BILYTICA AI OPERATING SYSTEM

Sovereign AI Framework

Residency, key custody, model routing and Arabic quality treated as architecture, not policy text.

SOVEREIGN · Sovereignty

WHAT IT IS FOR

Sovereignty claims are only as good as the architecture behind them. This framework fixes where data sits, who holds the keys, which model may see which class of data, and how you leave.

WHO RUNS IT

Owned by the CISO and the data protection officer, with architecture accountable for the design.

TYPICAL TIMELINE

Reference design in three to five weeks, then applied to every workload as a standing constraint.

STAGES AND EXIT GATES

1. Classify residency — Decide per data class where it may be processed and by what.
Exit gate: No workload starts before its data classes have a residency rule.
Artefacts: Residency matrix · Transfer register

2. Place the stack — Choose in-Kingdom, private cloud or on-premises per class, including the vector store and logs.
Exit gate: Logs, embeddings and caches follow the same residency rule as the source data.
Artefacts: Deployment topology · Data-flow diagram

3. Custody keys — Define who holds encryption keys, how they rotate and who can be compelled.
Exit gate: Key custody is documented and separable from the platform provider.
Artefacts: Key custody model · Rotation policy

4. Route models — Match each task to the smallest model that passes evaluation within its residency class.
Exit gate: Sensitive classes never route to a model outside their residency boundary.
Artefacts: Model routing policy · Model portfolio register

5. Assure Arabic — Test Arabic terminology, register and answer equivalence as a release condition.
Exit gate: Arabic performance is measured, not assumed from an English benchmark.
Artefacts: Arabic evaluation set · Terminology glossary

6. Plan the exit — Document how models, indexes and data are exported and what remains portable.
Exit gate: An exit can be executed without the incumbent vendor's cooperation.
Artefacts: Exit plan · Portability test

STANDARDS CROSSWALK

STANDARD	WHAT IT SATISFIES
PDPL	Cross-border transfer conditions and processor obligations per data class.

STANDARD	WHAT IT SATISFIES
NDMO	National data classification and sharing controls applied to AI pipelines.
NCA CCC / ECC	Cloud cybersecurity and key management expectations for hosted AI.
SDAIA	National AI ethics principles and Arabic language expectations.

WHAT YOU GET

- **Residency matrix** — Every data class with its processing boundary and transfer basis.
- **Reference architecture** — Deployment topology including vector store, logs and support access paths.
- **Model routing policy** — Task-to-model rules with residency and evaluation thresholds.

WHAT WE REFUSE TO DO

- **Sovereign in the slide only** — In-Kingdom compute with logs and embeddings leaving the boundary is not sovereign.
- **Frontier model for everything** — Routing every task to the largest model raises cost and shrinks the residency options.
- **Translated Arabic** — Machine-translated interfaces read as foreign to enterprise users and fail on terminology.

FREQUENTLY ASKED

Does sovereign mean on-premises?

No. It means the processing boundary, key custody and exit path are known and enforced — that can be in-Kingdom cloud, private cloud or on-premises.

How do you keep cost down under residency limits?

Through model routing: small local models handle most traffic, and only tasks that need more capability escalate within the same boundary.

What does sovereign AI mean in practice?

Data residency classified per workload, the stack placed accordingly, keys held by you, model routing that respects the classification, and a documented exit path — not a label on a slide.

Does sovereignty force one model for everything?

No. Routing sizes the model to the task and the data class, so in-Kingdom or on-premise models serve sensitive workloads while others use larger models where the classification allows.

Book the Data Foundation Assessment

Full framework detail: https://bilytica.com/frameworks/sovereign-ai-framework/

bilytica.com
info@bilytica.com