

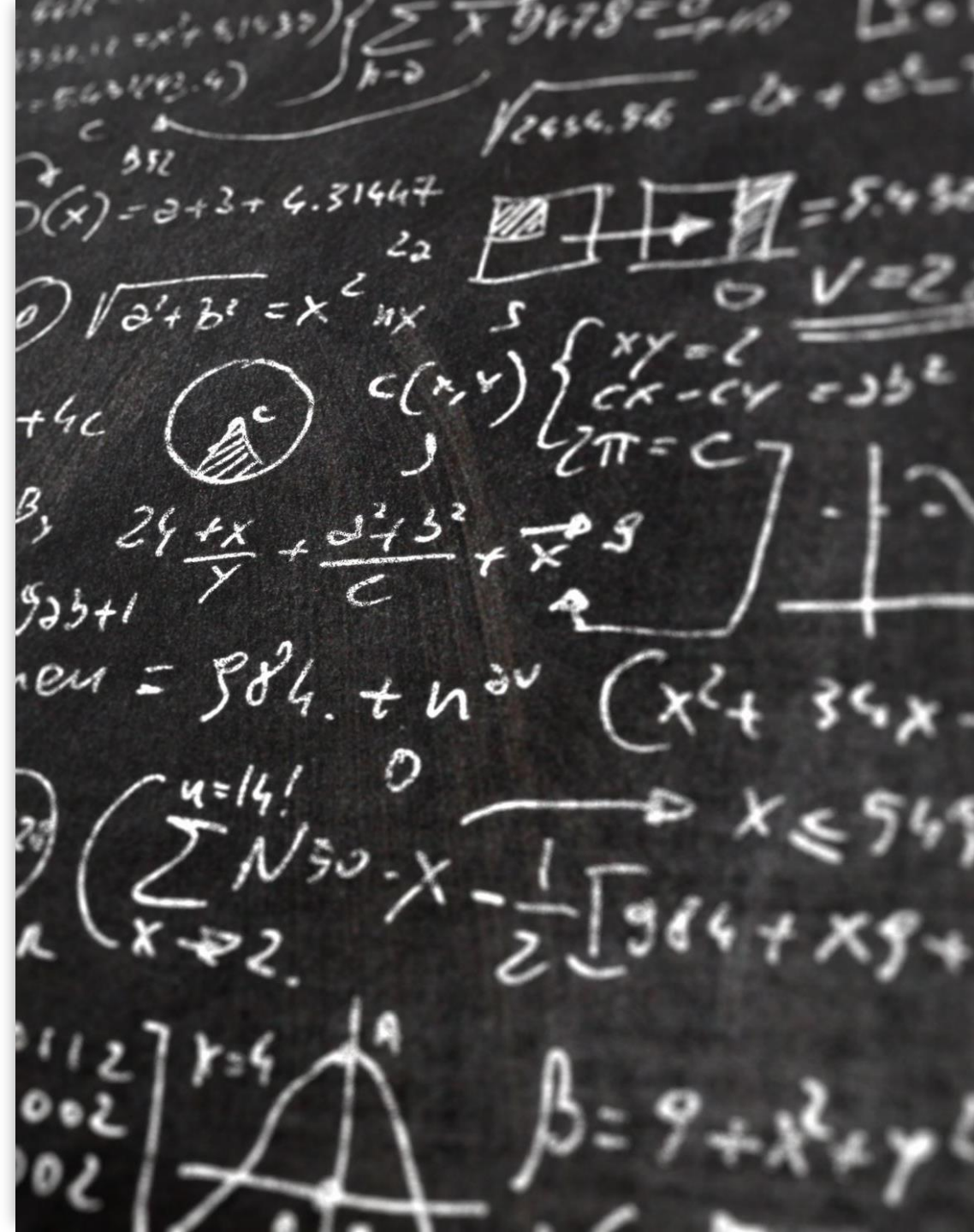


Query Optimization (QO) in Practice: What Are the Realities and Trends?

