

fabric
FEBRUARY



End-to-End Data Engineering with Microsoft Fabric

Luke Moloney & Stijn Wynants

Stijn Wynants

Espresso Data Drinker

Mind The Facts



A passionate community advocate, international speaker, blogger, Fabric Espresso video co-creator.

Stijn helps organizations drive real impact through strategic, practical data and AI solutions.

stijn@mindthefacts.be 

Li/stijn-wynants-ba528660 

<https://mindthefacts.be> 

Luke Moloney

Wannabe Espresso Drinker

Microsoft



lumolone@microsoft.com 
[in/moloneyluke/](https://www.linkedin.com/in/moloneyluke/) 
lukemoloney.net? 

Work for Microsoft with ISVs and Strategic Customers for Fabric, ex-CAT, Data Engineering/Consulting background

Data, Rugby and F1.

Agenda

Block 1 (08:30 – 10:00) Introduction, Mirroring & Data Pipelines

Block 2 (10:20 – 12:00) Spark

Block 3 (13:00 – 14:00) Warehouse & Streaming

Block 4 (14:20 – 15:30) Security & Monitoring

Block 5 (15:30 – 16:00) Q&A



Block 1: Mirroring

Getting data in

Block 1: Data Pipelines

Copying and transforming data with Pipelines

Data Pipelines in Microsoft Fabric

Copying, orchestrating, and transforming data end-to-end

- How Data Factory concepts live inside Fabric
- What's new, what's improved
- Real-world pipeline patterns



What are Data Pipelines in Fabric?

- Native orchestration engine built directly into OneLake
- Combines ADF's proven engine with Fabric's unified compute
- Pipeline activities trigger workloads across Fabric (Lakehouse, Warehouse, Spark, KQL, Notebooks, ML)
- Governance, lineage, and monitoring integrated across the workspace



Data Factory → Fabric: Concept Mapping

Activities & triggers → Pipeline activities & schedules

Integration Runtime → managed/serverless execution in Fabric

Linked services/credentials → workspace credentials & data gateways



Connectors & Ingestion



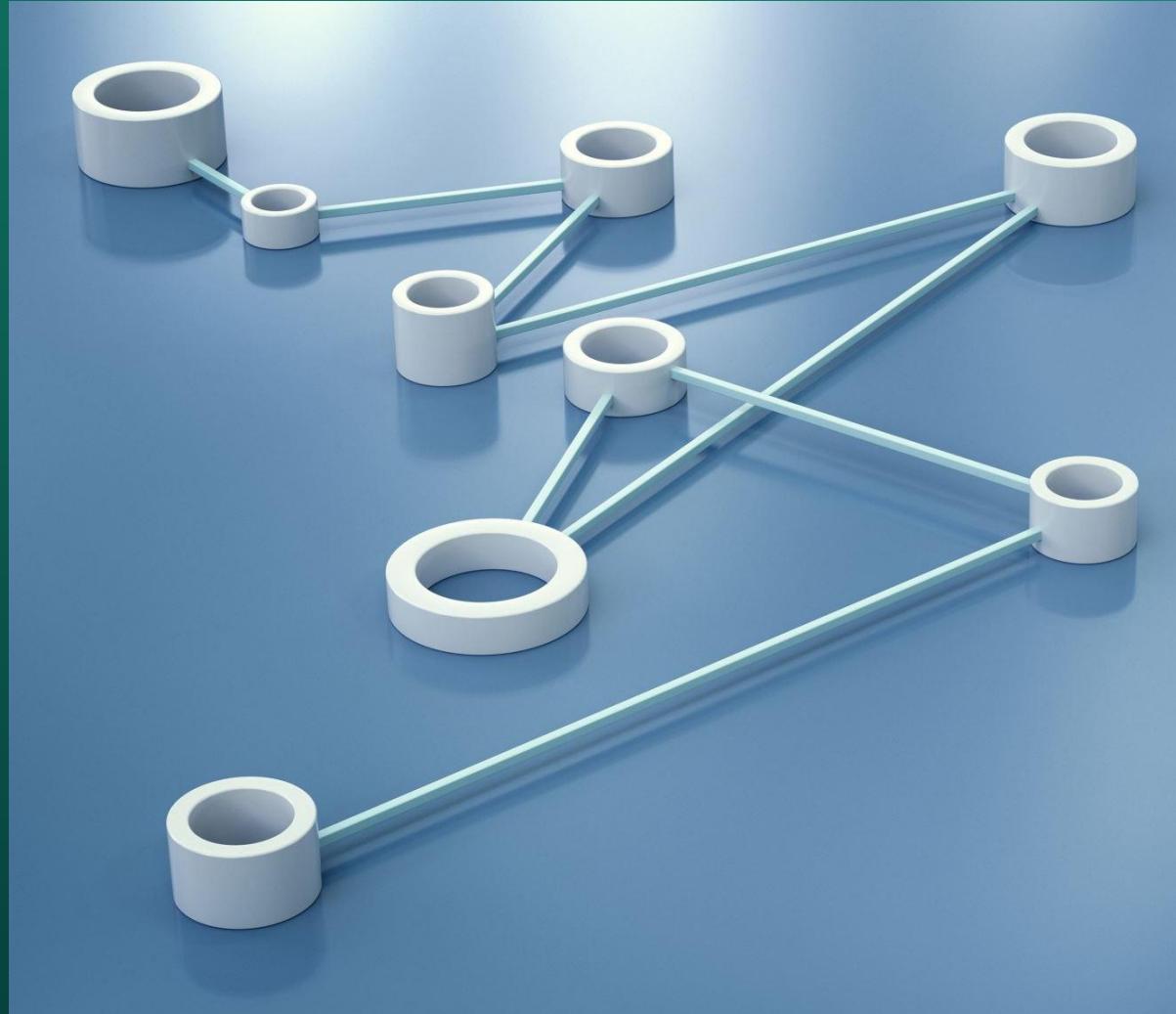
Broad connector catalog for DBs,
files, and SaaS

