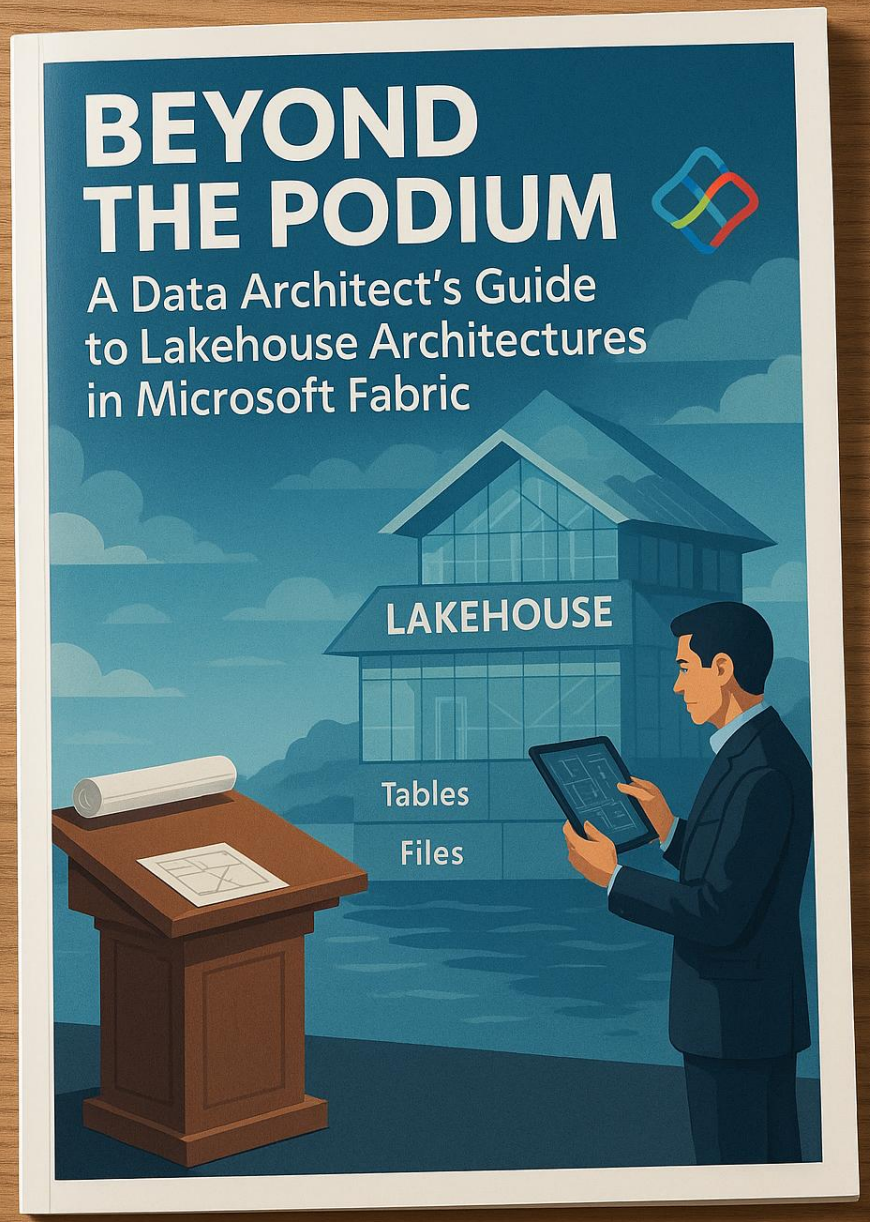


SQL Day

Matthias Nohl | b.telligent

Tim Spannagel | b.telligent

17th edition
SQLDay Conference

12-14 May 2025, WROCŁAW + ONLINE

SQL Day

Platinum sponsors

DATUMO

Woodler

Gold sponsors

lingaro

VOLVO

TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk

software one

Dataedo

Silver sponsors

Capgemini

DATA RISE LAB

elitmind

C&F

kursy FABRIC

Your Speakers

SQL Day

Matthias Nohl

Strategic Partner Manager &
Management Consultant

b.telligent
smart data. smart decisions.