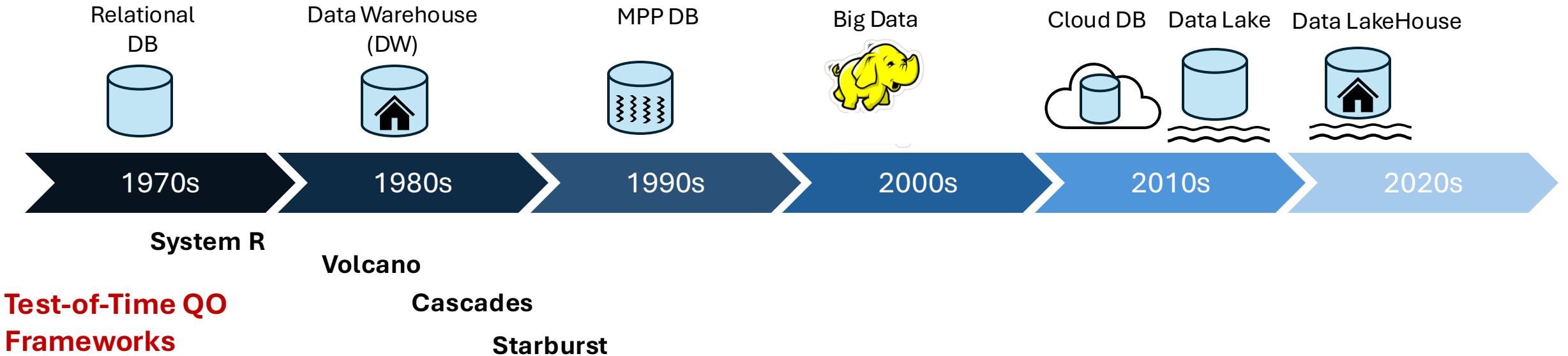
Yuanyuan Tian (Gray Systems Lab, Microsoft)

My personal view, based on learnings from...

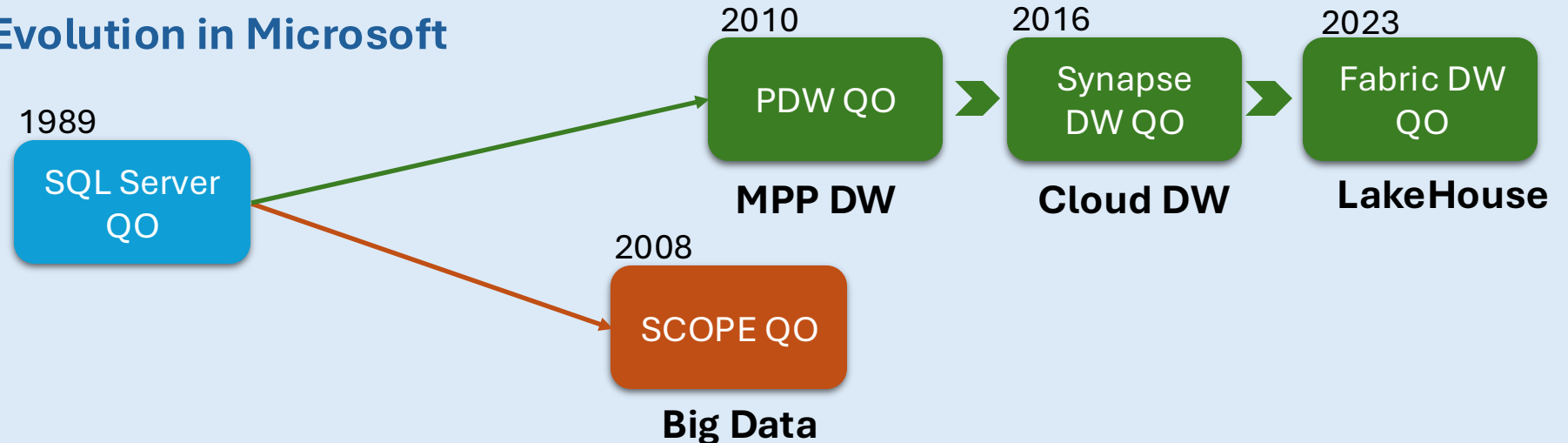
- QO gurus from IBM and Microsoft
 - DB2, SQL Server, Scope, Spark, Calcite
- Stumbles and successes of QO efforts of myself and my colleagues