High-throughput writes to OneLake
in Delta format

Credential vaulting & managed VNET
connectivity options



Control Flow vs. Data Flow



Control: orchestration, branches, loops, variables

Data: transformations via Dataflows Gen2 (Power Query) & Spark notebooks

Pipelines coordinate both for ELT in the Lakehouse



Pipeline Activities (Most Used)

Copy data (engine from ADF)

Notebook (Spark)

SQL (Lakehouse/Warehouse)

Dataflow Gen2

REST/Web & event/schedule triggers



Copy Activity



Optimized source→sink paths;
partition & parallelize

Native Delta sink and OneLake
shortcuts

Incremental loads with watermarks



Orchestration Patterns



Bronze → Silver → Gold

ELT with notebooks + SQL

Event-driven ingestion; re-run failed slices



Security & Governance

Workspace roles + item-level permissions

Credentials/gateways; audit & monitoring

Lineage to downstream Power BI



Demo



Summary

Fabric Pipelines bring ADF orchestration into the Lakehouse

Lower infra overhead; faster iteration

Best fit for modern ELT patterns



Block 2: Spark

Fundamentals

Block 2: Spark

Notebooks

What are Notebook in MS Fabric

- Interactive, code-first experience for data engineering and data science
- Supports Python, SQL, Scala, MD in a single workflow
- Ideal for exploration, ETL/ELT, analytics and automation
- Tight integration with OneLake, Lakehouse & warehouse



Multi-language Authoring

Execute Python, PySpark, SQL, Scala in separate cells

Use %sql, %scala, %md for switching languages

Mix code + narrative for clear storytelling

Ideal for combining transformations + documentation



Notebook Structure & Productivity

Rich cell types: code, markdown, visual output

IntelliSense, autocomplete & syntax highlighting

Inline visualizations from Pandas, Spark, Plotly, etc.

