK, OH Artificial Intelligence and Data Protection, Version 1.0, May 2024, in particular margin note 15 ff. and margin note 32 ff.

For the same reasons, the integrity of data in a RAG subsystem can generally be better maintained than in an AI system without the RAG method. However, attack vectors such as membership inference attacks or data poisoning attacks on the data in the RAG subsystem must also be taken into account here.[14] It should be noted that the RAG method can also be used as a measure against data poisoning (in relation to the training data of the LLM component).[15]

3.4. Purpose limitation

Providing certain documents to the LLM can enable targeted queries of personal data that are strictly limited to the defined processing purpose. To this end, different roles must be assigned to employees who use the RAG system for different purposes. It must be ensured that employees are assigned the correct role for the processing purpose before the query is made. The purpose limitation of personal data in the vector database can be implemented technically by separating clients or functionally separating the data. Already during the preparation of the reference documents, it should be checked whether the necessary client separation can be implemented, for example by dividing the documents.

Among other things, it must be checked whether the embedding of token sequences still serves the original purpose or has already been assigned to another purpose. This is relevant because the focus here is clearly on processing by a language model and semantic search is also enabled.

At the same time, the use of an RAG system poses a threat to purpose limitation. It can be assumed that controllers are generally unaware of what personal data is contained in the AI model used by the LLM. The transfer of personal data from the vector database to the LLM may result in the linking of personal data from the vector database with the personal data in the AI model of the LLM, thereby violating the purpose limitation principle. This is particularly problematic because linking carried out in this way may not be recognizable in the output of the RAG system. Effective measures that can be taken must be examined at the design stage.

3.5. Data minimization and storage limitation

Data minimization can be supported in the RAG subsystem by determining which documents are stored in the vector database. It is not necessary to add an unnecessary amount of personal data. The scope of personal data

[14] Data Poisoning Attacks, e.g., Zou et al, PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models, https://doi.org/10.48550/arXiv.2402.07867.

[15] OWASP, https://genai.owasp.org/llmrisk/llm042025-data-and-model-poisoning/.

Data in the LLM of an AI system with or without the use of the RAG method is not reduced by the use of the RAG method.

Since the LLM component has fewer requirements in terms of memorizing facts (see 3.3. Integrity and confidentiality), the use of a small language model (SLM) can offer an advantage. A small language model faces the same data protection challenges as a large language model, particularly with regard to the implementation of the rights of data subjects. However, they may memorize less personal data. The suitability of the language model should always be checked and documented.

Reference documents and entries in the vector database can also be deleted. Deletion is performed technically by deleting entries in a database and removing them from the reference documents. These entries can be addressed directly, which means that personal data can also be deleted in a targeted manner. Classic mechanisms for deletion after the expiry of retention periods can thus be used for both the vector database and the reference documents.

Regardless of the advantages of an RAG system, the problems with data deletion in the language model remain.

3.6. Legality

When assessing the legally compliant use of an AI system, the components and their interaction must be considered. In the case of RAG systems, it should be noted that if an LLM has been trained in an unlawful manner, the training would remain unlawful regardless of its integration into the RAG system. However, some of the associated risks to the rights and freedoms of data subjects can be mitigated by using the RAG method. However, the specific impact of an RAG system can only be assessed on a case-by-case basis.

The use of an RAG system requires a legal basis. Among other things, personal data from the reference documents is processed using the embedding model and stored in a vector database. If the use of an RAG system is being considered anyway, this can also have advantages from a data protection perspective. If the requirements for a legal basis under data protection law are not met for the use of an AI system without the RAG method, e.g., because the interests of the data subjects outweigh those of the controller, it can be examined whether the use of the RAG method allows for risk-mitigating measures to be taken that go beyond the measures that the controller is already required to take by law.

. [16] This must be checked in each individual case. It must always be noted that when processing personal data, especially when processing reference documents and in the vector database, a legal basis for data protection must be available.

3.7. Dealing with data subject rights

Since RAG systems use language models, the issues of data subject rights in connection with language models are also relevant here. Overall, it should be noted that the implementation of data subject rights in AI models, in particular LLM, is largely unresolved. This also affects RAG systems.

Regardless of whether the LLM used is personal or anonymous, data subjects are entitled to the rights set out in Art. 15 et seq. GDPR, in particular with regard to the input prompt and output, the reference documents, and the vector database, insofar as personal data is processed in this context. As explained in the previous sections, transparency, information, correction, and deletion can generally be implemented for these ([17]).

4. Conclusion

The outlined implications for data protection principles make it clear that an AI system using the RAG method can reduce some weaknesses, such as hallucinations and incorrect outputs of AI systems without the RAG method, to a certain extent. Challenges such as lack of transparency, purpose limitation, and implementation of data subject rights throughout the RAG system, as well as simplifications in the use of the RAG method, must be examined by the responsible body on a case-by-case basis. This applies in addition to the general data protection challenges. Depending on the design, the use of the RAG method may therefore be considered one of several mitigating measures within the meaning of EDSA Opinion 28/2024.

[16] See EDPB Opinion 28/2024, para. 96 ff.; EDPB Guidelines 1/2024 on processing of personal data based on Article 6(1)(f) GDPR, Version 1.0, October 8, 2024, para. 57.

