# Thank you sponsors!









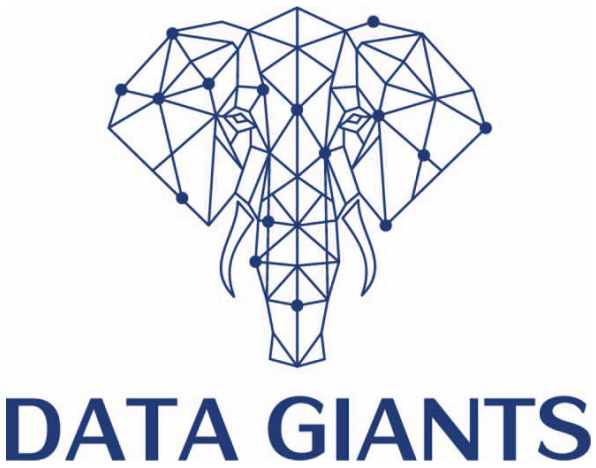






























From ingestion to insight: Understanding where performance gets lost – and what to do about it.

# Fabric Performance Pitfalls – and how to avoid them



# What is a Performance Pitfall?

From capacity planning

to spark clusters

to parallelism or sequential loading

to dimensional modeling

and Roches maxim

*"Little by little, a little becomes a lot"*

# LINDA TORRÅNG (SHE/HER)

mastermınds group

**Managing Director of Analytics**

Retail/Journalism

Mother

Adrenaline Junkie

Rubberduck

Microsoft® MVP Most Valuable Professional

Swedish Power BI & Fabric User Group





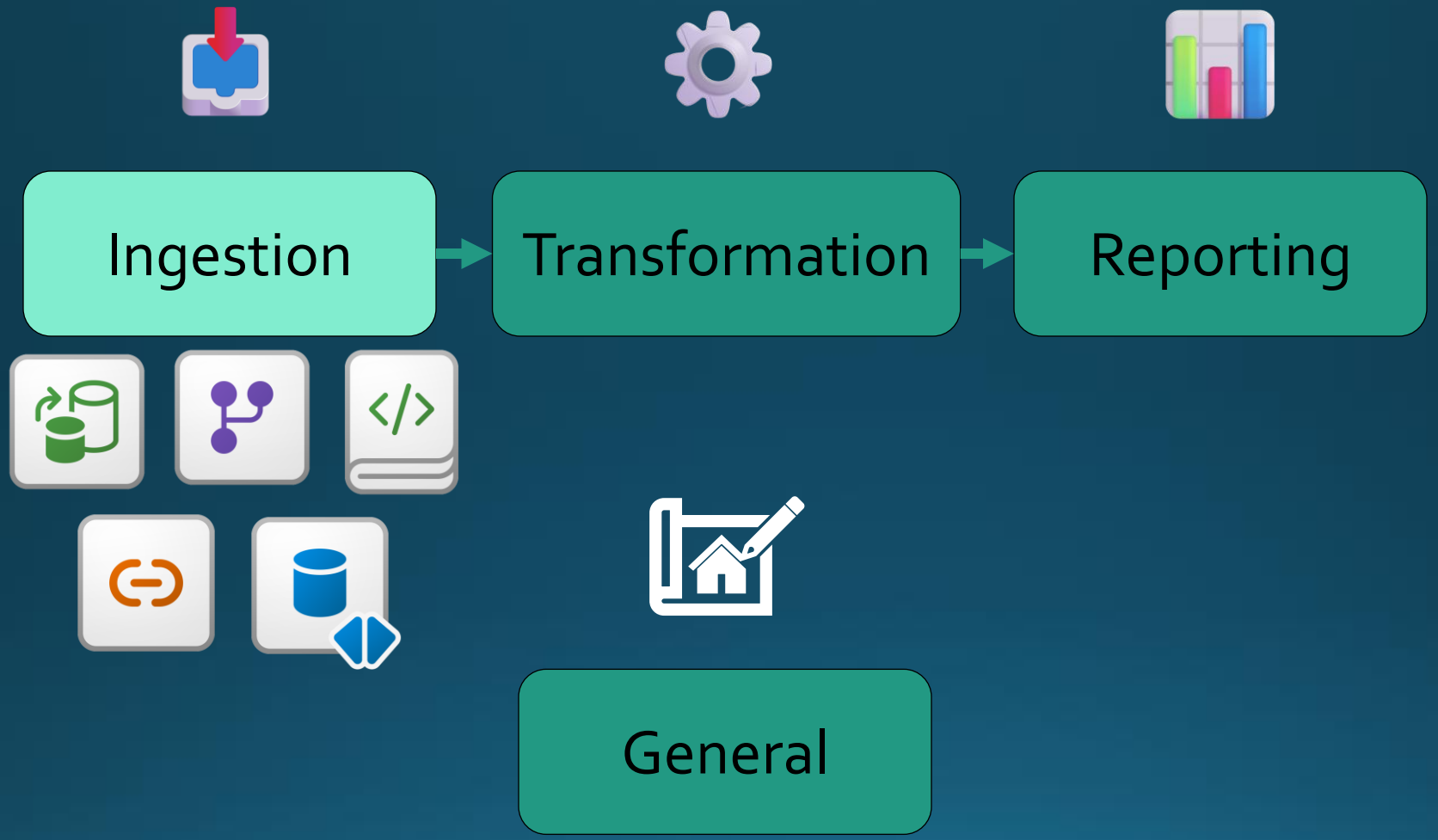




Ingestion → Transformation → Reporting



General


Ingestion
Transformation
Reporting
General

# Ingestion Pitfalls

High data volume without filtering

# Ingestion Pitfalls

High data volume without filtering

Missing or incorrect incremental load

# Ingestion Pitfalls

High data volume without filtering

Missing or incorrect incremental load

Too many small files

# Ingestion Pitfalls

- High data volume without filtering
- Missing or incorrect incremental load
- Too many small files
- Multiple ingestion points for the same data

# Ingestion Pitfalls

- High data volume without filtering
- Missing or incorrect incremental load
- Too many small files
- Multiple ingestion points for the same data
- Unused staging layers

# Ingestion Pitfalls

| | |
|---|---|
| High data volume without filtering | Leverage shortcuts |
| Missing or incorrect incremental load | |
| Too many small files | |
| Multiple ingestion points for the same data | |
| Unused staging layers | |

# Ingestion Pitfalls

- High data volume without filtering
- Missing or incorrect incremental load
- Too many small files
- Multiple ingestion points for the same data
- Unused staging layers

