Single platform or multi-engine? A decision framework that can say no

Capability is a set of gates. Need is scored separately. No amount of need buys you past a failed gate.

Version 1.0 · 18 September 2026 · Frank Garofalo · Chief Data Officer, Federal Civilian · written in a personal capacity

1 The capability gates

Clear all three, or run one platform. These are not scored and they do not trade off against each other — a team that cannot operate two estates safely will not be rescued by having a strong reason to want two.

Gate	Clearing it means	If you do not clear it
Platform engineering	At least one named engineer whose job is the platform itself, not the pipelines running on it.	Two upgrade cadences, two capacity models and two on-call surfaces, owned by nobody.
Governance	One owner for classification, access policy and lineage *across* engines, with authority to say no.	The mirror seam is where policy silently stops being enforced. You find out from an auditor.
FinOps	Capacity units and DBUs tracked separately and attributed to business units.	Two billing models blend into one number nobody can act on. Surprises arrive quarterly.

Gate rule. Fail any one and the answer is single-platform, whatever the need score says. Fix the gap first; it is cheaper than running two estates badly.

2 The need score

Only if all three gates cleared. Score each signal 0–2 and add them.

Signal	0	1	2
Workload diversity	BI and light transforms	Some ML or streaming	Heavy ML/streaming *and* heavy self-service BI
Existing investment	Greenfield	One platform in production	Two in production with real expertise in both
Data gravity	All data can live in one place	Some external sources	Large datasets that genuinely cannot move

0–2 · Single platform
You do not have a multi-engine problem yet.

3–4 · Single, open formats
Stay on Delta or Iceberg so the door stays open. Revisit in a year. The optionality is the point, not the second engine.

5–6 · Multi-engine
Justified, and your gates say you can operate it.

3 Where sovereignty fits

A sovereign-cloud requirement makes multi-engine **harder**, not more attractive. Treat it as a reason to raise the bar on the gates, never as a signal favouring more engines:

- **Feature parity lags.** Azure Government regions do not currently support Databricks SQL or Unity Catalog; Databricks' own November 2025 announcement targets the fuller platform for 2026.
- **Regional co-location is a real constraint.** Fabric for GCC High lists US Gov Virginia and US Gov Texas; Databricks in Gov is Arizona and Virginia. US Gov Virginia is the only region where the interim shortcut pattern can be co-located.
- **The AI layer is not there yet.** Copilot and data agents are absent from the GCC High preview feature table, so an architecture whose value depends on them does not extend into that environment.
- **Scope.** The GCC High preview covers GCC High, not DoD/IL5. Do not read one as the other.

4 The seam to pressure-test before go-live

Unity Catalog access controls do not apply to Fabric users consuming mirrored data. Queries run under the credentials of whoever configured the mirroring connection, so row- and column-level rules must be re-implemented in OneLake security — generally available since 4 May 2026. Confirm two things for your own tenant: the status for Mirrored Azure Databricks Catalog items specifically, and that GCC High preview still lacks Spark support for OneLake security.

Sources. OneLake security GA — community.fabric.microsoft.com/t5/Fabric-Updates-Blog/OneLake-security-Generally-Available/ba-p/5171978
OneLake table, column and row-level security — learn.microsoft.com/en-us/fabric/onelake/security/table-column-row-security
Mirroring from Azure Databricks — learn.microsoft.com/en-us/fabric/database/mirrored-database/azure-databricks
Azure support plans and severity — azure.microsoft.com/support/plans/response

Sovereign-cloud feature tables move quarterly. Verify anything time-sensitive against the source before planning around it — that is why this document is dated.