[17] See EDSA Guidelines 01/2022, https://www.edpb.europa.eu/system/files/2023-04/edpb_guidelines_202201_data_subject_rights_access_v2_en.pdf.

5. Glossary of key terms

This section provides detailed explanations of the definitions and technical terms used in the text to describe the specific IT components. Where possible, reference is made to the definitions provided in the AI Regulation[18] or in the
"Guidance on recommended technical and organizational measures for the development and operation of AI systems"[19] .

Query vector	Result of applying the embedding model to the input
Output	The output is to be understood as the result of the text-generating language model and thus of the RAG system.
Chunk (text section)	This is a text section with a predefined maximum length. Additional documents that are made available to a RAG system as supplementary information are divided into these text sections.
Data poisoning	An attack method for data-driven learning algorithms (including AI algorithms) that attempts to enrich the training data with erroneous data sets in order to degrade the quality of the learning results and, in the best case, cause an unusable model.
Data preparation	Data preparation serves, for example, to remove irrelevant content and divide reference documents into text sections (chunks), which are further processed as part of embedding.
Embedding space	This is a high-dimensional vector space that can encode the semantic meaning of text sequences. Texts encoded in this way serve as input for the language model and for the retriever to search for relevant additional information.
Input (prompt)	This is the input by a user or an upstream IT system into the RAG system.
Embedding	Describes the mathematical mapping rule used to map a text sequence to a vector space.
Embedding model	This is usually an AI model (mostly a language model) that maps text segments (chunks) into a vector space.

[18] https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32024R1689.

[19] https://www.datenschutzkonferenz-online.de/media/oh/DSK-OH_KI-Systeme.pd.

	can be performed. This model is also a component of the RAG system. It can be a language model or another conservative method for encoding text content (possibly without pronounced generative capabilities).
Extended query / extended prompt	The extended prompt consists of the input prompt, the relevant chunks, and, if necessary, additional text sections that are added as context (e.g., "output in German").
AI model	See OH-KI[20] and Art. 3 No. 63 AI Regulation for AI models with general purposes. An LLM is a type of AI model.
AI system	Is a machine-based system designed for varying degrees of autonomous operation, which may be adaptable after it has been put into operation and which, based on the inputs it receives, derives explicit or implicit objectives, such as outputs such as predictions, content, recommendations, or decisions that may influence physical or virtual environments. See Art. 3 No. 1 AI Regulation.
LLM / Text-generating language model	The abbreviation for Large Language Model generally refers to an AI model for general use, as the model is capable of solving text tasks. It receives the query and the text generated in the dialogue and generates the next most probable word. Words are generated until a termination criterion is met. There are LLMs that have fewer parameters but still exhibit generative language capabilities. They also belong to the group of language models and are also referred to as SLMs (Small Language Models).
LLM component	This is the part of the RAG system that implements the LLM.
Retraining/fine-tuning	In the context of language models, the term is used to indicate that the original language model is being modified. Specific documents (e.g., with a certain language style) are used for a particular area of application to adjust the model parameters. This changes the prediction.

[20] DSK, Guidance on recommended technical and organizational measures for the development and operation of AI systems, June 2025, Version 1.0, https://www.datenschutzkonferenz-online.de/media/oh/DSK-OH_KI-Systeme.pdf.

	the next most probable word (see LLM) and, to a certain extent, the style of language. However, the suppression of certain outputs can also be subsequently changed or brought about.
Prompt	The (text) input to the RAG system provided by the user.
RAG system	Represents the entirety of a functional system that RAG can perform. This means that a RAG system has a component that can transfer the additional documents to the embedding space (embedding), a component for storing the pre-processed additional documents (vector database), a component for querying the storage system (retriever), and a language-generating component (as the actual language model). Fig. 1 shows a complete RAG system. A RAG system is an AI system according to Art. 3 No. 1 AI Regulation.
RAG subsystem	A query vector is generated from the input prompt and used to perform a similarity search in the vector database to find the most relevant chunks. These are passed on together with the input prompt as an extended query to the text-generating language model.
RAG method	The abbreviation for Retrieval Augmented Generation describes a method that enriches queries to a language model with relevant information from additional sources in order to submit the query and additional information to the language model as a modified and enriched query.
Reference documents	Reference documents can be located both inside and outside the RAG system and form the basis for the data to be processed by the RAG system.
Retriever	This is the component of the RAG system that is responsible for querying additional information that matches the input to the language model. The additional information can come from a vector database.
SLM	Abbreviation for Small Language Model. See LLM.
Language model	This is the umbrella term for AI models that have been trained with language

	. Language models include, for example, LLMs and embedding models.
Token (sequence)	As a preliminary step to embedding, the text sequences of the query or additional information are broken down into words/word parts, which then represent "tokens." Token sequences are thus only a compressed representation of the text sequences and a preliminary step to embedding. Which words/word parts result in which tokens has also been learned or programmed in advance. The conversion of text into token sequences therefore also represents an AI model, but is more comparable to the function of a dictionary.
Transformer	A specific architecture of a language model for predicting the next most probable word.
Vector-based search mechanism	Searches the vector database for entries with similar embedding vectors. The embedding vector of the prompt is calculated from the prompt. Entries close to this vector are returned as relevant additional information in the search results. This type of search is used by the retriever.
Vector database	Stores, for example, the triples consisting of the elements chunk, token sequence of the chunk (text section), and vector (embedding of the chunk) in a database.