

NAGE

NAGE / PARTNER CONVERSATION

Build with inspectable AI

KLM candidate standard, SEDIM reference
architecture and the Nage ecosystem

KLM / SEDIM / Nage AI OS / Craton



The integration opportunity

Preserve an output's declared basis as it moves between systems.

Application teams

Present source and signal context in the user's workflow.

Knowledge providers

Connect governed domain sources to identifiable evidence.

Assurance tooling

Consume disclosed records and run independent checks.

The common contract

A candidate standard for shared records, with a reference service operated by Nage.

Record vocabulary

Claims, evidence, source classes and epistemic status.

Conformance

Cumulative KLM-0 to KLM-5 requirements.

Verification

Checks with an explicit scope, algorithm and disclosure boundary.

SEDIM and the base-model role

An implementation architecture with a replaceable FACIES slot.

Current path

Existing Qwen base with grounding and record services.

Layer mechanism

VARVE, CLAST, STRIA and STEMMA define the target composition path.

Craton / Pro

Development-stage bases intended to reduce unmanaged prior knowledge.

Trained weight injection is not yet the reported production inference path.

The Nage architecture

Connect to the record contract at a defined service boundary.

KLM / CANDIDATE RECORD STANDARD

Managed sources

Documents, versions
and lifecycle context



SEDIM

Grounding + inference;
source participation



Output + record

Answer, signals and
signature metadata

Nage AI OS carries the record to applications, reviewers and assurance tools

Conceptual service flow. KLM is a candidate standard; production and development paths remain distinct.

Nage AI OS

Knowledge and records cross several service boundaries.

Knowledge	SEDIM, CONFLUX and BEDROCK.
Substrate	NUCLEUS and RILL.
Delivery and observation	STRATUM, CONDUIT and OUTCROP.
Execution contract	MAGMA is designed; current production serving uses nage-api.

Component availability does not imply every planned feature is deployed.

Disclosure and independent checks

Access to the right artefacts determines what a reviewer can reproduce.

Public-key checks

Integrity and signature checks against an authenticated key.

Auditor disclosure

Claim and evidence inputs for selected recomputation.

Restricted scope

Worker internals and undisclosed calibration may remain unavailable.

A valid signed record is not proof of real execution or answer correctness.

A shared record at the boundary

Preserve the fields, their meaning and their measurement status.

Declared confidence

0.910

Grounded confidence

0.742

Confidence gap

+0.168

Retrieval relevance

0.9422

measured

Source authority

0.600

heuristic

Source freshness

null

unavailable: no source date

Included synthetic fixture, not production evidence. See verification/README.md for runnable checks.

Integration responsibilities

Define the boundary before exchanging records.

Nage

Document record semantics, supported interfaces and implementation scope.

Partner

Preserve identifiers and statuses across the application boundary.

Customer

Define access, retention, review authority and deployment requirements.

Partner evaluation tracks

Choose the track that matches your product.

Application integration

Display a record and its verification scope in an existing workflow.

Domain source integration

Test source versions, disclosure and lifecycle handling.

Verification integration

Check a disclosed record with an independently maintained tool.

Acceptance evidence

A useful pilot produces artefacts both teams can inspect.

Compatibility

Versioned records and reproducible format checks.

Behaviour

Supported, contradicted and absent-evidence examples.

Operations

Measured latency, access boundaries and withdrawal handling.

Programme boundaries

Current service and development commitments should remain distinct.

Current scope

Record and grounding services with documented limitations.

Open work

Public log commitments, historical provenance completion and selected production integrations.

Craton programme

Mechanism validation before scale or release commitments.

Let's explore a partnership

Bring an application, a source domain or a verification use case.

First conversation

Identify the shared workflow and required evidence.

Next step

Agree a scoped technical evaluation and responsibilities.