Requirements Document

1. Introduction and Objectives

1.1 Background

1 EFRAG operates the ESRS Knowledge Hub as a central knowledge platform for the European Sustainability Reporting Standards (ESRS). The Hub provides standards, Implementation Guidance (IG), Q&As, glossary content, the IG3 datapoint list (as a separate table within the Hub) as well as XBRL taxonomy elements and their complex cross-references in a structured and curated form.

2 As part of the further development of the Knowledge Hub, it is to be complemented by a Large Language Model (LLM) based assistant (the AI Assistant) that enables users to query the ESRS in natural language and receive context-accurate, source-bound answers.

3 The term "AI", standing for artificial intelligence refers to the fact that the assistant is able to understand natural language questions and queries and to respond accordingly.

4 EFRAG is welcoming proposals and/or quotes from service providers and software vendors on implementing the system as described in this document. Proposals can be submitted via mail to call.for.tender@efrag.org until 27 July 2026, see more information in chapter 8. Offer Submission Notes.

1.2 Project objective and scope

5 The objective of the project is the development and productive introduction of an AI-based chat assistant (or chat bot) that is fully integrated into the existing EFRAG Knowledge Hub and operates exclusively on the verified, curated content of the Hub. Every answer is supported by explicit source references. The data models, paragraph hierarchies, and cross-reference structures of the Hub are fully utilised. The assistant is operated on EU-based infrastructure with guaranteed data sovereignty and is made available to all registered users of the Knowledge Hub as a productive tool.

6 The most important principles of the design of the AI-assistant are:

(a) to avoid any reputational damage for EFRAG, e.g. through hallucinations, or providing wrong information;

(b) giving advice on anything that is beyond the KH content and scope, e.g. on the question if a certain disclosure is complaint to the ESRS or any other behaviour of the assistant that might potentially harm EFRAG or its reputation. Uploading of own documents, pictures, videos, etc. (like it is possible in tools like ChatGTP) is not foreseen, as well as content generation like "please tell me a joke about the omnibus";

(c) EFRAG requires a fixed recurring monthly pricing model for the hosting and operation of the LLM, in order to avoid any risk of unforeseen budget overruns.

Consequently, solutions based on token- or usage-based billing may be unsuitable and solutions running on dedicated hardware with custom models preferrable.

(d) keeping the environmental footprint of the solution low, i.e. balancing the benefits of a more accessible KH content through an AI-assisted solution with the costs, energy consumption, GHG emissions and other environmental aspects caused by it.

1.3 Strategic Principle — Integration into the Existing Knowledge Hub

7 The Knowledge Hub AI Assistant is not a standalone system, but an organic technical and functional extension of the existing EFRAG Knowledge Hub. All components (data integration, admin interface, frontend integration, rights system, deployment, and maintenance) should fit into the existing Hub architecture, its data structures, its codebase, and its operating logic.

8 The Knowledge Hub exposes a RESTful API that enabled retrieving its raw content.

9 Solutions that implement the assistant as a separate isolated system with its own user management, its own admin interface, its own deployment pipeline, or its own data storage are to be avoided and will be treated with lower priority in the evaluation process.

2. Functional Requirements

2.1 Core Functionality of the Assistant

No.	Requirement	Priority
F-1	The assistant answers user questions in natural language. While the primary focus is to deliver the Knowledge Hub content, the Assistant is supposed to respond in a formal but kind way, potentially with follow-up questions.	Must
F-2	Every answer is generated exclusively from the content of the Knowledge Hub. External model knowledge must not be used as a basis for answers. The content must include all texts from interactive and non-interactive documents (e.g. PDFs), as well as the content of the interactive side bar (including custom side by entries).	Must
F-3	Every answer contains precise, clickable source references that link directly to the corresponding KH content (standard, paragraph, Q&A, glossary entry).	Must
F-4	When no reliable answer can be derived from the available sources, the assistant transparently declines (e.g. "I cannot find a reliable answer to that in the available sources"). Hallucinations are to be structurally prevented.	Must
F-5	The assistant operates in mode, that considers a limited number of responses and questions of the conversion, in order to limit the token usage. A continuing chat conversation referring back to earlier and older answers (as in ChatGPT) is not foreseen. The limit must be configurable in order to fine-tune the experience and token usage.	Must
F-6	Input is exclusively textual. Uploading images, documents, or other file formats is not foreseen.	Must
F-7	Next to the input field, the user is provided with an optional scope selection allowing the search area to be narrowed (e.g. ESRS 2023, Simplified ESRS, VSME, Meetings). If the scope is not selected, either the full content is considered (and answers are provided on all content items), or the assistant needs to ask the user if the scope should be limited to a specific document or standard.	Should