Relational DB & QO History



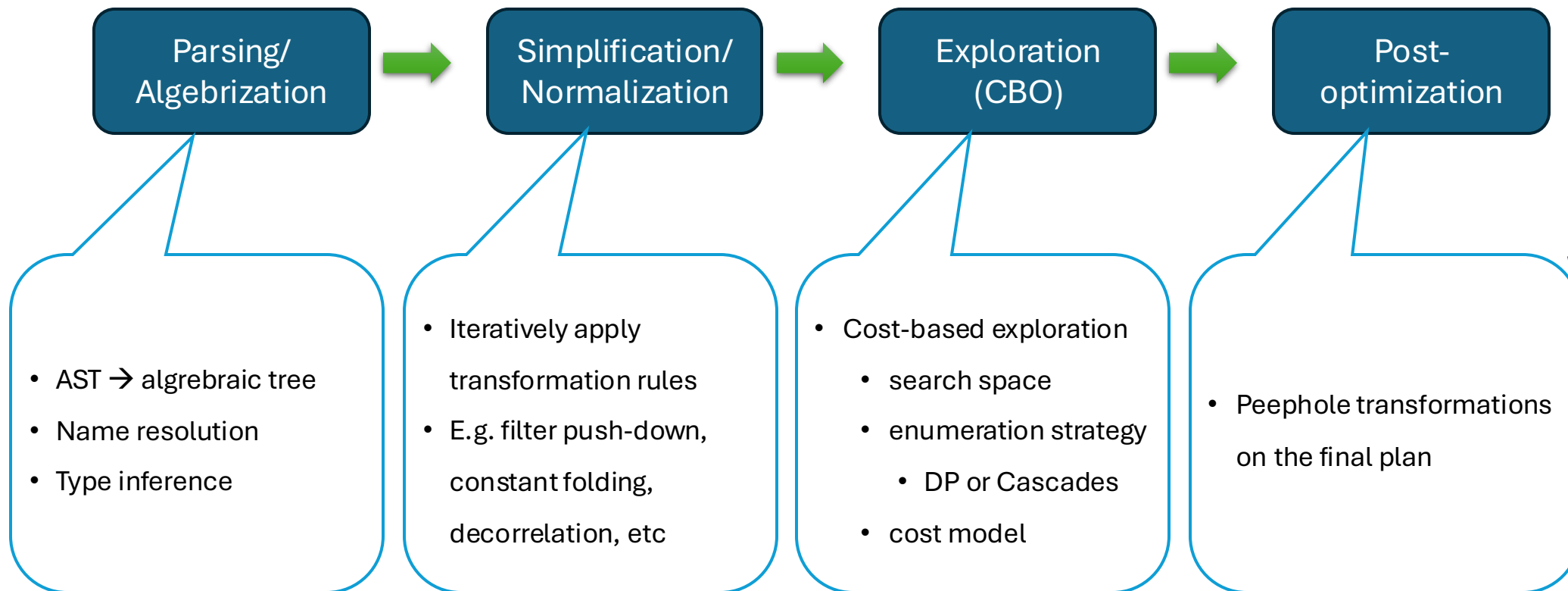
QO Evolution in Microsoft



Common QO Architecture

A proliferation of DB engines with QOs following similar patterns

- Same relational algebra, similar search spaces, and same stages



Is Query Optimization a “Solved” Problem?



Blog

Is Q

Apr

IMH
one
and
pla
disa

B. Mitschang et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2017),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 25

Query Optimization – Are We There Yet?

Guy Lohman¹

Abstract

After nearly 4 decades and hundreds of scientific papers, relational query optimization can hardly be characterized as anything but a huge scientific and commercial success. The

Is Query Optimization a “Solved” Problem?

Answer: **No!**

	Assumptions	Realities
Workload	One query at a time	Queries run simultaneously
Guy Lohman: “So c’mon, folks, let’s attack problems that really matter, those that account for optimizer disasters, and stop polishing the round ball.”		
Weighting	Certain types of cost (I/Os, memory, CPU) dominate others	Hardware changes dramatically over the years
Model Detail	Detailed models are more accurate	More details add MORE assumptions!