Variable state preserved across cells within a session



Working with Lakehouse & Onelake

Direct read/write to Lakehouse tables and files

Support for Delta Lake operations

Use %sql or PySpark for managed tables

Notebooks automatically reference workspace items

Default Lakehouse and attaching other lakehouses have risks



Data Acquisition and Transformation

Load data using Spark, Pandas, SQL

Transform using DataFrames, Delta Lake commands

Perform joins, aggregations, window functions

Write clean ELT flows end-to-end inside the notebook



Delta Table integration

Create and manage Delta Tables

MERGE, UPDATE, OPTIMIZE, VACUUM operations

Time travel for reproducible analysis

Transactional operations across OneLake



Viz & Explore

Built-in display functions for DataFrames

Create charts with matplotlib, seaborn, plotly, Spark display

Build exploratory data analysis (EDA) workflows

Great for teaching, demos, and stakeholder communication



Fabric Items integration

Trigger notebooks inside Pipelines

Schedule execution using Jobs

Consume from and write to:

- Lakehouse
- Warehouse
- KQL DB
- ML models & environments (without going into env details)

Notebooks behave as first-class Fabric items



Versioning & Collaboration

Automatic version history for all notebooks

Compare and restore previous versions

Git integration (Repos) for collaborative development

Safe CI/CD workflows with Fabric Deployment Pipelines



Operationalization & Reusability

Parameterize notebooks using Pipeline parameters

Reuse notebooks across multiple workflows

Log outputs for monitoring & lineage

Export results to Lakehouse, Warehouse, or Power BI



Block 2: Spark

More stuff

Block 2: Spark

Environments

Block 2: Spark

Data Validations and Expectations

Block 2: Spark

Delta Optimizations

Partitioning In Delta Lake

Organizes data into directory-level partitions

Reduces file scans through predicate pruning

Best for low-cardinality, high-selectivity columns
Mis-partitioning can cause small-file explosion

Delta operations remain partition-aware (OPTIMIZE, Z-Order, V-Order)



V-Order

Write-time optimization improving file layout

Applies:

- Advanced sorting
- Row-group distribution
- Dictionary encoding
- Compression strategies

Boosts read performance (10%–50%)

Fully Parquet-compliant & compatible with Z-Order, Delta features



Z-Order

Multi-column clustering technique

Reorders data to minimize file-level scatter

Efficient for high-cardinality filter columns

Reduces data scan by co-locating related rows

Applied via OPTIMIZE ZORDER BY



Delta File compaction & Optimize

Consolidates many small Parquet files

Reduces metadata overhead

Improves scan efficiency & caching

Essential after streaming/microbatch ingestion

Works with Z-Order & V-Order



Vacuum

Cleans unreferenced Parquet files

Maintains table hygiene & reduces storage

Works with Delta log retention

Default retention: 7 days

Removes files after log checkpoints validate they are no longer needed



Block 3: SQL Analytics Endpoint and Data Warehouse

Block 3: Streaming

What is Streaming in MS Fabric

Fabric supports end-to-end real-time data processing through Real-Time Intelligence (RTI).

Handles streaming ingestion, transformation, storage, analytics, visualization, and real-time actions in a unified platform.

Built for data in motion, not batch — enables millisecond- to second-latency decisioning.

Centralized governance and security across all real-time workloads.



Event Streams

Eventstreams capture, transform, and route high-volume data from sources like Azure Event Hubs, Kafka, IoT Hub, AWS Kinesis, GCP Pub/Sub.

Provide no-code authoring, built-in transformations (filter, aggregate, enrich), and schema-aware routing.

Support millions of events per second; ideal for IoT, anomaly detection, logs, telemetry, and high-frequency event data.



Eventhouse

Eventhouse stores high-performance real-time data optimized for sub-second analytics.

Uses KQL (Kusto Query Language) to query billions of rows quickly. Supports windowed aggregations, anomaly detection, pattern matching, and materialized views.

Glue between real-time ingestion and dashboards or automated actions.



Real-Time Dashboards & Actioning

Real-Time Dashboards update automatically as events arrive.

KQL Querysets visualize live signals: time charts, aggregates, anomaly flags.

Data Activator automatically triggers alerts and downstream actions based on event conditions.

Enables fraud detection, supply chain alerts, live personalization, and operational insights.



End – To– End Streaming

Ingest: Eventstream pulls data from IoT, Kafka, Event Hubs, apps.

Process: Transform events (filter, join, enrich) with low-code/no-code operations.

Store: Eventhouse + KQL database store time-series and event-driven data.

Analyze: Run sub-second KQL analytics, aggregations, anomaly detection.

Act & Visualize:

1. Real-time dashboards for monitoring
2. Data Activator triggers alerts or automated responses
Power BI embeds real-time visuals



Block 4: Security & Identity

What are our layers of security?