No.	Requirement	Priority
F-8	A chat history enables the user to revisit previous questions and answers. Users must be able to delete their history, or disable the history functionality entirely.	Must
F-9	Security mechanisms preventing prompt injection must be implemented. The applied mechanisms must be described in the solution concept and documented transparently in the technical documentation.	Must
F-10	In general, the full (public) content of the KH should be in scope for the assistant. The content that is considered by the assistant can be prioritized and excluded[1]. The assistant accesses exclusively the contents of the Knowledge Hub explicitly listed below (whitelist). Dynamic content added or updated using the built-in editors shall be considered as well (e.g. adding a new version to the standard, uploading a new PDF). When multiple versions of content exist, the assistant might consider the latest one, but should be able to answer questions to all versions and do comparisons if requested. Any other content not listed here must not be retrieved, indexed, or used in answers. **(a) Standards and standard texts (document sets)** The textual base of the assistant's knowledge. The following current document sets are in scope. If new versions are added or text is edited, this is to be considered accordingly, including understanding and ability to compare the versions • 2023 ESRS • 2023 ESRS Implementation Guidance • 2025 Draft Simplified ESRS • 2025 VSME Recommendation **(b) Structural text elements within the standards (links and relationships)** These elements are the navigational anchors of the standards. They are always retrieved together with their hierarchical context: • Links to paragraphs (including hierarchy and numbering logic) • Links to Application Requirements (AR), each of which is attached to a parent paragraph **(c) Implementation Guidance (IG)** The IG content extends specific paragraphs of the standards and must always be retrieved in the context of the paragraph it elaborates on: • IG 1 — Materiality Assessment (MA) • IG 2 — Value Chain (VC) • IG 3 — Datapoint List (a separate table within the Hub), with each datapoint directly linked to the paragraph it is derived from **(d) Q&As**	Must

[1] The ESRS reports provided on insights.efrag.org are not part of the KH and must not be considered by the assistant.

No.	Requirement	Priority
	Every Q&A is anchored to one or more paragraphs and must be retrievable assistant from the paragraph side and from the Q&A side: • EFRAG-published Q&As, each identified by its official ID • The mapping from each Q&A to the paragraphs it reference **(e) Glossary** Glossary entries are linked to the paragraphs in which the defined terms appear. The link is bidirectional and must be respected during retrieval: • Glossary entries including definitions and references to paragraph. **(f) XBRL Taxonomy** The XBRL taxonomy provides the machine-readable counterpart to the textual paragraphs and datapoints. The assistant uses the mapping to identify the formal taxonomy expression of a content item: • XBRL taxonomy elements • Mapping between XBRL elements and their corresponding datapoints and paragraphs **(g) Cross-References** Cross-references are the connective tissue of the entire knowledge graph and must traverse when answering: • All cross-references maintained within the Hub between paragraphs, ARs, IGs, Q&As, glossary entries, datapoints, and XBRL elements **(h) Document-set membership** Every content object carries an explicit assignment to a document set (2023 ESRS, Simplified ESRS, VSME, etc.). This is used as a hard filter during retrieval to ensure that answers never mix content from different regulatory frames: • Document-set assignment metadata must be retrievable from the index at all times and must be applied as a filter criterion during retrieval **(i) Custom Content Layer** The curated answer store defined in Section 2.3. Entries in this layer take precedence over content generated from the standards and are clearly labelled • Entries from the Custom Content Layer (see Section 2.3), with priority during answer generation. **(j) Interoperability data**	