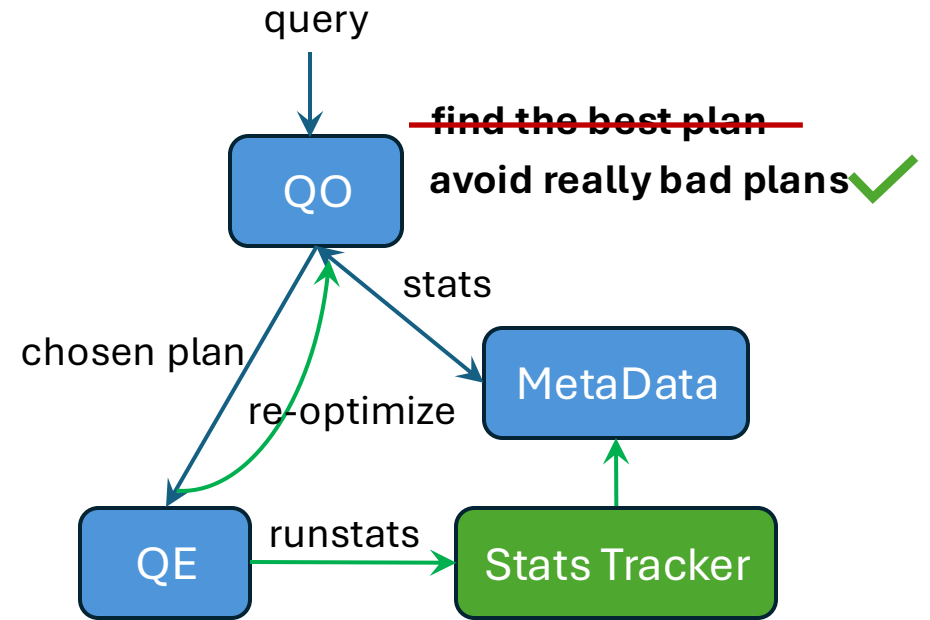


Trends in Practice

- T1: Collaboration between QO and QE (Query Execution)
 - T2: Converging QO with WO (Workload Optimization)
 - T3: From customized to composable QO
-