linkedin.com/in/matthiasnohl

@mnohlimits

Tim Spannagel

Team Lead &
Principal Consultant

linkedin.com/in/tim-spannagel

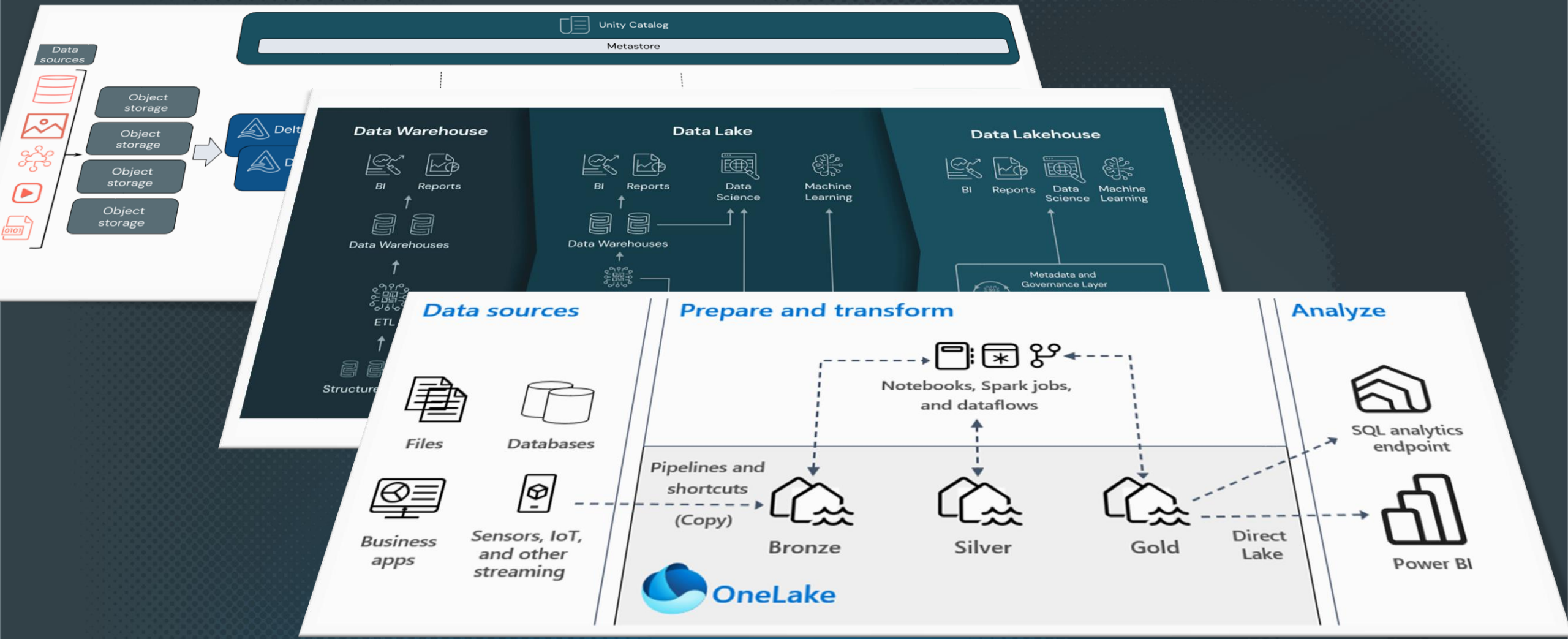
@TSpannagel

Agenda

- Introduction “Lakehouse Architectures”
- Drivers for Architecture
- Adoption to Microsoft Fabric
- Q&A

Introduction
“Lakehouse Architectures”

Introduction “Lakehouse Architectures”



What is a data lakehouse? - Azure Databricks | Microsoft Learn
What is a Data Lakehouse? | Databricks
Implement medallion lakehouse architecture in Fabric - Microsoft Fabric | Microsoft Learn

Introduction “Lakehouse Architectures”

“The medallion lakehouse architecture, commonly known as medallion architecture, is a design pattern that's used by organizations to logically organize data in a lakehouse. ***It's the recommended design approach for Fabric****. Since OneLake is the data lake for Fabric, medallion architecture is implemented by creating lakehouses in OneLake.”*

Implement medallion lakehouse architecture in Fabric - Microsoft Fabric | Microsoft Learn

Introduction “Lakehouse Architectures”

“Even though adopting a Lakehouse architecture is said to be the solution, there are still countless questions that remain unanswered.”

Every Data Architect

Drivers for Architecture

Drivers for Architecture

Organizational Considerations

- Business Needs & Goals
- Compliance and Governance
- Skillset
- Budget/Cost Efficiency

Drivers for Architecture

SQL Day

Technical Considerations

- Security
- Scalability
- Performance
- Integration
- Automation
- Platform Limitations

Adoption to Microsoft Fabric

Drivers for Architecture

Organizational Considerations

Business Needs & Goals	Compliance and Governance	Skillset	Budget Cost Efficiency
✓ Unified Data Platform ✓ Business-Aligned Governance via Purview ✓ Data-Driven Decision Making via Power BI ✓ M365 Integration	✓ Data Classification ✓ Information Protection ✓ Data Loss Prevention ✓ Data sovereignty ✓ Ownership	✓ Low-Code / No-Code Tools for Business Users ✓ Pro-Code Tools for Data Professionals ✓ Unified Experience Across Roles ✓ Fabric Community	✓ Cost Management ✓ Pay-as-you-go ✓ Resource Optimization ✓ Consolidation

Drivers for Architecture

Technical Considerations

Security	Scalability	Performance	Integration	Automation	Platform Limitations
✓ Workspace Design ✓ One Security / Data Security ✓ Data Encryption ✓ Private Links ✓ Conditional Access ✓ Trusted Workspace access ✓ Managed Private Endpoints ✓ Managed virtual networks ✓ Azure Service Tags ✓ IP Allowlists	✓ Workspace Design ✓ Dynamic Scaling ✓ Bursting and Smoothing ✓ Capacity Monitoring ✓ Availability zones	✓ Workspace Design ✓ Capacity Management ✓ Autoscale Billing for Spark	✓ Workspace Design ✓ Dataflows ✓ Notebook ✓ Data Pipelines ✓ T-SQL ✓ Mirroring ✓ Shortcuts ✓ Dataverse	✓ Workspace Design ✓ CI/CD ✓ Rest APIs ✓ Deployment Pipelines ✓ fabric-cicd ✓ Coding Conventions ✓ Naming Conventions ✓ DWH Automation / dbt?	✓ Private Link support at a workspace level ✓ Data exfiltration protection for Spark ✓ OneLake encryption with customer-managed keys ✓ Identity columns ✓ Eventstream (Custom SQL Code) ✓ Dataflows CI/CD 😂

Everything clear?

SQL Day

And now what?

How do we start?

Workspace Design is crucial 🤩

Workspace Types – Which type do we need?