- Leverage shortcuts
- Case sensitive

# Ingestion Pitfalls

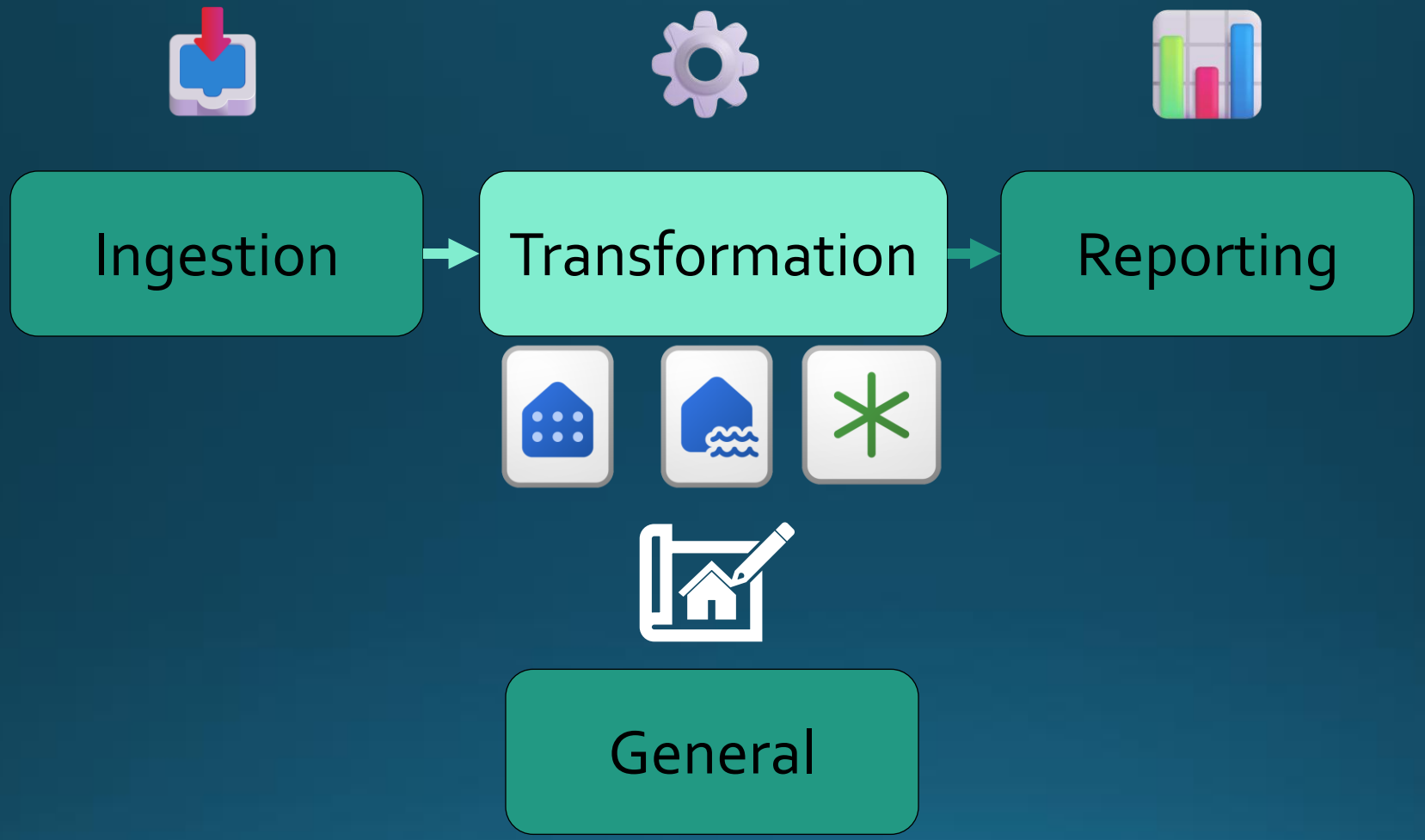
- High data volume without filtering
- Missing or incorrect incremental load
- Too many small files
- Multiple ingestion points for the same data
- Unused staging layers
- Leverage shortcuts
- Case sensitive
- Retry storm

# Ingestion Pitfalls

- High data volume without filtering
- Missing or incorrect incremental load
- Too many small files
- Multiple ingestion points for the same data
- Unused staging layers
- Leverage shortcuts
- Case sensitive
- Retry storm
- Mirrored ownership

# Ingestion Pitfalls

- High data volume without filtering
- Missing or incorrect incremental load
- Too many small files
- Multiple ingestion points for the same data
- Unused staging layers
- Leverage shortcuts
- Case sensitive
- Retry storm
- Mirrored ownership
- Schema Evolution


Ingestion
Transformation
Reporting
General

# Transformation Pitfalls



Partitions

# Transformation Pitfalls



Partitions



No use of OPTIMIZE/VACCUM

# Transformation Pitfalls



Partitions



No use of OPTIMIZE/VACCUM



Overuse of SCD2

# Transformation Pitfalls

- Partitions
- No use of OPTIMIZE/VACCUM
- Overuse of SCD2
- Liquid clustering

# Transformation Pitfalls

- Partitions
- No use of OPTIMIZE/VACCUM
- Overuse of SCD2
- Liquid clustering
- No modular or reusable code

# Transformation Pitfalls

- Partitions
- Spark cluster
- No use of OPTIMIZE/VACCUM
- Overuse of SCD2
- Liquid clustering
- No modular or reusable code

