Mozout

Generated on 15/11/2026 00:39

KNOWLEDGE BASE · PUBLIC DOCUMENT

AI that answers from your own documents: RAG explained

How AI Works

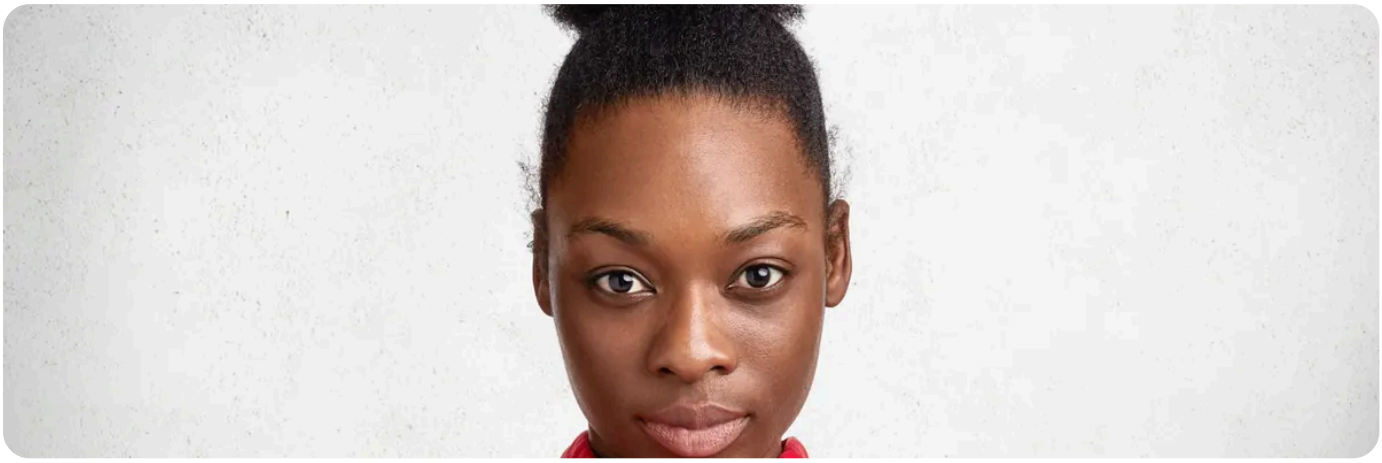
CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-2976	v2026.10.10	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/2976

AI that answers from your own documents: RAG explained

How AI Works Created on **October 10, 2026** 3 min read



RAG (retrieval-augmented generation) is the technique behind nearly every assistant that "answers from our documents". Instead of training a model on your texts, the system **looks up the relevant excerpts and hands them to the model together with the question**. The model answers from them. It cuts down invented answers, but it does not remove them.

How it works, in five steps

1. **The documents are gathered:** the frequently asked questions, the site pages, the manuals, the standard contracts.
2. **They are cut into small pieces,** so that each deals with one thing.
3. **Each piece is turned into numbers** (an "*embedding*") that represent its meaning, by a model made for that. Texts with a similar meaning get similar numbers.
4. **They are stored in a database built to search by meaning** (a vector database). Some common databases have an extension for this.
5. **When someone asks,** the question is turned into numbers too, the system fetches the closest pieces and hands them to the model with the instruction "answer only from this".

RAG or something else

Approach	Good for	Limit
Pasting the text into the chat	One document, once	Does not scale, and you pay for the whole text with each request
RAG	Many documents that change, and varied questions	Only as good as the documents and the way they are cut
Fine-tuning a model	Teaching a style or a format	Not a good way to teach it facts that change

What you need to build one

1. **The documents, clean and current.** If the policy changed and the old document is still there, the assistant will quote it.

2. **An embedding model and an answering model,** from a provider or run by you. See **getting an API key**.

3. **Somewhere to keep the vectors.** A ready-made service, or a database of your own with the right extension. If you install it on a VPS, see **installing PostgreSQL on a VPS** and check that your version supports the extension you pick.

4. **The glue between it all.** An application of your own or a flow in a tool such as n8n. See **what n8n is**.

Mind who can see what.
If you put public and internal documents in the same index, the assistant may quote the internal ones to anyone. Separate the indexes or control access per user.

It can still make things up.
If the excerpts do not contain the answer, a badly instructed model fills the gap with what it “knows”. Tell it to say “I did not find it”, and always show which document the answer came from. See **AI hallucinations**.

Start with a few documents and a list of test questions whose answer you know. If it does not get those right, it is not ready for customers.

RECOMMENDED

Want to run an assistant like this on a VPS of your own? See the plans and choose the size.

See VPS plans →

SEE ALSO

- Adding an AI chat assistant to your website: the options →
- Tokens explained: how AI model APIs are counted →
- Installing PostgreSQL on a VPS →
- Hosting an AI agent: why it needs a VPS and not shared hosting →

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-2976 · v2026.10.10 · Generated on 15/11/2026 00:39