No.	Requirement	Priority
	Interoperability mappings are attached to individual paragraphs and connect them to corresponding provisions in adjacent frameworks. They are retrieved together with the paragraph they belong to: • All interoperability data attached to paragraph. **(k)Side-bar content (paragraph-attached text content as part of the interactive side-bar)** The interactive sidebar allows additional textual content to be attached to individual paragraphs. This mechanism was for instance used for the 2025 Draft Simplified ESRS to represent the *Log of Amendments*. • This includes any links to other KH content embedded in the entry are extracted and respected as cross-references. The content of external links to 3rd part webpages are out of scope. **(l) Static content (PDFs)** The static content of the KH, as provided under the "Historical documents" and "SRB and SR TEG meeting documents" shall be in scope.	
F-11	The number of questions that are possible to be answered per hour or day for users should be configurable via backend, giving EFRAG a way to reduce usage and improve performance when needed.	Must
F-12	A backend configuration enables EFRAG to finetune the confidence level on how certain the LLM is to provide an answer to a specific question.	Should
F-13	The assistant must operate on the English version of the KH content, but it should be able to respond to questions raised in other languages by translating them to English. Quotes from the sources must always be provided without translation (this might result in mixed-language responses).	Must/ Should
F-14	If a question or the assistant explicitly refers to multiple versions of a document, e.g. 2023 ESRS and 2025 Draft Simplified ESRS, the response must consider this and be able to provide information on both and provide a comparison if needed or appropriate.	Must

2.2 Connection to the Hub Data Structures

No.	Requirement	Priority
F-15	The assistant accesses the Knowledge Hub data model, see F-10 for full list of contents to be included in the knowledge base.	Must
F-16	The cross-reference logic of the Hub (paragraph → AR → IG → Q&A → glossary → XBRL element) is fully utilised by the assistant. Answers must not be restricted to isolated text fragments.	Must
F-17	The assistant may only draw on content that has been explicitly connected, prepared, and indexed as part of an explicit integration (i.e. using the interactive side-bar content).	Must
F-18	Answers do not mix content from different document sets (e.g. ESRS Set 1 (2023) and Simplified ESRS) within a single answer, unless clearly separated in the answer. A document-set classification must be	Must

No.	Requirement	Priority
	performed prior to content retrieval, unless the user explicitly selected to cover all content, and the subsequent search must be restricted accordingly.	

2.3 Custom content layer

10 In addition to the answers derived from the Hub, EFRAG requires a separate component in which its own technical question-answer pairs can be stored. These are considered by the assistant additionally to the content of the KH. This component is referred to as the Custom Content Layer in the following:

No.	Requirement	Priority
F-19	The assistant is to be complemented by a Custom Content Layer. EFRAG Secretariat can store selected questions and answers within it specifically for the assistant (e.g. news items, contact details, new information on workplan, frequently asked questions received via the EFRAG Secretariat mailbox).	Must
F-19a	Optionally, PDF files with content can be stored in the Custom Content Layer, and its content is considered by the assistant.	Should
F-20	The Custom Content Layer is to be implemented as an extension of the existing Knowledge Hub admin interface. A separate interface is to be avoided.	Should
F-21	The Custom Content Layer uses the existing rights and role system of the Knowledge Hub. A separate authorisation model is to be avoided.	Should
F-22	Entries in the Custom Content Layer must be clearly distinguished from the existing Hub ESRS Q&As. The existing ESRS Q&As of the Knowledge Hub are independent content with no relation to the assistant.	Must
F-23	Curated entries of the Custom Content Layer are prioritized by the assistant and clearly labelled to users as "EFRAG Secretariat Answer".	Must
F-24	Entries must be able to reference existing Hub content (standards, paragraphs, Q&As, glossary) directly. The linking should be possible within the admin interface without manually copying IDs or URLs.	Must
F-25	Entries are subject to versioning with change history.	Must
F-26	A custom “instruction” field can be maintained in the backend, enabling adding and editing short custom instructions that are considered before every question, in order to react quickly on issues discovered. Example: “Do not answer any question to topic XYZ”.	Must