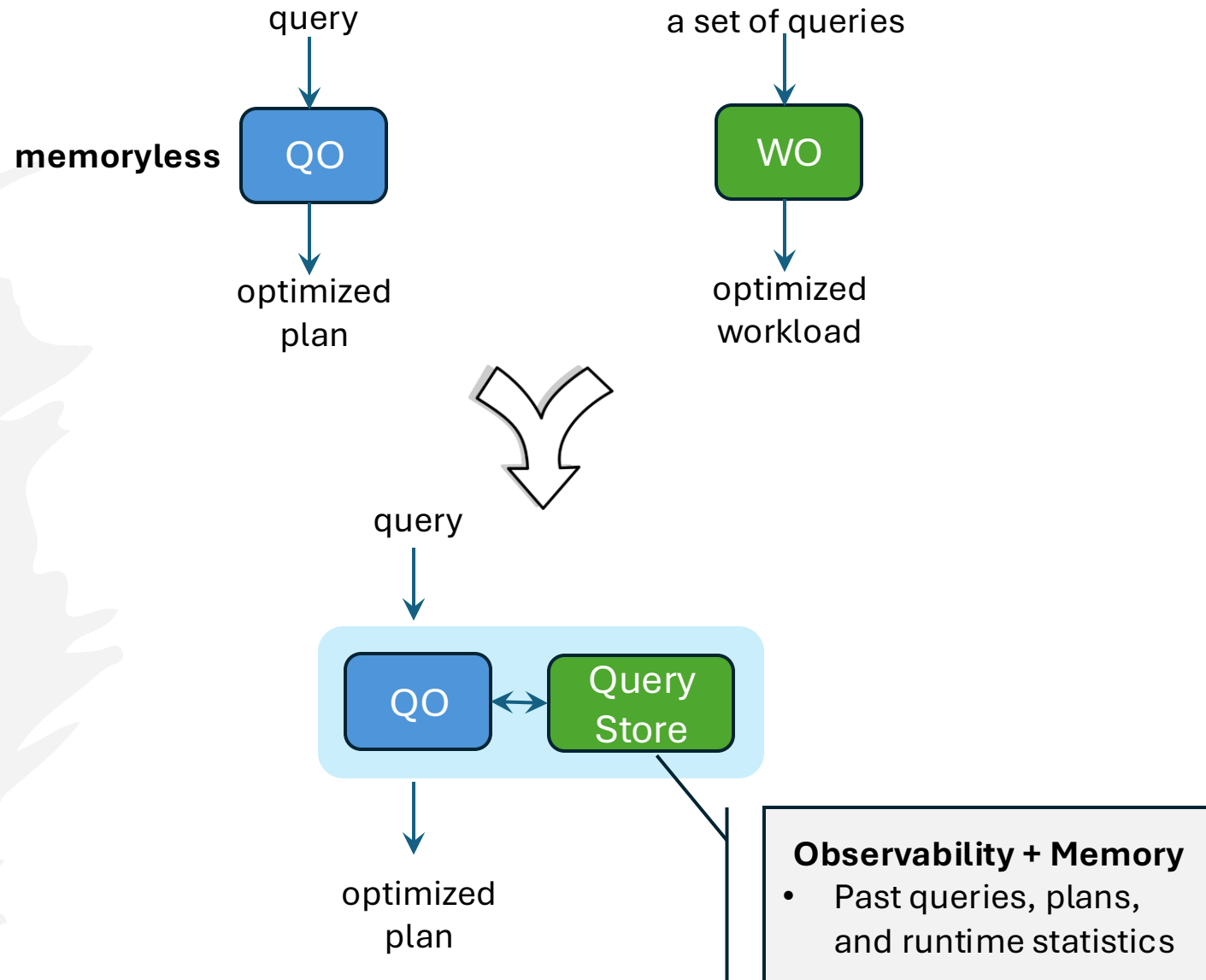
T1: Collaboration between QO and QE

- Compensating QO with robust QE
 - Bloom filters
- Adaptive Query Optimization
 - SQL Server, Oracle:
 - hash join → nested loops join
 - Big Query:
 - shuffle join → broadcast join
 - Databricks:
 - shuffle ↔ broadcast
 - adjust degree of parallelism
 - Restart scans with added bloom filters



T2: Convergence of QO with WO

- Adding workload insight into QO
 - SQL Server
 - Query Store: monitors the performance of query to detect plan changes and regressions
 - Oracle:
 - SQL Plan Management: keep track of a set of valid plans for a query



About Learned QO

- All learned QO approaches are essentially WO
 - Rely on the fact: Queries are often recurrent and similar
 - Learning from the past to improve the future

“Pacemaker” is easier to adopt than a “heart transplant”!