# Transformation Pitfalls

- Partitions
- No use of OPTIMIZE/VACCUM
- Overuse of SCD2
- Liquid clustering
- No modular or reusable code
- Spark cluster
- Lakehouse vs Data Warehouse

# Transformation Pitfalls

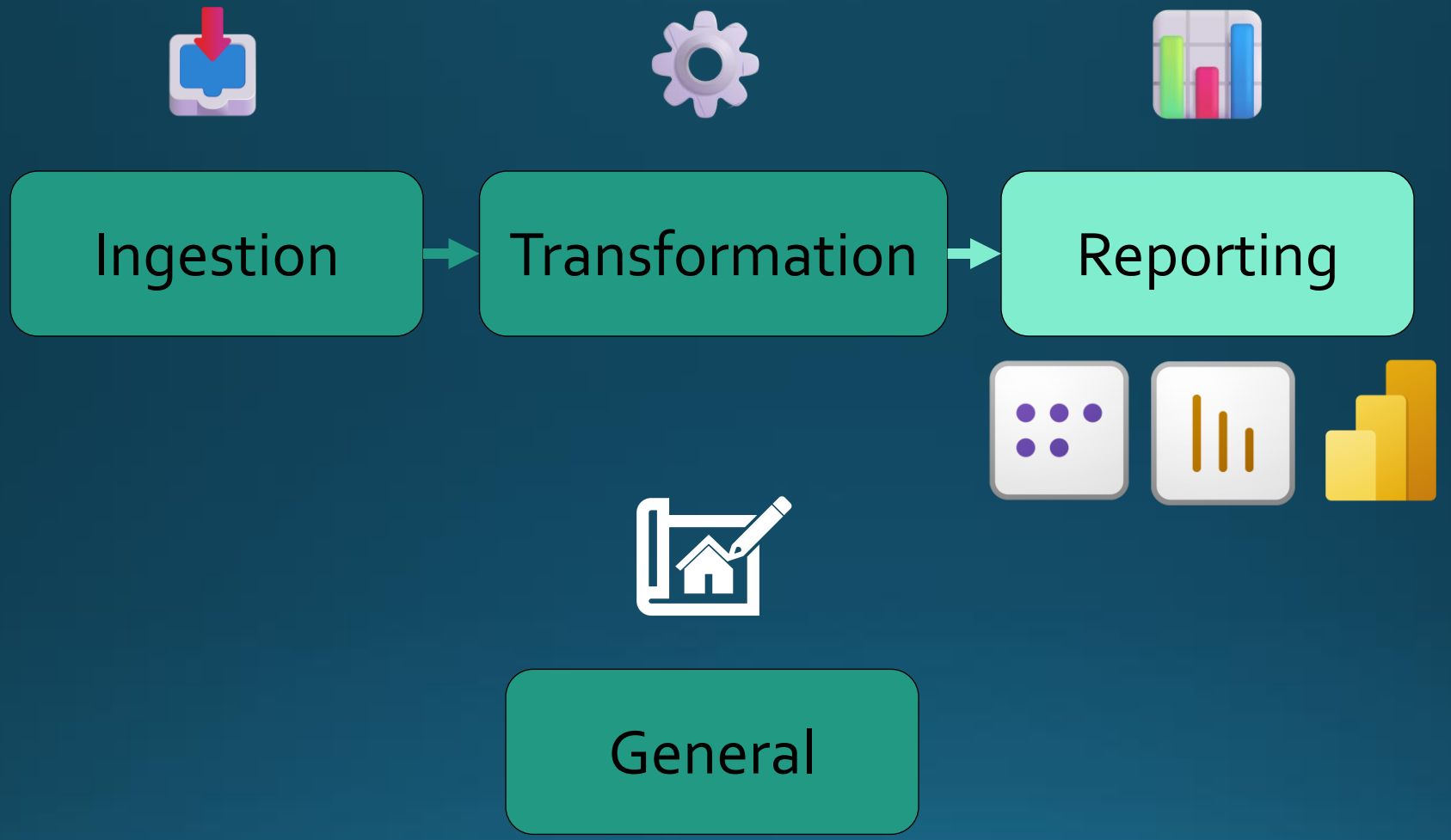
- Partitions
- Spark cluster
- No use of OPTIMIZE/VACCUM
- Lakehouse vs Data Warehouse
- Overuse of SCD2
- Native execution engine
- Liquid clustering
- No modular or reusable code