Item-Level Security

- Controls permissions

Data-Level Security

- Fine-grained control over actual data
- Includes RLS, OLS, Column-Level Security, table permissions, and OneLake file-level access

Tenant-Level Security

- Controls Fabric availability, governance, and enterprise-wide policies

Capacity-Level Security

- Controls workloads, performance, scaling, and workspace assignment

Workspace-Level Security

- Managed by Capacity Admins



Tenant-Level (Organization-wide)

- Enable/disable Fabric
- Workspace creation rules
- DLP policies, sensitivity labels
- Audit logs & monitoring
- Conditional Access, MFA, network restrictions



Capacity-Level (Resource Governance)

- Assign workspaces to capacities
- Manage workload settings (Lakehouse, Warehouse, KQL, AI, etc.)
- Monitor performance & utilization
- No access to workspace content





Admin

- Access to all items in admin mode
- Can share access with other users

Member

- Access to all items in admin mode
- Can share access with other users

Contributor

- Access to all items in admin mode

Viewer

- Read access to all items

Item-Level Security



Read

- Allows you to connect to SQL DB, SQL Endpoint

Write

- Admin-like permission, full read access to item and full modification permissions to item

Reshare

- Allows you to share the item with other users

ReadData

- Gives you db_datareader on the SQL Endpoint

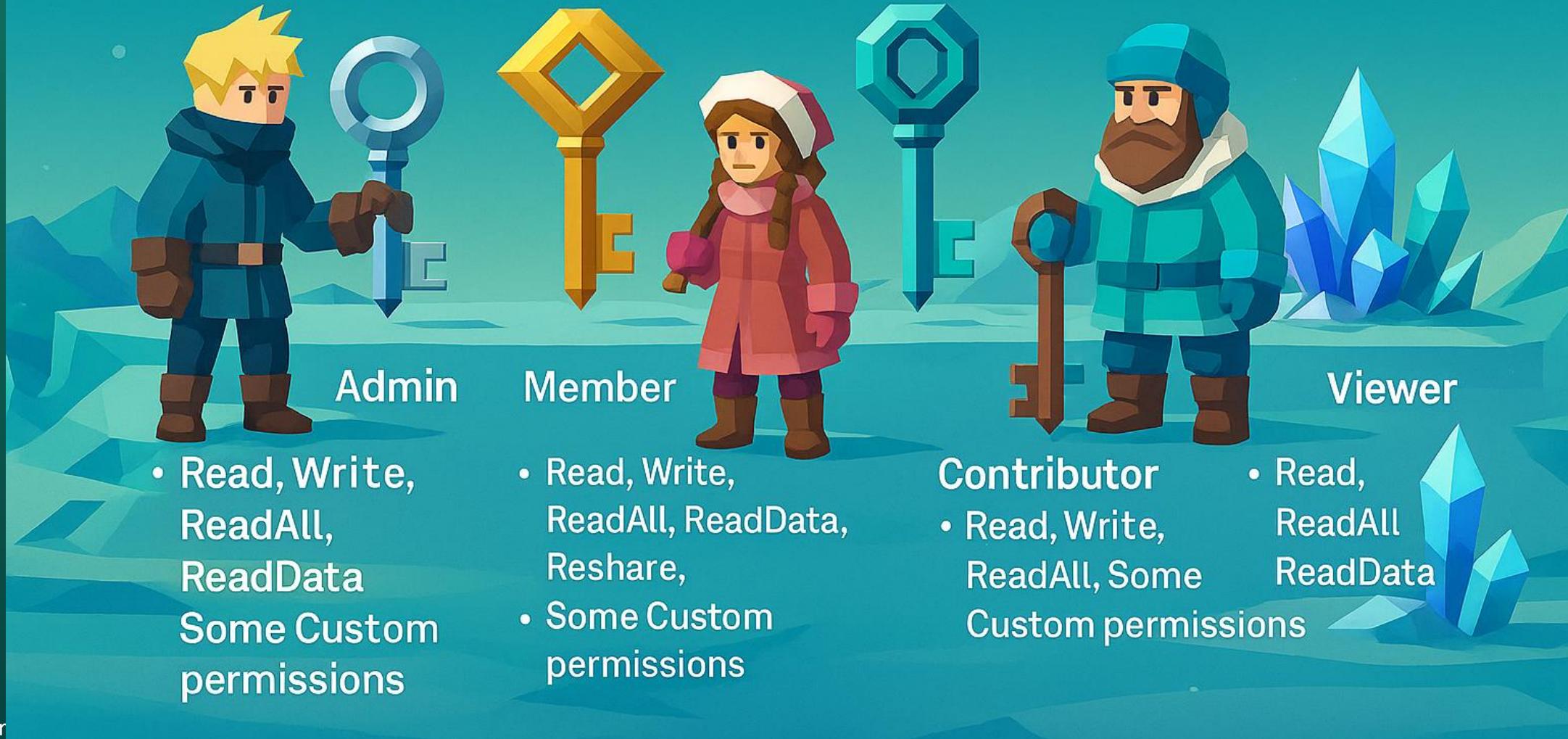
ReadAll

- Gives you access to all files in Onelake (Della tables and Files)

