

Fabric - Capacity design and settings

 **SYSM-363** - Jira project doesn't exist or you don't have permission to view it.

 **SYSM-358** - Jira project doesn't exist or you don't have permission to view it.

 This page defines the **capacity-level configuration options** that must be evaluated for our Microsoft Fabric platform, with a focus on:

- production stability
- workload isolation
- controlled scaling
- governance of shared resources
- independence of the Data Platform Core workspace

Our target operating model uses Fabric primarily as a **data storage and exposure platform** based on **Lakehouse and Warehouse**, serving both BI consumption and external data exposure.

INFO

Fabric capacity configuration is a platform governance topic, not only an infrastructure topic.

RECOMMENDATION

The primary control for protecting Data Platform Core is capacity isolation.

WARNING

Shared capacity between Data Platform Core and Domain production creates shared operational risk.

DECISION

A dedicated capacity for Data Platform Core is the recommended baseline for this architecture.

Version	Date	Description	Contributor
V0.1	15 Apr 2026	Initial document	COLOMBANI Théo
V0.2	20 Apr 2026	Updated with schema proposal and checklist	COLOMBANI Théo

- [Key messages](#)
- [What we should implement](#)
 - [Recommended target model](#)
 - [Capacity design principles](#)
- [Proposed capacity design](#)
- [Decision guide](#)
- [Checklist](#)
- [Recommended configuration matrix](#)
 - [Recommended target model](#)
- [Detailed design sections](#)
 - [Workspace-to-capacity assignment](#)
 - [Capacity administration and reassignment governance](#)
 - [Capacity sizing and scaling](#)
 - [Monitoring and operational visibility](#)
 - [Disaster recovery](#)
 - [Data Engineering and Spark-related settings](#)
 - [Architecture conclusion](#)

Key messages

- **Capacity design must be driven by isolation first, then by optimization.**
 - **The Data Platform Core workspace must remain operational independently from Domain workspaces, including when Domain workloads generate higher or less predictable compute consumption.**

- **Capacity is an architecture boundary**, not only a billing or administration object.
- The **Data Platform Core workspace** should not share the same production risk envelope as Domain workloads.
- **Non-production must be isolated** from production capacities.
- Capacity governance must cover both **technical setup** and **operating model**.

What we should implement

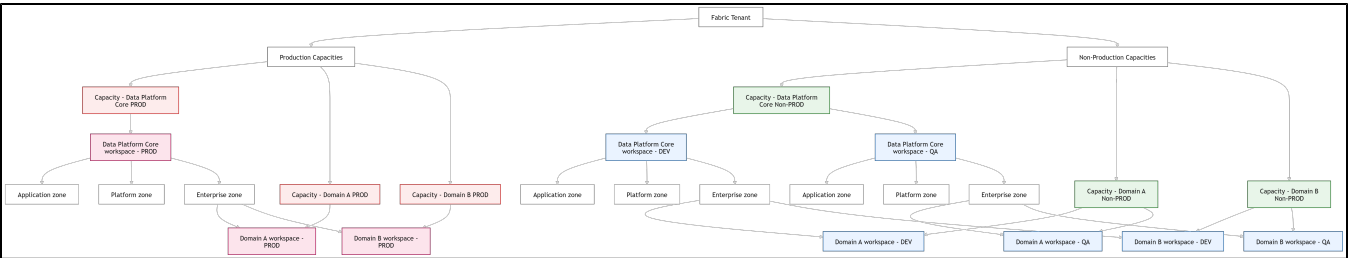
Recommended target model

- **one dedicated Fabric capacity for Data Platform Core**
- **one separate Fabric capacity for Domain production**
- **one separate non-production capacity**
- **centralized control of workspace assignment**
- **formal governance for capacity admin and reassignment rights**
- **standardized monitoring and review cadence**
- **explicit disaster recovery assessment for Data Platform Core**

Capacity design principles

- isolate critical and variable workloads
- avoid shared production capacity between Core and Domain if operational independence is required
- keep non-production outside production capacities
- define ownership and review rules for every production capacity

Proposed capacity design



Decision guide

Decision area	Use this approach when	Recommended decision
Dedicated capacity for Data Platform Core	bronze and silver are production-critical; downstream BI or external exposure depends on them; Domain workloads are more variable than Core workloads; Core continuity is a priority	assign Data Platform Core to a dedicated capacity
Separate Domain production capacity	multiple Domain workspaces coexist; Domain workloads may create contention; business-facing usage is less predictable; Domain growth should not affect Core operations	assign Domain production to a separate capacity
Separate non-production capacity	development and testing are active; experimentation may generate compute spikes; production stability must be protected from non-production activity	keep non-production on a separate capacity
Architecture review required	Core and Domain are still planned on the same capacity; workspace reassignment is not tightly governed; production and non-production still share capacity; capacity sizing issues become recurrent	escalate to architecture review

Checklist

- ☐ Has a dedicated capacity been confirmed for **Data Platform Core**?
- ☐ Has **Domain production** been isolated from **Data Platform Core**?