# Transformation Pitfalls

- Partitions
- Spark cluster
- No use of OPTIMIZE/VACCUM
- Lakehouse vs Data Warehouse
- Overuse of SCD2
- Native execution engine
- Liquid clustering
- Parallel / Sequential
- No modular or reusable code

# Transformation Pitfalls

- Partitions
- Spark cluster
- No use of OPTIMIZE/VACCUM
- Lakehouse vs Data Warehouse
- Overuse of SCD2
- Native execution engine
- Liquid clustering
- Parallel / Sequential
- No modular or reusable code
- Code execution


Ingestion
Transformation
Reporting
General

# Reporting Pitfalls

- Poor data model

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables
- AI-assisted optimization

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables
- AI-assisted optimization
- DAX

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables
- AI-assisted optimization
- DAX
- No use of hybrid tables

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables
- AI-assisted optimization
- DAX
- No use of hybrid tables
- TMDL

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables
- AI-assisted optimization
- DAX
- No use of hybrid tables
- TMDL
- Performance analyzer

# Reporting Pitfalls

- Poor data model
- Missing aggregation tables
- AI-assisted optimization
- DAX
- No use of hybrid tables
- TMDL
- Performance analyzer
- RLS

# Reporting Pitfalls

- Poor data model
- TMDL
- Missing aggregation tables
- Performance analyzer
- AI-assisted optimization
- RLS
- DAX
- Direct Lake options
- No use of hybrid tables

# Reporting Pitfalls

- Poor data model
- TMDL
- Missing aggregation tables
- Performance analyzer
- AI-assisted optimization
- RLS
- DAX
- Direct Lake option
- No use of hybrid tables
- Roches maxim





Ingestion → Transformation → Reporting



General





# General Pitfalls

- Capacity adjustments

# General Pitfalls



Capacity adjustments



Protect your capacity

# General Pitfalls

- Capacity adjustments
- Protect your capacity
- Structure to find what you need

# General Pitfalls



Capacity adjustments



Protect your capacity



Structure to find what you need



Version / Source control

# General Pitfalls



Capacity adjustments



Protect your capacity



Structure to find what you need



Version / Source control



Logging

# General Pitfalls

- Capacity adjustments
- Protect your capacity
- Structure to find what you need
- Version / Source control
- Logging
- Lack of pipeline overview and documentation

# General Pitfalls

- Capacity adjustments
- Protect your capacity
- Structure to find what you need
- Version / Source control
- Logging
- Lack of pipeline overview and documentation
- No alerting

# General Pitfalls

- Capacity adjustments
- Protect your capacity
- Structure to find what you need
- Version / Source control
- Logging
- Lack of pipeline overview and documentation
- No alerting
- No clear ownership

# General Pitfalls


Capacity adjustments


Protect your capacity


Structure to find what you need


Version / Source control


Logging


Lack of pipeline overview and documentation


No alerting


No clear ownership


Inefficient language choice

# General Pitfalls

- Capacity adjustments
- Protect your capacity
- Structure to find what you need
- Version / Source control
- Logging
- Lack of pipeline overview and documentation
- No alerting
- No clear ownership
- Inefficient language choice
- AI Agents

# Summary Table of Pitfalls



General:

Monitoring, Governance, Capacity planning



Ingestion:

Unified ingestion, Source evolution



Transformation:

Maintenance, Spark, Reusability



Reporting:

Bad model, Agents, Roches maxim

# Closing Thoughts



Where are your current bottlenecks?



What can you improve right now?



Performance starts with design – not post-fix tuning.

# Rate this session

Microsoft Fabric Performance Pitfalls and How to Avoid Them