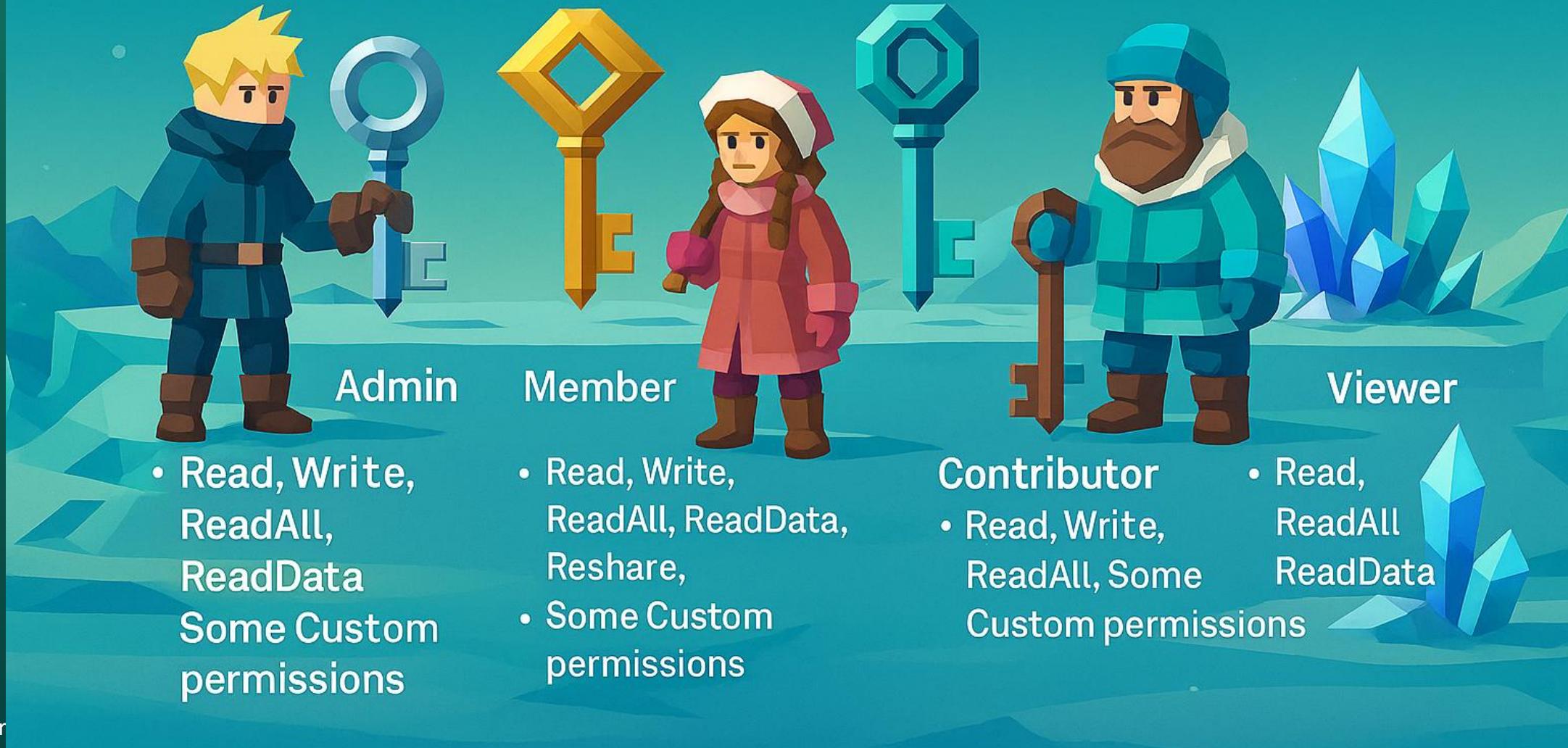
Many custom ones



Item level Security and Workspace level security



Item level Security and Workspace level security



Item level Security and Workspace level security

Users with Reshare permission on an item
Can grant permission they have themselves on an item

Workspace Roles with the Reshare permission
Can grant permissions they have themselves on an item except the WRITE permission



Some thoughts

Workspace Roles
are just a
combination of
Item permissions
propagated to all
items



It is kind
of a mess

Admin, Member,
Contributor are
basically GOD mode
permissions due to
the WRITE permission



So where does Onelake Security fit in?



There are no changes
to roles with WRITE
permissions

- Admin, Member,
Contributor

Onelake security
applies to Viewer
workspace roles
and Users