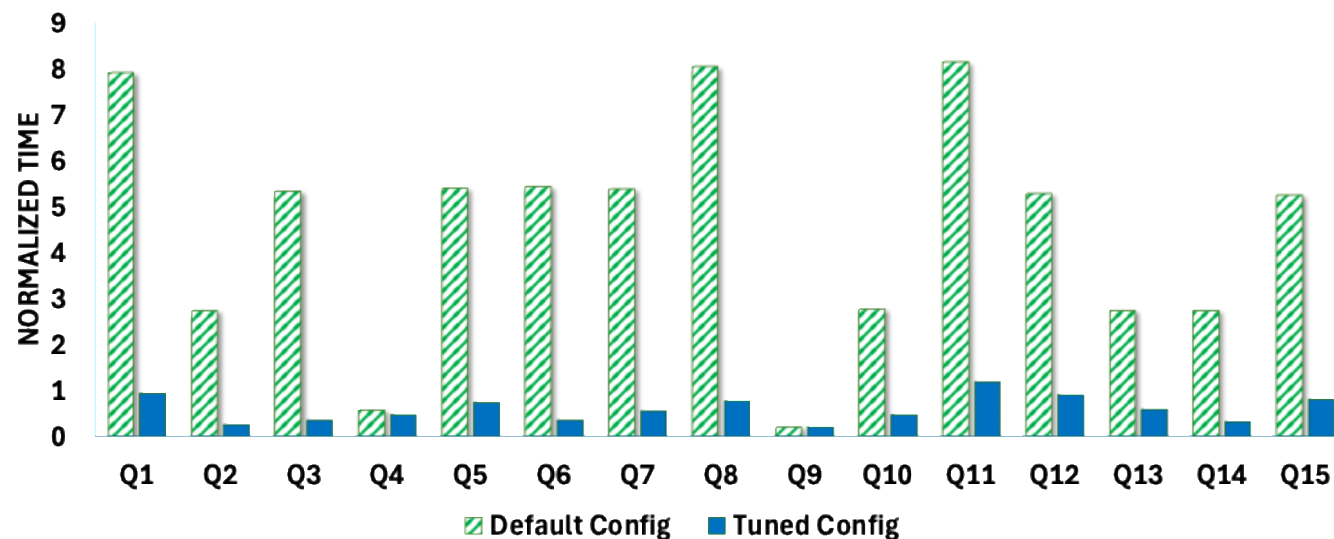
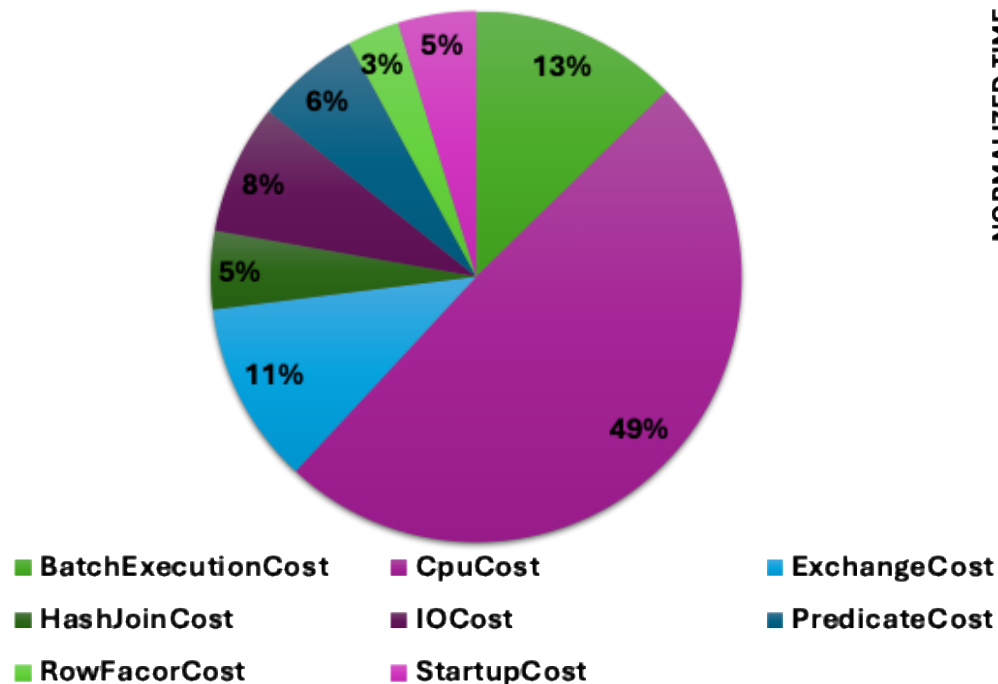
⚠ Complexity
⚠ Explainability
⚠ Debuggability
⚠ Regression
⚠ Training Cost



