Fabric Architecture Decision Framework

5 Questions to Choose the Right Pattern

Matthias Falland — Microsoft Data Platform MVP

FabCon Atlanta 2026

1   How to Use This Framework

You don't need to read 500 pages of documentation. Answer five questions, and this framework maps you to the right architecture pattern — the one that minimizes refactoring six months from now.

Start simple. Every pattern below was battle-tested across 20+ real-world Fabric implementations. Begin with the simplest pattern that fits, and evolve as your needs grow.
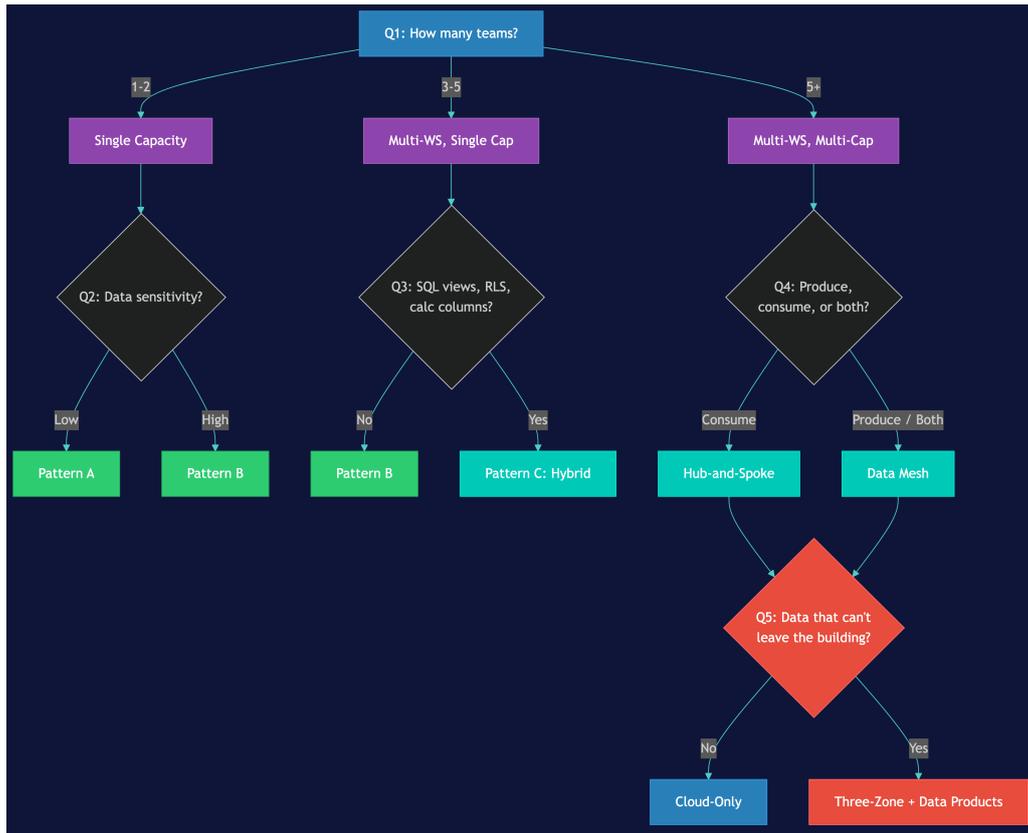
1   How to Use This Framework

Figure 1: The 5-Question Architecture Decision Framework

2   The 5-Question Decision Tree

2.1   Q1: How many teams will use Fabric?

This determines your **capacity and workspace topology**.

- **1–2 teams** → Single Capacity, few workspaces
- **3–5 teams** → Multiple workspaces, single capacity with resource governance
- **5+ teams** → Multiple workspaces, multiple capacities, Fabric Domains for organizational structure

2.2   Q2: What's your data sensitivity level?

This determines whether you can **co-locate data in a single Lakehouse**.

- **Low sensitivity** → **Pattern A** — Single Lakehouse with all medallion layers
- **High sensitivity** → **Pattern B** — Separate Lakehouses per zone or team, with OneLake shortcuts for controlled sharing

2.3   Q3: Do you need SQL views, RLS, or calculated columns?

This determines whether you need a **Warehouse in your Gold layer**.

- **No** → Stay with **Pattern B** — Lakehouse-only, Direct Lake semantic models
- **Yes** → **Pattern C** — Hybrid architecture with Lakehouse (Bronze/Silver) + Warehouse (Gold) for SQL-native features

2.4   Q4: Do teams produce data, consume it, or both?

This determines your **cross-workspace sharing topology**.

- **Mostly consume** → **Hub-and-Spoke** — Central platform team owns the hub, domain teams consume via shortcuts
- **Produce and/or both** → **Data Mesh** — Each domain owns its data products, shares via inbox/outbox pattern

2.5   Q5: Is there data that can't leave the building?

This determines whether you need **zone-based architecture**.