with item permissions



What does Onelake Security cover

Data residing in Onelake

- Read access to files and tables
- Row-level security to tables
- Write access to tables (Lakehouse only*)



The data access
through the SQL Endpoint
If the right mode is selected*

Direct lake
semantic models

Shortcuts*



Onelake Security & SQL Endpoint

Delegated Identity mode

- Access to Delta tables (on Onelake) happens through the item owner's identity

Onelake Security mode

- User Identity is used to access the tables through the Compute engine and on Onelake

Switching between modes removes all current SQL security permissions



Onelake Security & Shortcuts



Passthrough
shortcuts

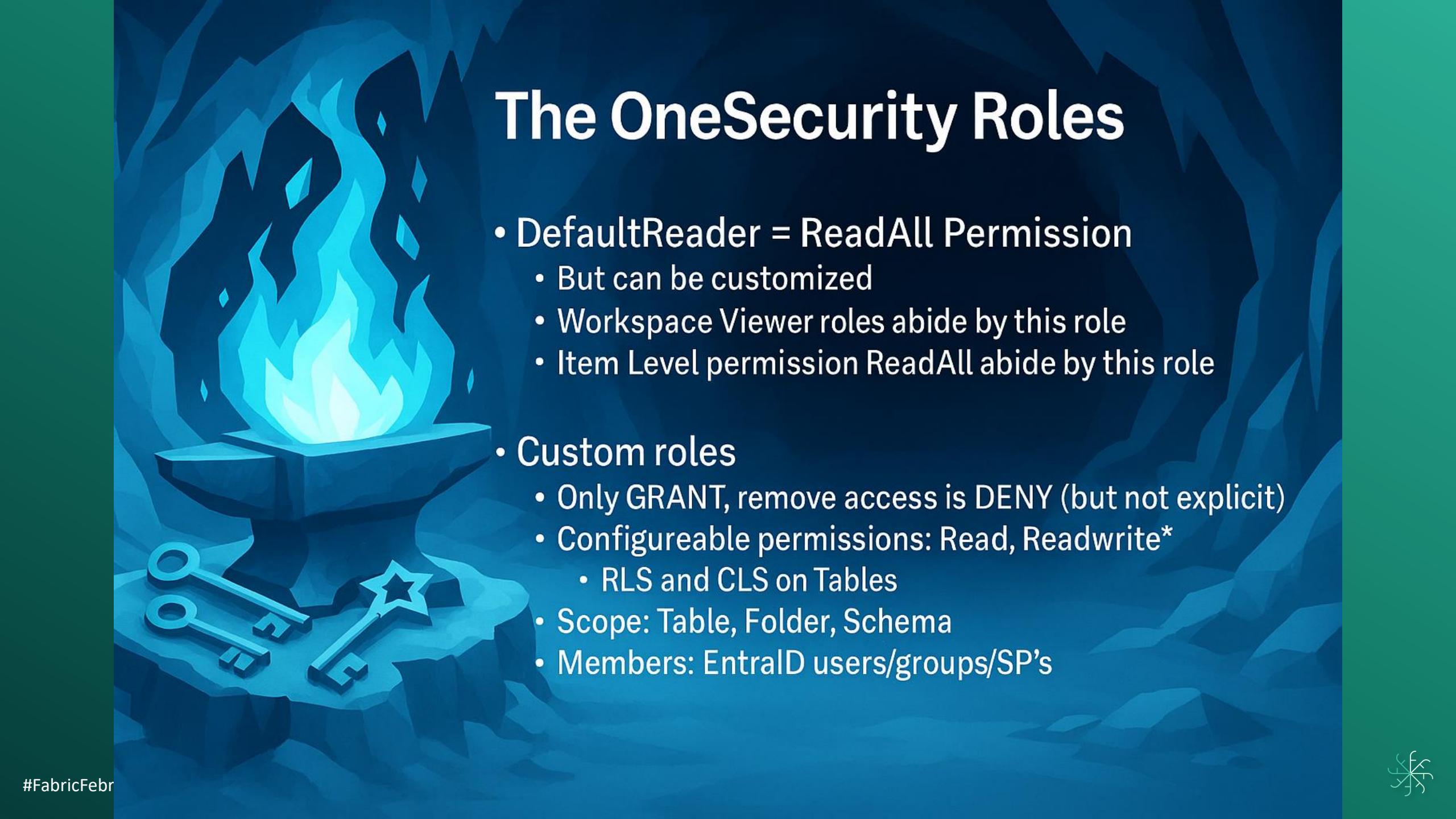
The credential of the querying user is evaluated against the shortcut target to determine what data is allowed to be seen.