Example: Recalibrating and Tuning the Cost Model

- Recalibrating and tuning *constant parameters* in SQL Server cost formula

60 parameters



MSSales Workload

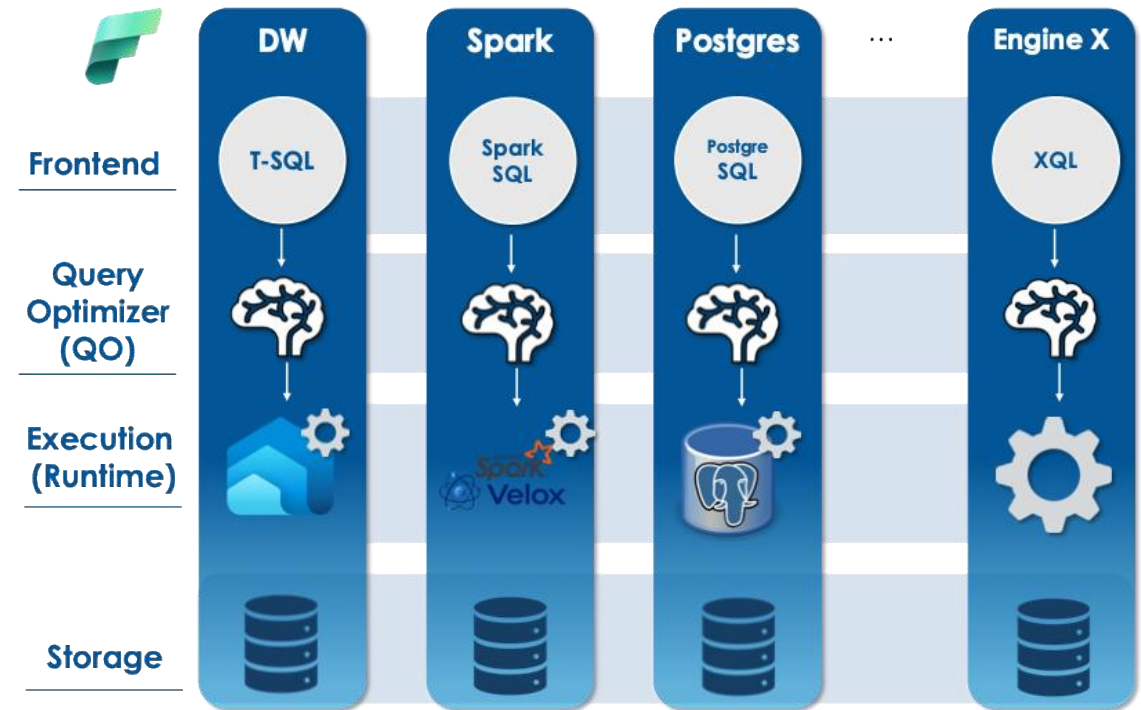
- 627 tables on OneLake
- Highly templated queries, join heavy