Central

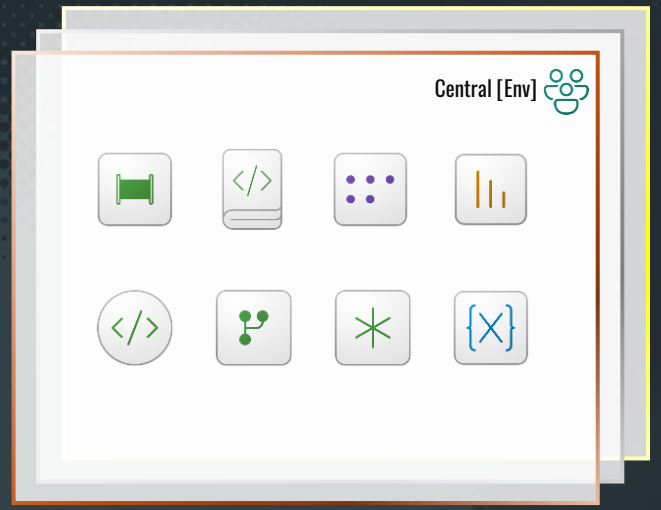

Developer


Dataspace


Domain/Department

Workspace Types: Central

- **Purpose**
  - Provide access to central code artifacts
- **Access**
  - Central data engineering team
    - Contributor or viewer in [DEV]
    - Viewer in [PROD]
- **Item Types**
  - Code artifacts (e.g. data pipelines, notebookes, semantic models, reports, environments, variable libraries,...)
- **CI/CD**
  - No GIT integration
  - Deployed from main or environment branch via fabric-cicd

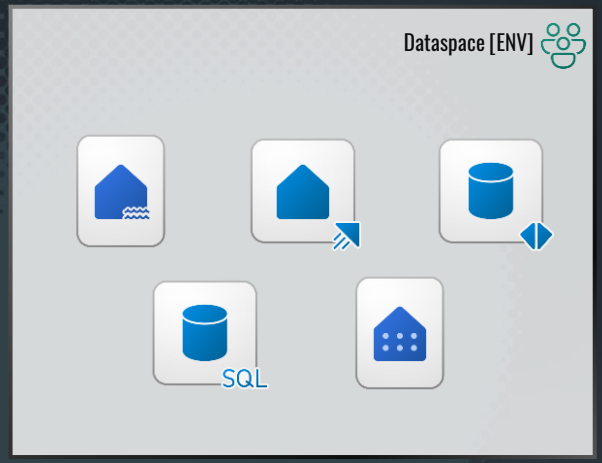


Workspace Types: Developer

- **Purpose**
  - Personal workspace for developers working on features and bugfixes
- **Access**
  - Individual developers (Admin)
- **Item Types**
  - Code artifacts (e.g. data pipelines, notebookes), semantic models, reports, environments, ...)
- **CI/CD**
  - Connected to current feature branch.
  - Merged back via Pull Request into main/environmental branch

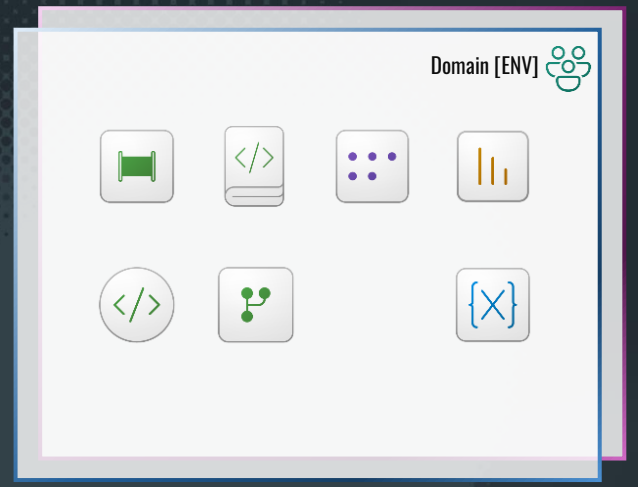
Workspace Types: Dataspace

- **Purpose**
  - Provide access to data
  - Avoid integration issues based on Git conflicts
- **Access**
  - Central data engineering team (Contributor)
  - Analysts (Viewer + Item-level permissions)
- **Item Types**
  - Data artifacts (e.g. Lakehouse, Warehouse, Eventhouse, Mirrored DB, ...)
- **CI/CD**
  - Optional Git integration (e.g. if queries are stored)
  - Deployment via Fabric CLI or terraform



Workspace Types: Domain/Department

- **Purpose**
  - Owned by the Domain/Department. Working on specific reports, analytics
- **Access**
  - Analytics Engineers (Contributor)
- **Item Types**
  - Code artifacts (e.g. data pipelines, notebookes), semantic models, reports, environments
- **CI/CD**
  - Git integration optional, recommended; depends on the skill level
  - Provide trainings and processes for collaboration and deployment
  - Deployment via deployment pipelines or azure pipelines