Delegated
shortcuts

The shortcut uses a fixed credential to access the target and the querying user is evaluated against OneLake security prior to checking the delegated credential's access to the source.





The OneSecurity Roles

- DefaultReader = ReadAll Permission
 - But can be customized
 - Workspace Viewer roles abide by this role
 - Item Level permission ReadAll abide by this role
- Custom roles
 - Only GRANT, remove access is DENY (but not explicit)
 - Configureable permissions: Read, Readwrite*
 - RLS and CLS on Tables
 - Scope: Table, Folder, Schema
 - Members: EntralD users/groups/SP's



Limitations!

No Warehouse/Eventhouse (but are roadmap)

Changes to role definition take 5 minutes to apply

Changes to a user group in a OneLake security role take about an hour for OneLake to apply the role's permissions on the updated user group.

Some Fabric engines have their own caching layer, so might require an extra hour to update access in all systems.

OneLake security doesn't support cross-region shortcuts. Any attempts to access shortcut to data across different capacity regions result in 404 errors.

If you add a distribution list to a role in OneLake security, the SQL endpoint can't resolve the members of the list to enforce access. The result is that users appear not to be members of the role when they access the SQL endpoint. DirectLake on SQL semantic models are subject to this limitation too.

To query data from a Spark notebook using Spark SQL, the user must have at least Viewer access in the workspace they're querying.

Mixed-mode queries are not supported. Single queries that access both OneLake security enabled and non-OneLake security enabled data will fail with query errors.

Spark notebooks require that the environment be 3.5 or higher and using Fabric runtime 1.3.

OneLake security doesn't work with [private link protection](#).

The [external data sharing preview](#) feature isn't compatible with the data access roles preview. When you enable the data access roles preview on a lakehouse, any existing external data shares might stop working.

Azure Mirrored Databricks Catalog does not support Manage Catalog functionality if OneLake security is enabled on that item. This functionality is coming in November, 2025.

The following table provides the limitations of OneLake data access roles.



Block 4: Monitoring



fabric
FEBRUARY

Thank you to our Fabric February Friends!

bouvet



twoday

KOKAI

sopra  steria

 Evidi

 Crayon
CONSULTING

 Profisee
Master Data Management



Tabular Editor

 ANALYTICS
masterminds

VASS

 Fraktal

 Microsoft

KURANT



EUROPEAN
MICROSOFT
FABRIC
Community Conference
BARCELONA 21-22 Sept + 01 Oct 2020

 profitbase
PART OF HYPERGENE

 Capgemini

 fabric
FEBRUARY



Share your thoughts and help our speakers!



fabfeb.app/feedback