2.4 Frontend Integration

No.	Requirement	Priority
F-27	Integration into the Hub is required, providing a modern and accessible UI experience using the EFRAG branding.	Must
F-28	Assistant answers contain deep links into the corresponding Hub view (standard, paragraph, Q&A, glossary entry).	Must

No.	Requirement	Priority
F-29	The assistant interface is to be implemented fully responsively for desktop, tablet, and smartphone, including touch-optimised interaction patterns.	Must
F-30	The assistant interface complies with WCAG requirements (keyboard operation, screen reader compatibility, sufficient contrasts).	Must
F-31	The user receives transparent feedback regarding the current processing steps of the assistant (e.g. "Searching relevant standards", "Checking cross-references", "Verifying sources", "Formulating answer").	Should
F-32	"EFRAG Secretariat Answers" from the Custom Content Layer must be visually clearly distinguished from AI-generated answers in the display.	Must
F-33	Prior to the technical implementation of the frontend, a UI design and an interactive clickdummy are presented and iteratively agreed upon with EFRAG.	Must
F-34	The assistant is integrated within the Knowledge Hub with the existing authentication mechanism (Active Directory / SSO of the EFRAG platforms).	Must
F-35	Rate-limiting and token caps per user are to be provided in order to prevent misuse by individual users.	Must
F-36	The assistant's answers are to be presented in a factually correct manner, in natural language and in well-readable formatting (sensible paragraphs, highlighted sources, clear structuring of longer answers).	Must
F-37	Use of the assistant frontend takes place after passing through the existing login/registration access mechanism. If the paywall logic of the Knowledge Hub is activated, it must be possible to require a valid subscription to use the assistant.	Must

2.5 MCP Interface (Model Context Protocol)

No.	Requirement	Priority
F-38	The Knowledge Hub must expose its content through a dedicated MCP server (Model Context Protocol), enabling external AI clients (e.g. Claude Desktop, ChatGPT, IDE tools, custom agents) to access public Hub content via a standardised protocol.	Must
F-39	Authentication is performed via a separate, per-user app password, generated and revoked by the user from within the Knowledge Hub. EFRAG SSO credentials are not exposed to MCP clients.	Must
F-40	The MCP server provides read-only tools for content retrieval (e.g. search_content, get_paragraph, list_cross_references, get_secretariat_answer, lookup_glossary_term, get_datapoint). Tools for curating, editing, or managing Custom Content Layer entries must not be exposed.	Must
F-41	Rate-limiting and audit logging apply to the MCP interface analogously to the assistant frontend (per-user quotas, anonymised query logging, abort/error handling).	Must

2.6 Knowledge Base Updates and Update Governance

No.	Requirement	Priority
F-42	New content (including new versions) in the Knowledge Hub (e.g. new standards, Q&As, guidance) is integrated into the assistant's knowledge base through an automated update pipeline. A complete rebuild of the index is not necessary; only affected knowledge units are updated incrementally.	Must
F-43	Content that will be added to the KH in a new or different data structure (e.g. new format classes such as embedded IGs in standard texts or new Q&A formats which might require adjustments to the underlying database or API) is not automatically part of the automated update pipeline. Such extensions require explicit technical adjustments in order to digest the content accordingly.	Must
F-44	After every update of the knowledge base or the AI model, an automatic A/B test are performed by the system and which delivers reports with the results, including findings on the KPIs security, performance, and reliability. The report documents the respective change relative to the previous state.	Must
F-45	For every update, a protocol must be issued that complies with the requirements of the EU AI Act (Regulation (EU) 2024/1689), in particular with regard to traceability, risk assessment, and documentation obligations.	Must

3. Non-Functional Requirements

3.1 Architecture and Technology

No.	Requirement	Priority
NF-1	The system must guarantee source traceability, accuracy, and controlled knowledge boundaries when delivering answers.	Must
NF-2	A citation enforcement step must be implemented to make sure, that every generated statement is retrieved from the available sources. Statements without source backing must not be delivered to users, e.g. by removing them before the answer reaches the user.	Must
NF-3	The architecture of the entire system must follow the requirements as defined in “9. Technical Integration Requirements”.	Must
NF-5	The assistant stack is to be implemented in a containerised form (Docker), analogously to the existing Knowledge Hub architecture (DB, backend API, cache, web application).	Must
NF-6	The AI model runs on an EU-based hosting environment. The AI workload must be separated from the regular Hub operations so that Hub performance for non-assistant users is not affected.	Must
NF-7	A later replacement or upgrade of the AI model in use (e.g. when a newer, more capable model becomes available) is treated as a separate engagement and is not part of the initial commission.	Notice