Automation – digression

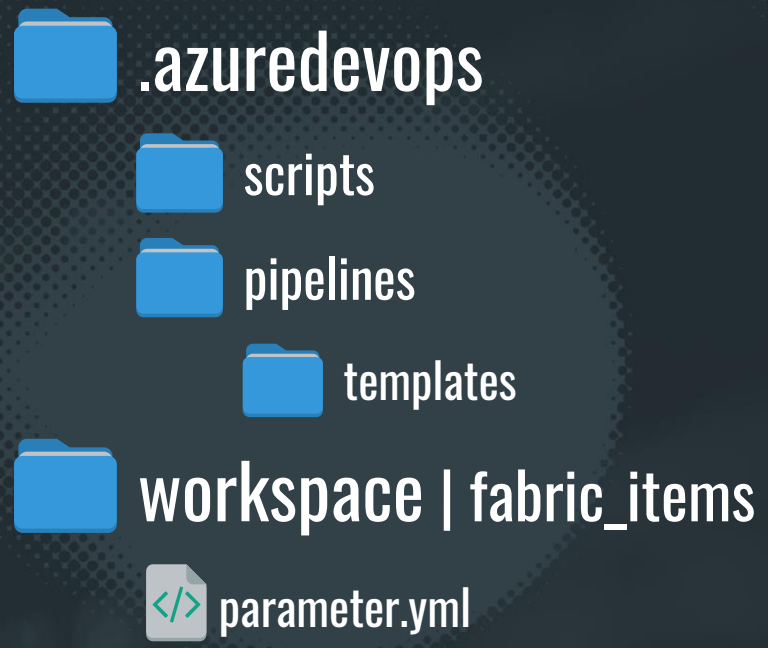
SQL Day

“No CI/CD, no effective automation in delivery workflows 😫“

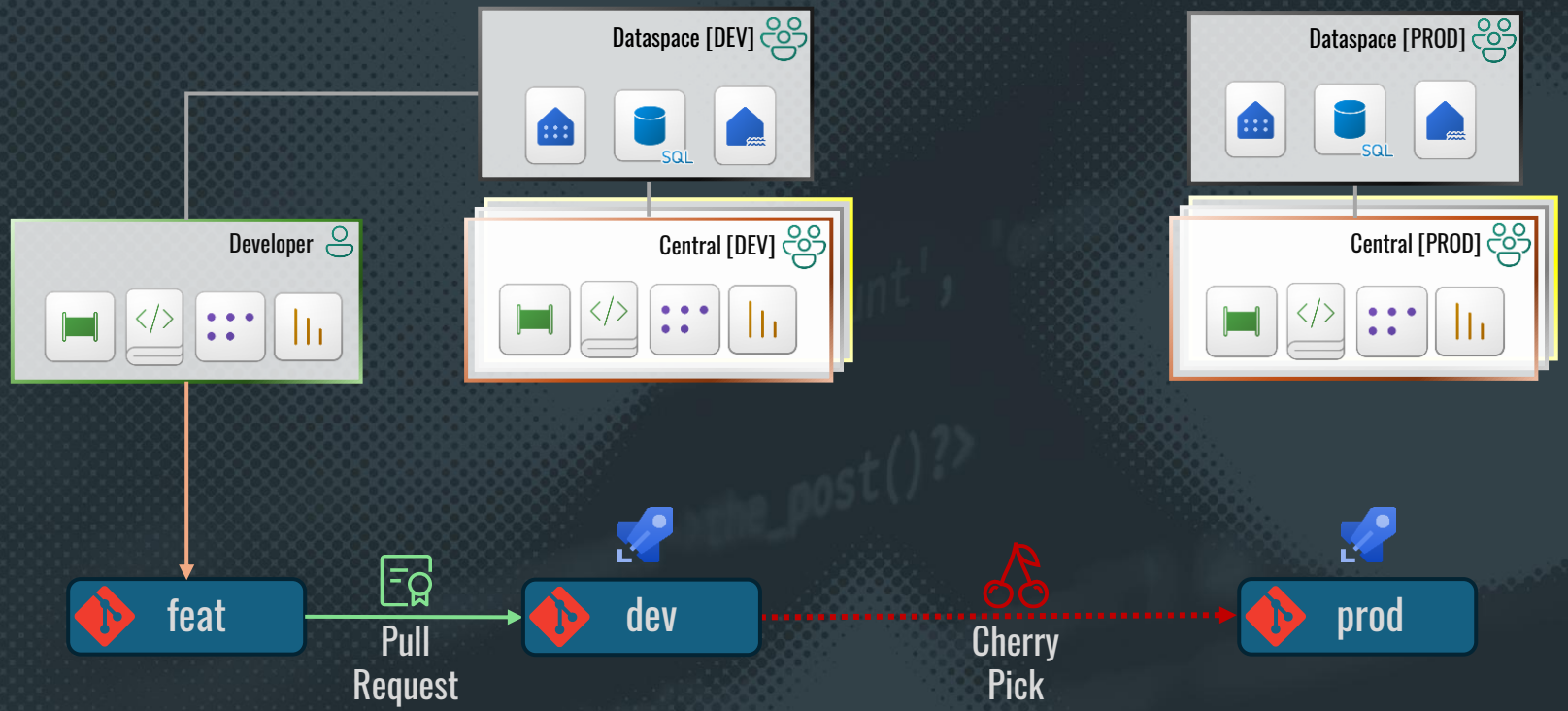
Hopefully Everybody

CI/CD & Repo Structure

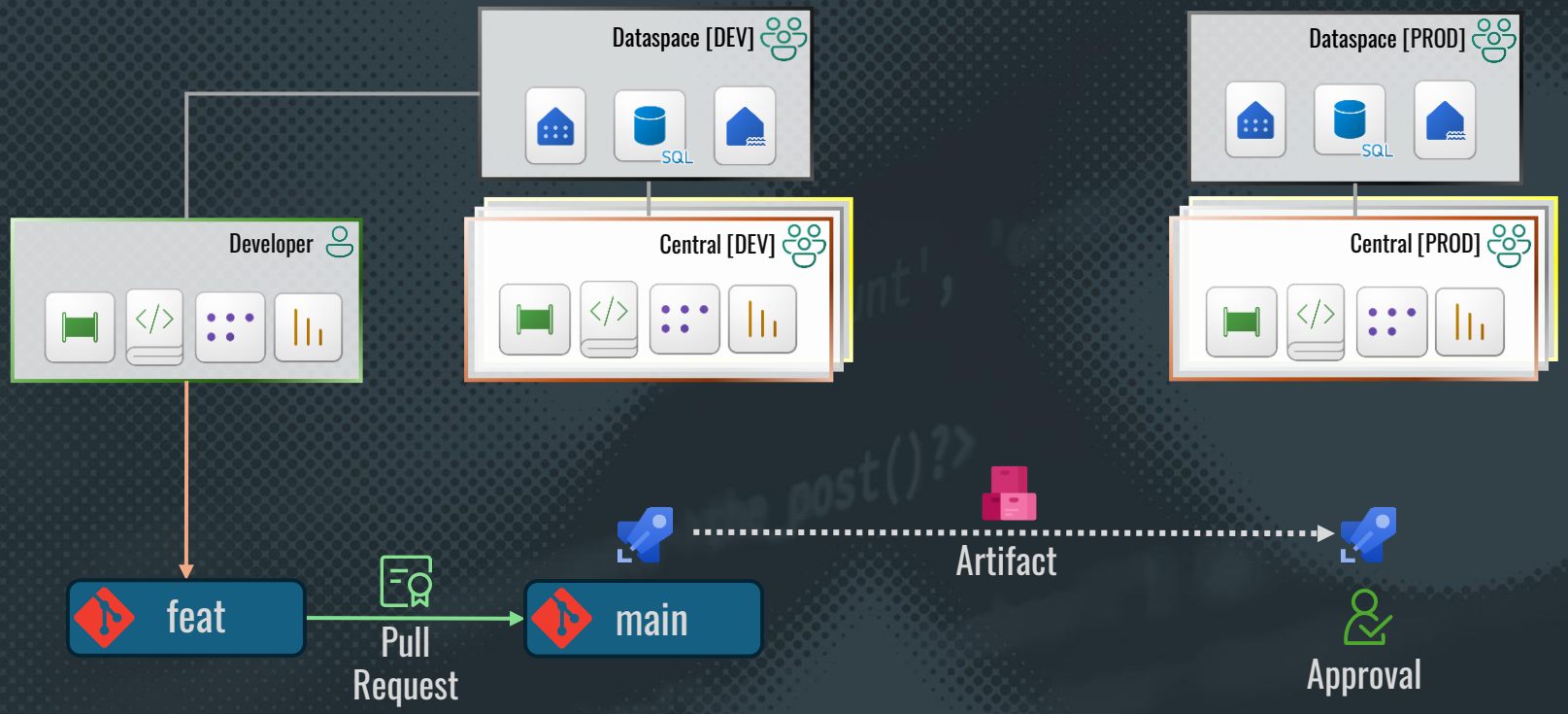
- **Platforms**:
  - Azure DevOps
  - GitHub
- **Tooling**:
  - fabric-cicd (GitHub)
  - Fabric tools (GitHub)
  - DacFx/SSDT (Learn)
  - Best Practice Analyzer (Website)
  - ~~Delta Kusto~~
- **Branching Strategies**:
  - Trunk-based
  - GitLab Flow