- **No** → Cloud-only deployment, standard patterns apply
- **Yes** → **Three-Zone Architecture** — Protected zone (on-premises), Hybrid zone (controlled gateway), Cloud zone (Fabric). Data Products serve as border control between zones.
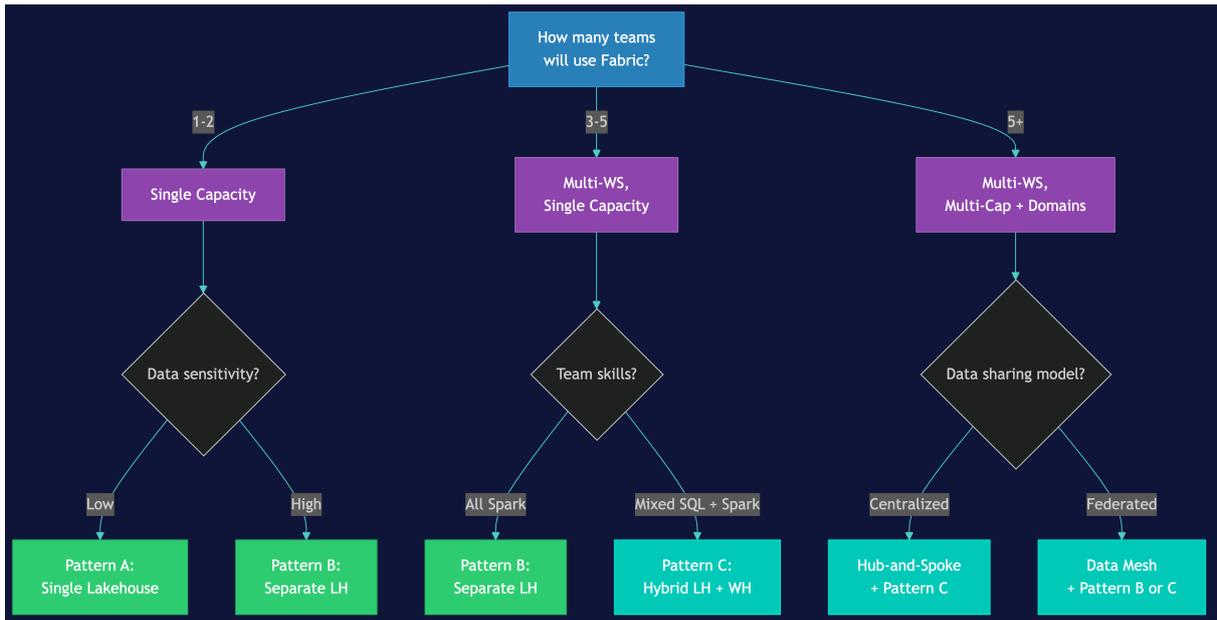
Figure 2: Original Decision Tree

3 Architecture Patterns at a Glance

3.1 Pattern A: Single Lakehouse

Best for: Small teams, low data sensitivity, getting started fast.

One Lakehouse holds Bronze, Silver, and Gold layers as folders. Simple, minimal over-head. The SQL Analytics Endpoint provides read-only SQL access to Gold.

Outgrow signal: When you need RLS, SQL views, or different teams need different refresh schedules — move to Pattern B or C.

3.2 Pattern B: Separate Lakehouses

Best for: Multiple teams, data isolation requirements, all-Spark teams.

Each medallion layer (or each team) gets its own Lakehouse. OneLake shortcuts connect them without copying data. Each Lakehouse has its own SQL Analytics Endpoint and default semantic model.

3.3 Pattern C: Hybrid Lakehouse + Warehouse

Best for: Mixed SQL and Spark teams, enterprise reporting with RLS, calculated columns, SQL views.

Lakehouse handles Bronze and Silver (Spark-based ingestion and transformation). Warehouse handles Gold (SQL-native features: views, RLS, stored procedures). This is the most common enterprise pattern.

Why hybrid? The SQL Analytics Endpoint on Lakehouse is read-only by design — Delta Lake uses optimistic concurrency managed by Spark. Allowing direct SQL

writes would risk transaction conflicts. The Warehouse gives you full SQL write capabilities on the Gold layer.

3.4   Hub-and-Spoke

Best for: Central platform teams serving multiple consuming domains.

A central Hub workspace owns shared data assets (reference data, master data, curated Gold layers). Spoke workspaces consume via shortcuts. The platform team controls quality; domain teams control their analytics.

3.5   Data Mesh

Best for: Organizations where domains both produce and consume data.

Each domain owns its Fabric workspace(s) and publishes Data Products. The inbox/outbox pattern keeps boundaries clean: producers write to their outbox, consumers shortcut from producers' outbox to their inbox.

3.6   Three-Zone Architecture

Best for: Regulated industries, data sovereignty requirements, hybrid cloud.

Three zones with clear boundaries: **Protected** (on-premises, never leaves), **Hybrid** (controlled transfer via gateway), **Cloud** (full Fabric). Data Products define the contracts between zones.

4   Anti-Patterns to Avoid

"One Workspace to Rule Them All" — Putting everything in a single workspace. Works for a POC, breaks at scale. No isolation, no independent deployments, no capacity governance per team.

"Skipping the Medallion" — Loading raw data directly into Gold. When source schemas change (and they will), everything downstream breaks. Bronze absorbs the chaos, Silver normalizes it, Gold serves the business.

"Copy Instead of Shortcut" — Duplicating data across workspaces using pipelines. OneLake shortcuts provide zero-copy access with security inheritance. Copies create drift, waste storage, and multiply refresh failures.

5   Resources

Session Links & Downloads:
the-trusted-advisor.com/events/fabcon-atlanta-2026

Official Documentation:

- Fabric deployment patterns (Azure Architecture Center)
- Medallion lakehouse architecture
- Lakehouse vs Warehouse decision guide
- OneLake shortcuts
- Fabric site reliability & enterprise guidance

Weekly Deep Dives:
Fabric Friday on YouTube — New episode every Friday
Fabric Friday Playlist

Connect:
LinkedIn — Matthias Falland

Weekly Deep Dives: Fabric Friday on
YouTube | youtube.com/@The-Trusted-Advisor
Playlist: youtube.com/playlist?list=PLfxrgmYYKLDYzoq27ykAihFqPR69RczKi
Connect: linkedin.com/in/matthias-falland