

Grounding-Gate for RAG-Based Public-Service Assistants

Technical briefing for a website, GitHub repository and Hugging Face release. This document explains the failure assumptions, the core prototype function and the proposed safety layer. It deliberately does not disclose internal details of broader Drift-Limiter mechanisms.

Short claim

A Grounding-Gate checks a language-model draft before it is shown to users, using the sources that were actually retrieved for the answer. If the draft is not sufficiently supported, it is not shown unchanged: the system asks a clarifying question, shows only the supported part, blocks the answer or routes the user to an official source.

Scope	Public description
Prototype	Installable Python core with CLI, test set, claim checking, pluggable judge and audit log.
Demo	Static website demo without an API, using curated cases and a visible before/after explanation.
Boundary	Not an Ida clone, not a production public-service answer system, not a guarantee against all hallucinations.

1. Background

Publicly described systems such as ida combine a language model with Retrieval-Augmented Generation (RAG): the model receives relevant material from a knowledge base and formulates an answer from it. According to BRZ, ida uses reviewed content from [oesterreich.gv.at](https://www.oesterreich.gv.at) and [id-austria.gv.at](https://www.id-austria.gv.at), runs on BRZ LLMAaS with a European Mistral model, and uses RAG to reduce hallucinations and false statements. The official ida page also states that erroneous results and hallucinations cannot be completely excluded.

Our starting point is therefore not that RAG is wrong. The starting point is that RAG alone does not guarantee that every individual statement shown to the user is actually supported by the retrieved sources.

2. Assumptions About Failure Modes

The following assumptions are intentionally simple and publicly explainable. They are sufficient to motivate the prototype.

Failure mode	Assumption	Risk
Unsupported added claim	The model may use real sources and still add a claim that is not supported by them.	A plausible but incorrect answer.
Ambiguous question	A short question may refer to several legitimate public-service or help paths.	The system chooses the wrong path too early.

Failure mode	Assumption	Risk
Timeliness / hot facts	Some questions require current authoritative sources, not general FAQ pages.	Outdated or off-topic answer.
User challenge drift	Under follow-up pressure, a model may revise a correct answer into an incorrect one.	Loss of trust through inconsistent behavior.
Scope boundary	Not every legitimate question belongs to the current knowledge domain.	The system either appears unhelpful or invents authority.
Overconfident language	The wording may sound more certain than the evidence warrants.	Users overestimate how binding the answer is.

3. Core Idea: Grounding-Gate

The Grounding-Gate is an output-side checking layer. It sits after retrieval and answer drafting, but before the answer is shown to users. It does not replace RAG or the language model. It checks whether the draft actually fits the sources.

Principle

The generator does not grade itself. A separate checking layer evaluates the answer claim by claim against the sources.

Prototype Pipeline

Step	Description
1. User question	The original question is preserved because short answers can only be checked in context.
2. Sources	The source passages that were actually retrieved are passed to the gate.
3. DraftAnswer	The language model first produces only a draft answer.
4. Claim check	The draft is split into checkable claims, and each claim is evaluated against the sources.
5. Decision	Depending on the result: show, clarify, show only the safe part, block or route to an official source.
6. Audit	The decision is logged so failure cases can become tests.

4. Decision Logic

The demo uses plain-language decisions. Technically, the gate can still use scores and confidence bands; what users see are clear actions.

Demo term	Technical meaning	User-facing result
Show answer	All essential claims are supported by sources.	Show the answer with a visible source.
Ask briefly	The question is ambiguous or the correct path is unclear.	Ask one short clarifying question.
Show safe part only	Part of the answer is supported, but an added claim is not.	Show only the supported part.
Do not show	The answer does not match the sources.	Do not display a false answer.
Route onward	The question is outside the current knowledge domain or needs an authoritative source.	Provide an official link or suitable point of contact.

Note on CalibrateState

CalibrateState, as mentioned in the demo, should be understood as a prior sharpening step: What kind of question is this, how clear is the task, is a short clarification needed, and which knowledge domain is responsible? The term belongs to a broader idea called Drift Limiter (DOI: <https://doi.org/10.5281/zenodo.20633292>). This public briefing only shows the derived user value; internal Drift-Limiter mechanisms are intentionally outside the scope of this document.

5. What the Prototype Can Do Today

- Installable Python core without mandatory external dependencies.
- CLI commands: check, harvest and build-testset.
- Pluggable judge interface: a simple offline checker for development, plus a Mistral-compatible checker for semantic tests.
- Fail-closed behavior: without sources, or if checking fails, the answer is not simply passed through.
- Test set with supported FAQ questions and curated negative cases.
- Audit log as a basis for error analysis and regression tests.
- Static website demo without an API, making the behavior visible.

Implementation status

The current prototype implements the Grounding-Gate core. Not all extensions described in this briefing are fully implemented. Parts of the proposed future work - especially CalibrateState and Drift-Limiter-adjacent functions - are currently research and roadmap components, not part of the public base prototype.

6. What Is Deliberately Not Claimed

- The prototype is not a rebuild of ida.
- It is not a production public-service answer system.
- It does not prove complete error-freedom.
- It does not replace domain review or production-grade calibration against the real infrastructure endpoint.
- It does not publish internal Drift-Limiter details; the public core is the Grounding-Gate.

7. Why This Is Attractive for Open Source

The prototype is small enough to understand, fork and test. At the same time, it addresses a real problem in public AI systems: How do we prevent a plausible answer from being delivered when its source basis is not strong enough?

Audience	Value
GitHub	A clear, testable core with CLI and readable code. A good base for issues, pull requests and extensions.
Hugging Face	A demonstrable safety component for RAG systems: dataset cases, demo and reproducible evaluation.
Public sector / civic tech	The decision logic is explainable: show, clarify, show only the safe part, block, route onward.
Accessibility community	The demo can show how sources, status and next steps can also be made available to assistive use cases.

8. Public Short Description

Short description

Grounding-Gate is a compact open-source prototype for independent source checking of RAG answers. It shows how a public-service assistant can check, before output, whether individual claims are actually supported by the retrieved sources.

Sources and Reference Points

- BRZ: Technical background on ida, the intelligent digital assistant, 22 June 2026.
<https://www.brz.gv.at/presse/technischer-hintergrund-ida.html>
- oesterreich.gv.at: Chatbot ida. <https://www.oesterreich.gv.at/de/landingpages/chatbot-ida>
- Austrian Federal Chancellery: Federal ministers and state secretaries.
<https://www.bundestkanzleramt.gv.at/bundestkanzleramt/die-bundesregierung/bundesministerinnen-und-bundesminister.html>
- Methodological reference point: Drift Limiter / CalibrateState concept. DOI: <https://doi.org/10.5281/zenodo.20633292>
- Project attribution: Verein "AI Forschungsinstitut fuer Menschen", 1100 Vienna, Austria, office@aifim.at, ZVR number 1547230392; Lead Engineer: Dejan Nikolic.
- Prototype materials: ida-gate, demo and test set in the repository package.