CI/CD: GitLab-Flow



CI/CD: Trunk-based

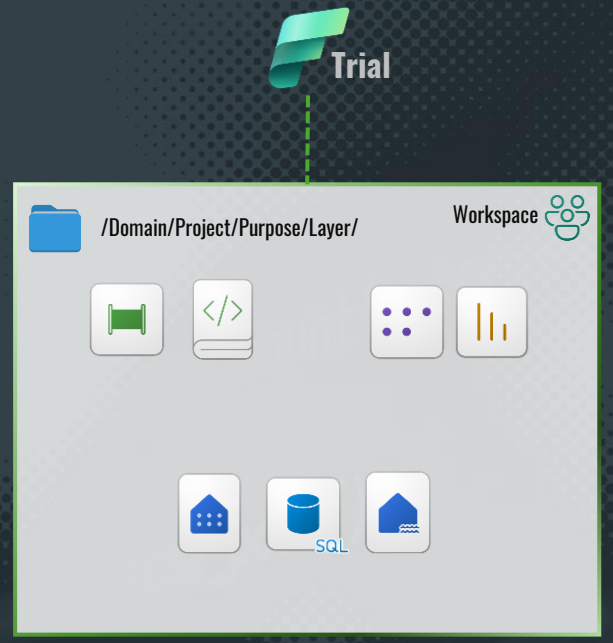


Automation – digression

SQL Day

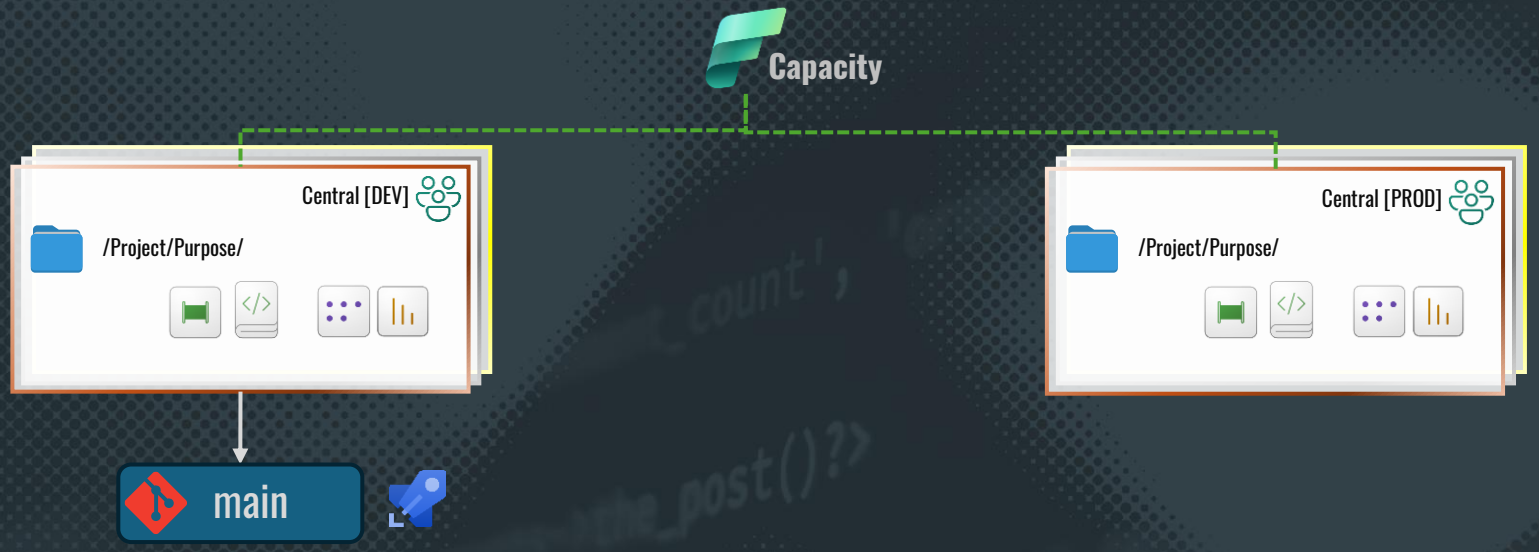
Back to the workspaces... 🏭

The Beginnings

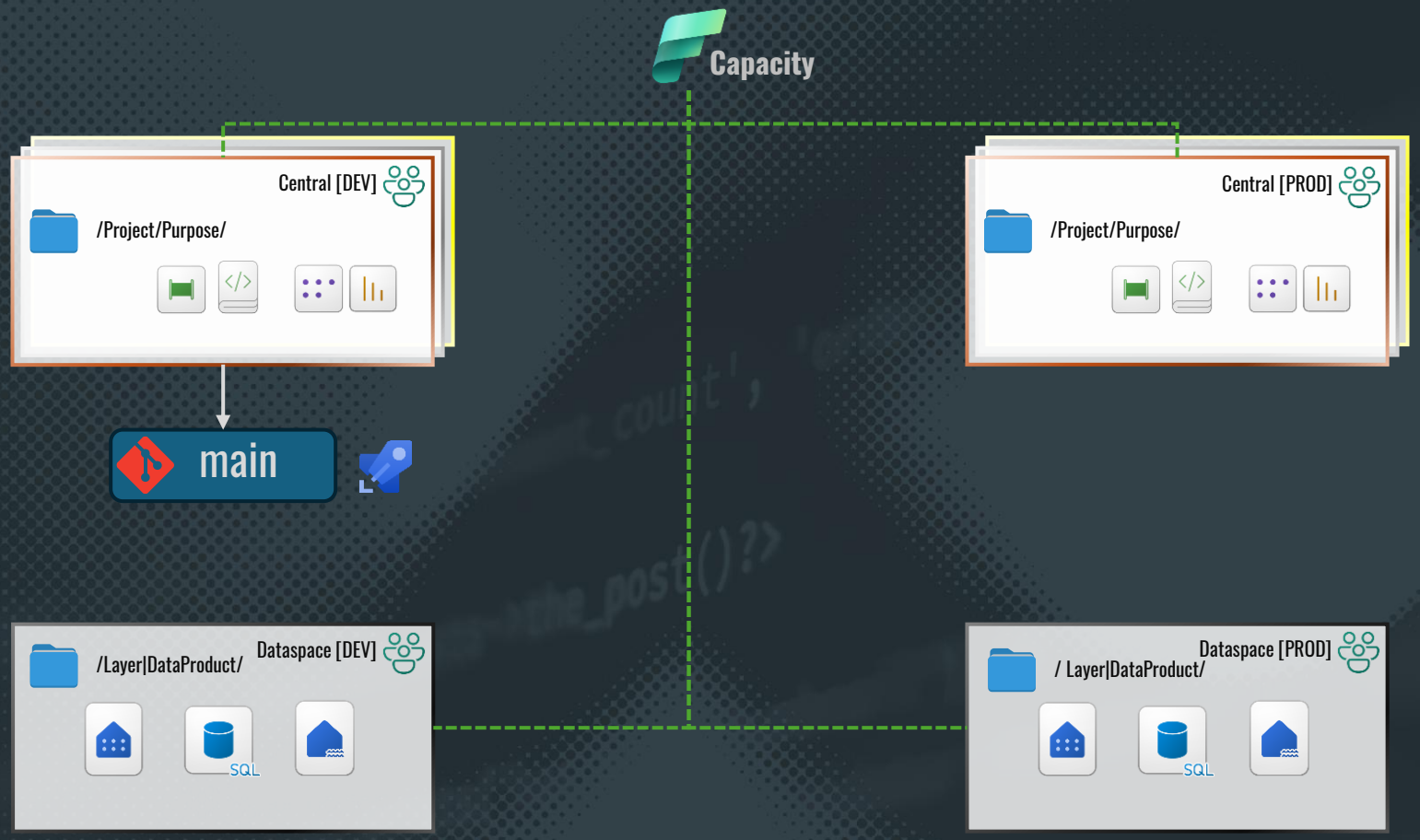


Small

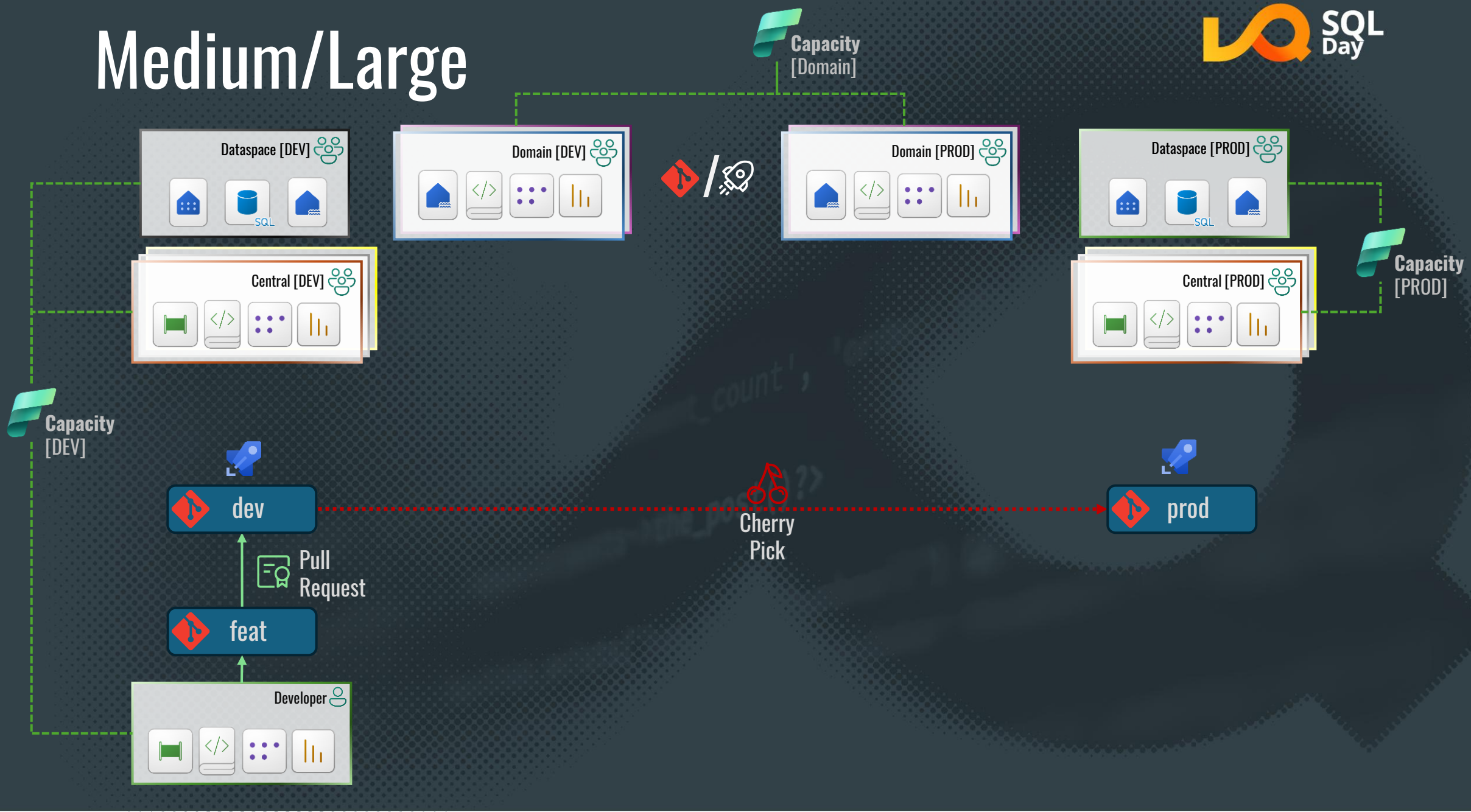
Small


Capacity
Central [DEV]
/Project/Purpose/
main
Central [PROD]
/Project/Purpose/

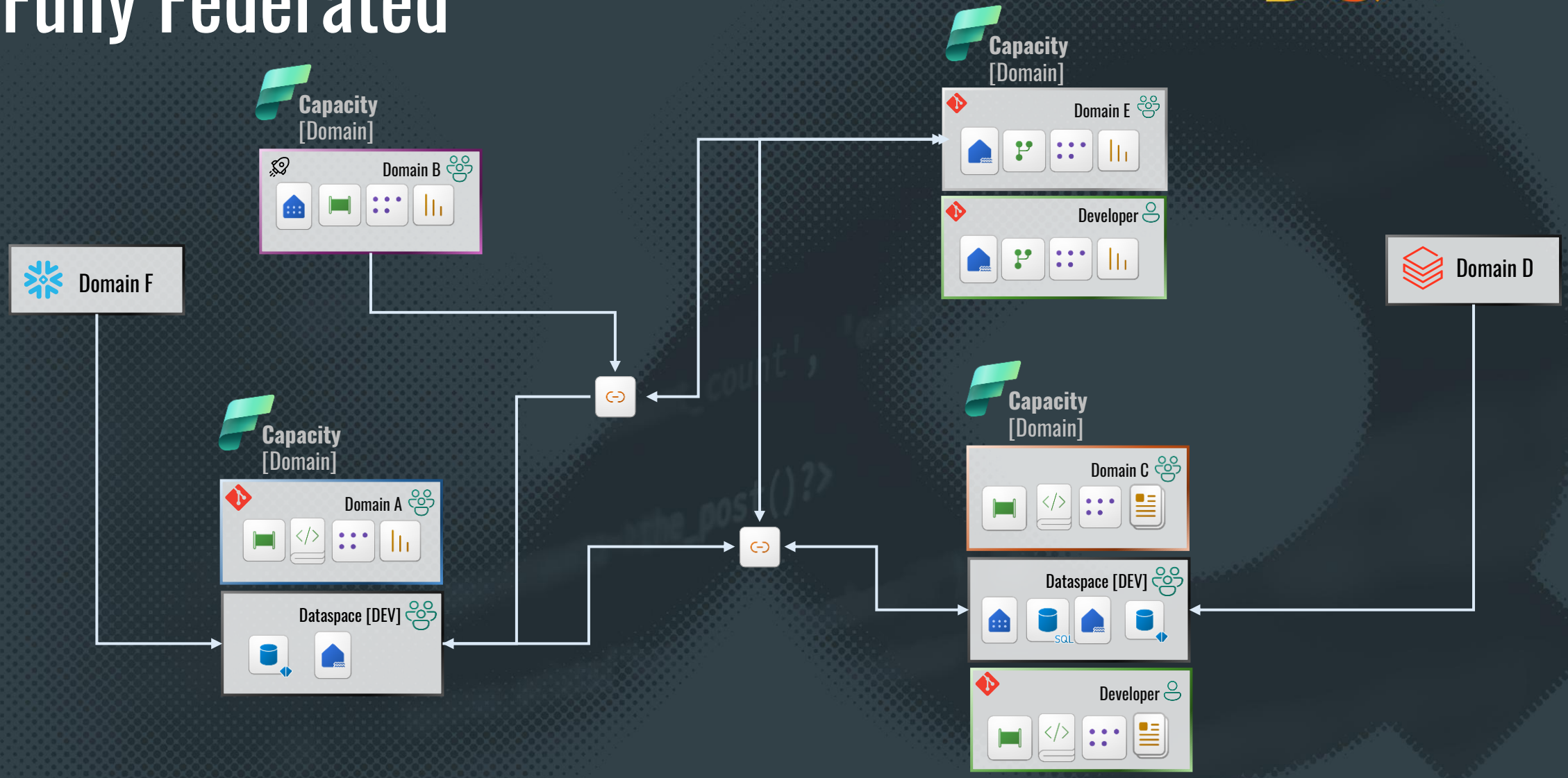
Small


Capacity
Central [DEV]
/Project/Purpose/
main
Central [PROD]
/Project/Purpose/
/Layer|DataProduct/
Dataspace [DEV]
SQL
Dataspace [PROD]
/ Layer|DataProduct/
SQL

Medium/Large



Fully Federated



Recap

SQL Day

How do we start?

Design the Workspace! 😈

Drivers for Architecture

Organizational Considerations

Business Needs & Goals	Compliance and Governance	Skillset	Budget Cost Efficiency