- ☐ Has **non-production** been separated from production capacities?
- ☐ Have **capacity admin** roles been limited to the central platform team?
- ☐ Have **workspace reassignment rights** been formally governed?
- ☐ Has a **monitoring owner** been assigned for each production capacity?
- ☐ Has **disaster recovery** been explicitly assessed for Data Platform Core?
- ☐ Have **Spark-related settings** been reviewed, if applicable?
- ☐ Has the **target capacity model** been approved as part of platform governance?

Recommended configuration matrix

Setting	Data Platform Core	Domain Production	Non-Production	Recommendation
Dedicated capacity	Yes	Preferred	Separate	Mandatory for Data Platform Core
Shared with Data Platform Core	NA	No	No	Not allowed
Workspace reassignment rights	Very restricted	Restricted	Controlled	Govern centrally
Monitoring	Mandatory	Mandatory	Recommended	Standard operating baseline
DR assessment	Mandatory	Case by case	Not priority	Explicit decision required
Spark governance	Case by case	Case by case	Flexible	Only where relevant
Scaling review cadence	Regular	Regular	Periodic	Metrics-driven

Recommended target model



Recommendation

Do not place Data Platform Core production and Domain production on the same capacity if Data Platform Core must remain operational independently.

Capacity	Scope	Used for
Capacity A — Data Platform Core	Central platform production capacity	Core production ingestion, preparation, and exposure foundations
Capacity B — Domain Production	Domain production capacities	- Domain gold workspaces - business-facing data products - BI-oriented workloads - potentially more variable usage patterns
Capacity C — Non-Production	Shared or segmented non-production capacities	- Development - testing - experimentation - validation before production promotion

Detailed design sections

Workspace-to-capacity assignment

Key message

Workspace assignment is the primary mechanism used to guarantee production isolation and operational independence. Fabric capacity settings let admins manage assigned workspaces, and workspace reassignment directly affects how workloads share compute risk. ([Microsoft Learn](#))

Area	Summary
What it is	The assignment of workspaces to specific Fabric capacities. (Microsoft Learn)
Why it matters	This is the most important configuration decision because it determines which workloads share the same compute risk domain. Capacity planning guidance frames capacity allocation as a governance decision, not only a technical sizing choice. (Microsoft Learn)
What to watch	If Core, Domain, and non-production workloads share the same capacity, they also share the same contention and throttling risk envelope. Capacity growth guidance explicitly recommends governance patterns adapted to centralized and decentralized models. (Microsoft Learn)
Recommended posture	Keep Data Platform Core on a dedicated capacity, separate Domain production where possible, and isolate non-production from all production capacities. (Microsoft Learn)

Recommendation block

- Assign **Data Platform Core** to a dedicated capacity.
- Assign **Domain production** to a separate capacity whenever possible.
- Keep **DEV / QA / other non-production** off production capacities.
- Avoid mixing critical platform workloads with variable business workloads.

Capacity administration and reassignment governance



Key message

A dedicated production capacity loses most of its value if workspace assignment is not tightly governed. Fabric allows reassignment through admin and workspace-level paths, so governance rules must be explicit. ([Microsoft Learn](#))

Area	Summary
What it is	The set of permissions allowing administrators to manage a capacity and move workspaces into or out of it. (Microsoft Learn)
Why it matters	Even with a good target architecture, weak governance can reintroduce risk if workspaces are moved without control. Workspace admins can also reassign workspaces in some cases, which increases the need for formal guardrails. (Microsoft Learn)
What to watch	A capacity model can look clean on paper but drift over time if reassignment rights are too broad. Governance guidance emphasizes strong controls when multiple teams share Fabric at scale. (Microsoft Learn)
Recommended posture	Restrict critical-capacity administration to the central platform team and formalize approval for any reassignment that affects the Core production perimeter. (Microsoft Learn)

Recommendation block

- Restrict **capacity admin** rights to the central Data Platform or IT team.
- Restrict workspace reassignment rights on critical capacities.
- Require formal approval for any workspace added to the **Data Platform Core** capacity.
- Prevent self-service reassignment into critical production capacity.