3.2 Performance and Scalability

No.	Requirement	Priority
NF-8	Response times under productive operating conditions should meet the following targets: - Under 10 seconds when the question can be answered from a focused content scope of approximately one document set (e.g. a single standard such as ESRS E1 including its Application Requirements, related Implementation Guidance, and linked Q&As — roughly equivalent to 30 pages of text). - Under 30 seconds when the question requires a broader content scope spanning multiple connected sections within the same document set (roughly equivalent to 100 pages of text). - Never more than 60 seconds, regardless of question complexity and content scope. Queries that cannot be answered within this hard ceiling must be terminated with a user-facing notice rather than left to time out silently. The response time targets apply under typical Knowledge Hub usage conditions, defined as approximately five concurrent users actively submitting questions. Under elevated load (see NF-9), response times may degrade gracefully, but the 60-second hard ceiling remains in effect for every individual request.	Must
NF-9	The assistant must reasonably absorb load peaks: under elevated load, response times may slow down, but requests must not abort or be lost. A suitable queuing, throttling and notification logic is to be provided.	Must
NF-10	In order to avoid a high peak load during the launch of the assistant, a phasing-in with an invitation system is foreseen, that enables EFRAG to give access to 100 users, that can then share one invite per day with 5 additional users, and so on.	Must

3.3 Data Protection, Data Sovereignty, and Security

No.	Requirement	Priority
NF-11	All content, user interactions, and model requests are processed exclusively on EU-based infrastructure. Data flows to providers outside the EU are not permitted. European Large Language Models (LLMs) are preferred.	Must
NF-12	The solution must be hardened against prompt injection and misuse.	Must
NF-13	The assistant must not generate answers from the general knowledge of the language model used, but exclusively from the indexed, explicit Hub context.	Must

3.4 Statistics, Operations, Logging, and Observability

No.	Requirement	Priority
NF-14	A monitoring, statistics and logging concept is to be provided (availability, latency, error rate, token consumption).	Must
NF-15	On demand, user queries are logged anonymously for continuous quality evaluation, together with the user's subsequent reaction where applicable (e.g. rating, follow-up question, premature abort). This feature can be disabled after a certain testing period.	Must

No.	Requirement	Priority
NF-16	An automated test suite based on the acceptance questions (see Section 4) is automatically executed prior to every productive release.	Must

4. Hosting and Ongoing Operation

4.1 Hosting

No.	Requirement	Priority
H-1	Hosting is provided by the contractor on EU-based infrastructure.	Must
H-2	Hosting costs are to be quoted on a monthly basis to enable predictable budgeting. A fixed hosting price for a hardware, allowing EFRAG to calculate the ongoing cost for the usage of the KH is preferred over an dynamic token-based invoicing, that will lead to varying costs based on the usage.	Must
H-3	A price indication for ongoing hosting operation must be submitted together with the offer. A concrete hosting offer, based on the requirements measured during the project, is to be submitted after project completion.	Must

4.2 Maintenance, SLA, and Optimisation

No.	Requirement	Priority
H-4	A Service Level Agreement (SLA) is to be offered, regulating availability, response times, and maintenance windows.	Must
H-5	A continuous maintenance and optimisation capacity is to be offered (regular fine-tuning, prompt optimisation, model evaluation, maintenance of the test suite). The capacity is to be offered optionally as a monthly fixed allocation or on an hourly basis.	Must
H-6	For SLA agreements with an annual term, volume discounts on the reserved capacity are expected.	Should