✓ Unified Data Platform ☑ ✓ Business-Aligned Governance via Purview ✓ Data-Driven Decision Making via Power BI ☑ ✓ M365 Integration	✓ Data Classification ✓ Information Protection ✓ Data Loss Prevention ✓ Data sovereignty ☑ ✓ Ownership ☑	✓ Low-Code / No-Code Tools for Business Users ☑ ✓ Pro-Code Tools for Data Professionals ☑ ✓ Unified Experience Across Roles ☑ ✓ Fabric Community ☑	✓ Cost Management ☑ ✓ Pay-as-you-go ☑ ✓ Resource Optimization ☑ ✓ Consolidation ☑

☑ addressed in the session

Drivers for Architecture

Technical Considerations

Security ☑	Scalability ☑	Performance ☑	Integration ☑	Automation ☑	Platform Limitations ☑

✓ Workspace Design
✓ One Security / Data Security
✓ Data Encryption
✓ Private Links
✓ Conditional Access
✓ Trusted Workspace access
✓ Managed Private Endpoints
✓ Managed virtual networks
✓ Azure Service Tags
✓ IP Allowlists
	✓ Workspace Design
	✓ Dynamic Scaling
	✓ Bursting and Smoothing
	✓ Capacity Monitoring
	✓ Availability zones
		✓ Workspace Design
		✓ Capacity Management
		✓ Autoscale Billing for Spark
			✓ Workspace Design
			✓ Dataflows
			✓ Notebook
			✓ Data Pipelines
			✓ T-SQL
			✓ Mirroring
			✓ Shortcuts
			✓ Dataverse
				✓ Workspace Design
				✓ CI/CD
				  ✓ Rest APIs
				  ✓ Deployment Pipelines
				  ✓ fabric-cicd
				✓ Coding Conventions
				✓ Naming Conventions
				✓ DWH Automation / dbt?
					✓ Private Link support at a workspace level
					✓ Data exfiltration protection for Spark
					✓ OneLake encryption with customer-managed keys
					✓ Identity columns
					✓ Eventstream (Custom SQL Code)
					✓ Dataflows CI/CD 😂

☑ addressed in the session

Please share feedback via Whova App! 😍

Dziękujemy
Thanks
Dankeschön

Data Community

Q&A