Capacity sizing and scaling



Key message

Data Platform Core capacity sizing must prioritize service continuity over cost minimization. Microsoft recommends estimating size from workload characteristics and validating with real usage in the Capacity Metrics App. ([Microsoft Learn](#))

Area	Summary
What it is	The sizing of Fabric capacity and the ability to adjust it as workload volume evolves. (Microsoft Learn)
Why it matters	Even a well-isolated architecture can fail if the capacity is persistently undersized. Strategic planning guidance recommends budgeting, scaling, and optimization as ongoing activities. (Microsoft Learn)

What to watch	Resizing should be based on observed patterns, not only on incident response. The Metrics App is the main evidence source to understand CU usage, peaks, and the item types driving load. (Microsoft Learn)
Recommended posture	Size Data Platform Core with operational headroom. Review Domain production more frequently because its usage is more variable and business-facing. (Microsoft Learn)

Recommendation block

- Size **Data Platform Core** for continuity first.
- Review **Domain production** more frequently.
- Use monitored usage trends to drive scaling decisions.
- Avoid reactive resizing without understanding the underlying workload pattern.

Monitoring and operational visibility



Key message

Capacity monitoring must be part of normal run operations, not only incident management. The Fabric Capacity Metrics App is designed to help admins monitor health, top consumers, compute usage, and issues such as throttling or query rejections. ([Microsoft Learn](#))

Area	Summary
What it is	Monitoring of capacity usage, saturation patterns, top consumers, and operational degradation signals. (Microsoft Learn)
Why it matters	Capacity governance is only effective if usage and saturation can be observed and acted upon. Microsoft recommends using the Metrics App to identify top consumers and optimize before throttling becomes recurrent. (Microsoft Learn)
What to watch	Monitoring data should be interpreted operationally: recurring peaks, saturation patterns, top-consuming items, and correlations with refresh, ingestion, or user activity. The app also has refresh latency, so near-real-time assumptions should be avoided. (Microsoft Learn)
Recommended posture	Every production capacity should have a monitoring owner, review cadence, threshold model, and escalation path. (Microsoft Learn)

Recommendation block

For each production capacity, define:

- **monitoring owner**
- **review cadence**
- **alert thresholds**
- **escalation path**
- **expected remediation actions**

Minimum baseline:

- monitor recurring peaks
- identify top consuming workspaces and items
- review saturation or degradation patterns
- correlate issues with refresh, ingestion, or usage spikes

Disaster recovery



Key message

For Data Platform Core, disaster recovery should never be left undocumented. Fabric capacity settings include disaster recovery controls, and Microsoft's recovery guidance makes clear that recovery planning must be explicit. ([Microsoft Learn](#))

Area	Summary
What it is	The capacity-level disaster recovery posture associated with production data continuity. Capacity settings include disaster recovery options and related status information. (Microsoft Learn)
Why it matters	The Data Platform Core workspace supports bronze and silver foundations, making it a central dependency for downstream exposure. In that model, DR is an architecture topic, not just an ops topic. This last point is an inference from your target design, supported by Microsoft's capacity governance framing. (Microsoft Learn)
What to watch	DR should be documented as an explicit decision: enabled or not enabled, with assumptions and limitations clearly stated. (Microsoft Learn)

Recommended posture	Assess DR first for Data Platform Core , then extend case by case to Domain production depending on criticality. This prioritization is a design recommendation based on your architecture. (Microsoft Learn)
----------------------------	--

Recommendation block

- Perform an explicit DR assessment for **Data Platform Core**.
- Document whether DR is enabled or not.
- Document recovery assumptions and limitations.
- Ensure DR is an explicit architecture decision, not an omission.

Data Engineering and Spark-related settings

 **Key message**
This is a secondary topic unless Spark becomes a major production dependency. Fabric capacity admins can manage Data Engineering and Data Science settings, including workspace-level compute, runtime defaults, and Spark properties. ([Microsoft Learn](#))

Area	Summary
What it is	Capacity-level settings related to Data Engineering and Data Science, including Spark governance options. (Microsoft Learn)
Why it matters	These settings become relevant when Spark-based processing is materially used in the platform. Microsoft explicitly positions them as admin-governed capacity settings. (Microsoft Learn)
What to watch	If Spark usage grows without governance, compute sprawl can become harder to control. Spark planning guidance also treats development and production needs differently. (Microsoft Learn)
Recommended posture	Keep Spark governance centralized and document Spark rules separately if Spark is not a central workload in the platform. The “secondary topic” positioning is a design recommendation for your model. (Microsoft Learn)

Recommendation block

- Keep Spark governance centralized.
- Avoid uncontrolled compute sprawl.
- Document Spark rules separately if Spark is not a central workload in the platform.

Architecture conclusion

This is the most coherent model for a Fabric platform used primarily as a storage and exposure layer, where the **Data Platform Core workspace** must remain stable independently from Domain activity.