5. Delivery and Ownership Conditions

No.	Requirement	Priority
L-1	The technology developed for EFRAG, including the source code are provided at the end of the project, enabling EFRAG to independently developing and maintaining the system. EFRAG owns the full rights of the intellectual property. If parts of the source code (or external modules) are integrated into the system, that require a license and/or specific agreement to be used, those need to be listed in the proposal including the detailed terms and conditions under which they are being used, made available to EFRAG and under which restrictions.	Must
L-2	Open-source components used are documented with version, licence, and source.	Must
L-3	A technical administrator documentation (installation, update, backup, recovery) is to be provided.	Must
L-4	The public go-live (and final payment) will only happen after a successful acceptance test, that ensures that no hallucinations and/or harmful responses occur, that might damage EFRAGs reputation.	Must

6. Schedule and Milestones

11 EFRAG targets the following timeline:

Milestone	Timing
Project start	July/August 2026
Start of internal testing phase	Q3 2026
Start of extended testing phase / introduction phase	Q4 2026
Start of productive operation	End of 2026 or early 2027

12 Deviations are to be justified in the offer.

7. Architecture and Technical Integration Requirements

13 Overview of the technical requirements for the Knowledge Hub Assistant, derived from the existing system architecture of the Knowledge Hub. The planned extension must follow this architecture, as it ensures a unified codebase, prevents duplicate implementations and guarantees long-term maintainability.

7.1 Technology Stack (mandatory)

14 The assistant must be based on the same technologies as the existing system in order to be maintainable within the monorepo. The components listed are required in their native form; they must not be implemented as wrappers around alternative integration shells. The goal is a unified, harmonised architecture that consistently supports future maintenance, refactorings, and version upgrades.

Area	Requirement
Backend	NestJS 11, Node 22
Frontend (chat widget)	Angular 21, standalone components (no NgModules)
Frontend (admin UI)	Angular 21, standalone components
SSR	`@angular/ssr` + Express 5 (for all publicly accessible pages)
State management	`@ngrx/signals` SignalStore — no NgRx with actions/reducers/effects
Language	TypeScript 5.x, consistently strongly typed
Monorepo	Nx 22 — all packages registered as Nx projects

7.2 Backend Structure (NestJS)

7.2.1 Module Structure

15 Every new functional feature of the assistant must be implemented as a self-contained NestJS module that follows the existing pattern:

```
apps/api/src/app/<assistant-feature>/
  <assistant-feature>.module.ts
```

```
  <assistant-feature>.controller.ts
  <assistant-feature>.service.ts
  <assistant-feature>.dto.ts
  <assistant-feature>.model.ts          ← Sequelize model
  trans-<assistant-feature>.model.ts    ← multilingual support
```

16 The new module must be registered in the application module file.

7.2.2 Database Models

- ORM: `sequelize-typescript` — all models inherit from `Model<TModel, TCreate>`
- Every table requires a migration file and must be registered in the migration service
- Migrations must be executed automatically on API startup to ensure update capability — including assistanth up and down migrations

7.2.3 Multilingual Support (Translation Pattern)

17 Table translations must be integrated into the existing translation system:

- `A Trans*` shadow table is required for every translated entity
- Columns: FK to the main table, `languageId`, translated fields, `searchVector` (tsvector)
- Resolution in queries via `COALESCE (Trans*.column, *.column)`

7.2.4 DTOs and Validation

- Every API endpoint receives a DTO with class-validator decorators
- DTOs are stored in the `<feature>.dto.ts` file of the respective module
- No custom error handling — the global `DatabaseErrorFilter` and `SentryGlobalFilter` apply automatically and must be supported

7.2.5 Rich-Text Content (Curated Answers)

18 If the assistant stores curated answers that contain formatted content:

- Content is stored as **ProseMirror JSON** (analogous to paragraph content)
- The existing ProseMirror schema must be used — custom schemas are not permitted
- The admin editor must use the existing `RichTextEditorComponent`

7.3 Shared Types and Interfaces

7.3.1 API Interfaces

19 All new types, interfaces, and enums shared between API and frontend belong in:

```
libs/api-interfaces/src/lib/
  models/<entity>.interface.ts
  consts/<enum>.enum.ts
```

20 They are re-exported via `libs/api-interfaces/src/index.ts`. No duplicate type definitions in API and frontend.

7.3.2 Permissions

21 New assistant functions (e.g. "create curated answer") require new permission entries. No separate rights system — only extend the existing one.

- Register the namespace

- Register the namespace together with the role-permission files and the corresponding permissions

7.4 Frontend Integration

22 General rules for frontend development:

- Implementation must be done as **an Angular standalone component**
- The component must be able to communicate with neighbouring and hierarchical components via inputs and outputs
- It must be SSR-safe: all window/document accesses must be guarded by `isPlatformBrowser()` or `afterNextRender()`
- `ChangeDetectionStrategy.OnPush` must be set to optimise rendering
- Access protection via the existing `RequiresPermissionDirective` from `libs/shared-components`
- All components must be lazy-loadable; the use of Web Components is not permitted

7.4.1 Chat Widget (Content App)

23 This component will be embedded by us into the interactive view in order to expose the assistant to end users.

7.4.2 Admin UI (Assistant Management)

24 New admin pages (e.g. for managing curated answers) follow the existing pattern:

```
apps/admin/src/app/<assistant-feature>/
  <assistant-feature>-index.component.ts
  <assistant-feature>-index.store.ts     ← @ngrx/signals signalStore
  <assistant-feature>-edit/
    <assistant-feature>-edit.component.ts
    <assistant-feature>-edit.store.ts
```

25 Permission guards must be used to secure the API endpoints

7.4.3 State Management

26 Exclusively `@ngrx/signals signalStore()` — no classic `NgRx` store, no `BehaviorSubject`, no custom state management

27 Stores are route-scoped (registered as providers on the route), not global

7.5 Authentication and Authorisation

- **No separate user management** — exclusively use the existing Users table and the role/permission system
- The assistant backend authenticates via `POST /auth/login` (email/password) as a dedicated service account

28 Token rotation: POST /auth/refresh before expiration of the `accessToken`

- Microsoft Entra ID (Azure AD) must **not** be used for the assistant service account — only local JWT authentication
- Subscription status (Pro access) must be considered in the retrieval logic

7.6 Deployment

- The assistant stack is added as additional services to the Docker Compose stack
- All new containers attach to the existing `backend_network` Docker network
- Routing is configured via Traefik using label configuration
- Container images are built via the existing GitLab CI pipeline (Kaniko, custom registry)
- Secrets are passed as Docker environment variables or via `env_file` — no separate secret management

7.7 Content Update Pipeline

29 The assistant index must be updated whenever content changes.

30 A suitable event system (webhooks, message queue, or equivalent mechanism) must be proposed by the contractor and implemented as an extension of the existing API module. An isolated solution outside the existing API architecture is not permitted.

7.8 General Framework Conditions

Requirement	Rationale
No separate user management	All identities must be stored centrally
No separate auth service	JWT and Entra are system-wide and uniform
No separate CSS framework	Angular Material + existing SCSS system
No separate search index	PostgreSQL FTS is the sole search infrastructure for existing content; an additional projection into a vector database / knowledge graph is required for the assistant
No NgModules	The system is fully migrated to standalone components
Reuse of the existing ProseMirror schema	No custom rich-text format
Up/down migrations for all DB changes	Automatic schema management on startup
EU data residency	All data remains on EU infrastructure (existing hosting agreement)

8. Offer Submission Notes

31 The offer is to be submitted with the following components:

(a) Description of the solution approach including architecture (LLM, connection to existing Hub structures, document-set routing) and illustrative examples.

(b) Detailed project and phase plan with milestones, effort, and duration.

(c) Cost calculation for development, hosting, and maintenance/SLA, including a price indication for ongoing hosting operation.

(d) Evidence of experience with the existing EFRAG Knowledge Hub, or a concept of how the Hub-specific expertise will be built up.

(e) Description of the hosting concept including EU data sovereignty.

(f) SLA proposal including maintenance capacity and hourly rates for additional services.

(g) Documentation of the security mechanisms against prompt injection.

(h) Description of the planned A/B test methodology and the AI-Act-compliant update protocolling.

(i) Optional: Suggestions for complementary business models in the context of the AI assistant.

(j) Confirmation of compliance with all Must-criteria from this document.