T3: From customized to composable QO

 Demand: Fragmentation → Convergence

Microsoft Fabric

- Shared Data on Lake
- Shared Compute Resource
- Shared Governance Experience

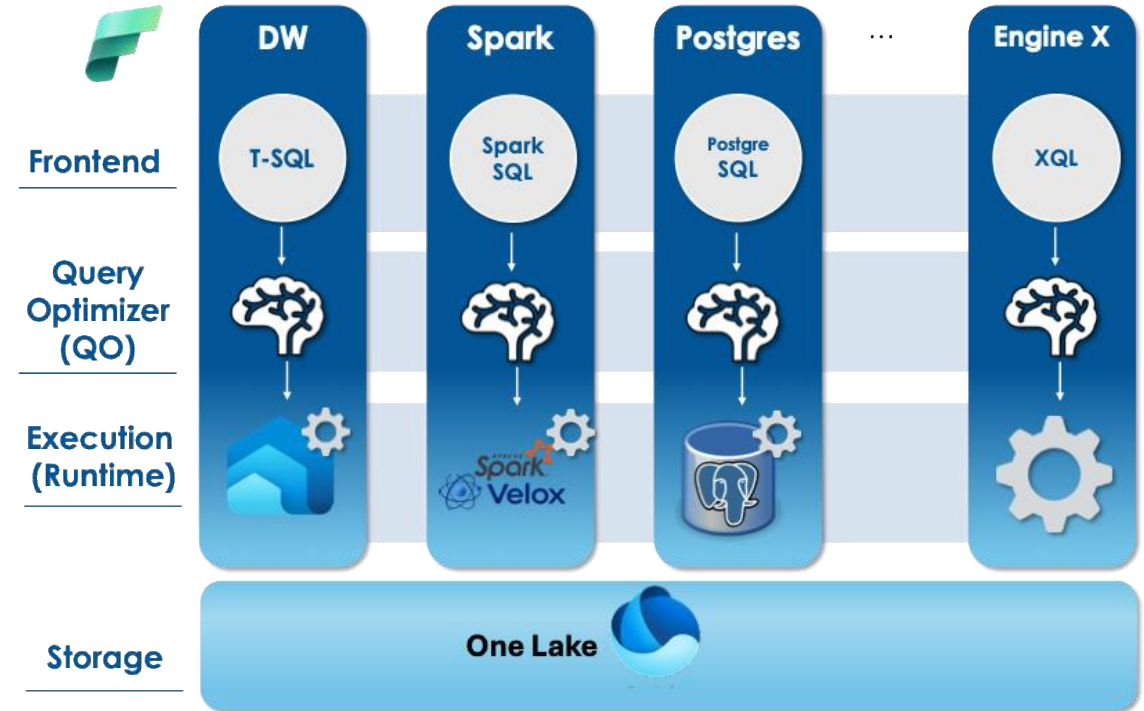


T3: From customized to composable QO

 Demand: Fragmentation → Convergence

Microsoft Fabric

- Shared Data on Lake
- Shared Compute Resource
- Shared Governance Experience



T3: From customized to composable QO

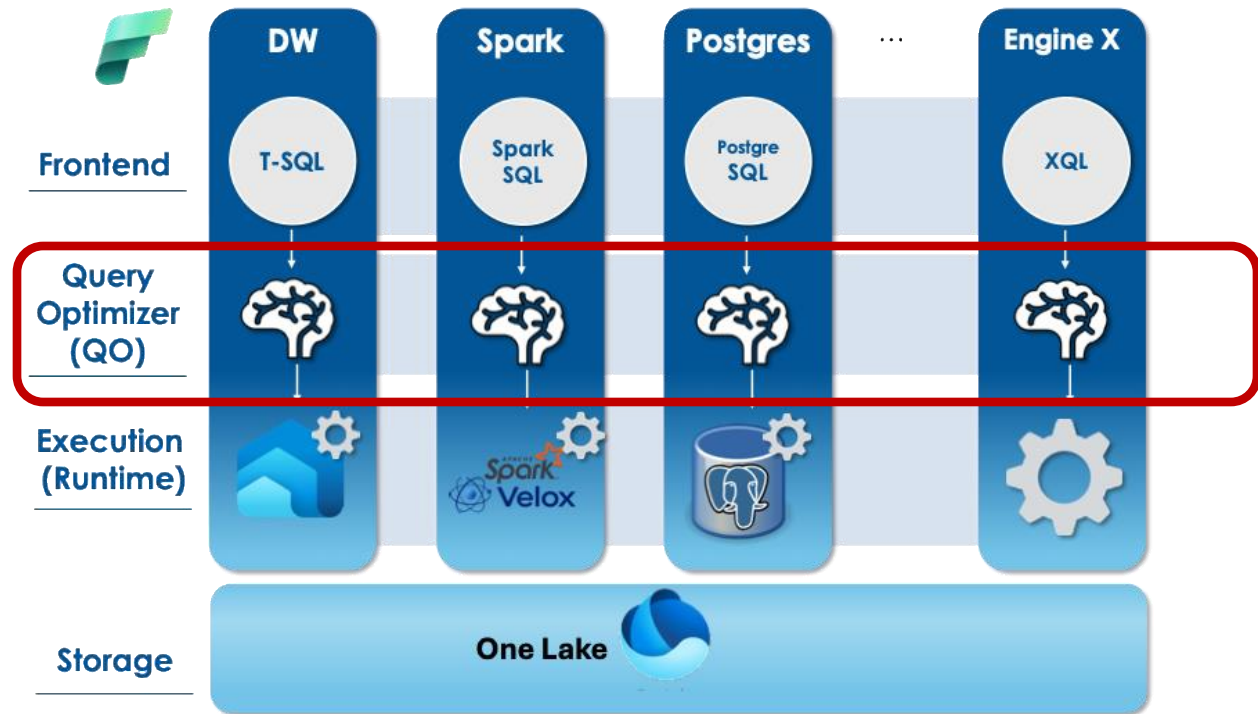
Demand: Fragmentation → Convergence

Microsoft Fabric

- Shared Data on Lake
- Shared Compute Resource
- Shared Governance Experience

Architecture: Monolithic → Composable

- Cloud DB: separation of storage from compute
- Open standards
 - Parquet, Arrow, Substrait
- OSS system-building libraries
 - QE: Velox, Datafusion
 - **QO: Calcite, Orca**



Customized QO → QO as a Library → QO as a Service (QOaaS)

QOaaS: Independent QO service interacting with multiple engines over RPC

Features	Custom QO	QO as a Library (Calcite, Orca)	QOaaS
Innovation speed	✗	✓	✓
Engineering efficiency	✗	✓	✓
New engine time-to-market	✗	✓	✓
QO scalability	✗	✗	✓
Workload Observability	✗	✗	✓
Workload Optimization	✗	✗	✓
Cross-engine optimization	✗	✗	✓

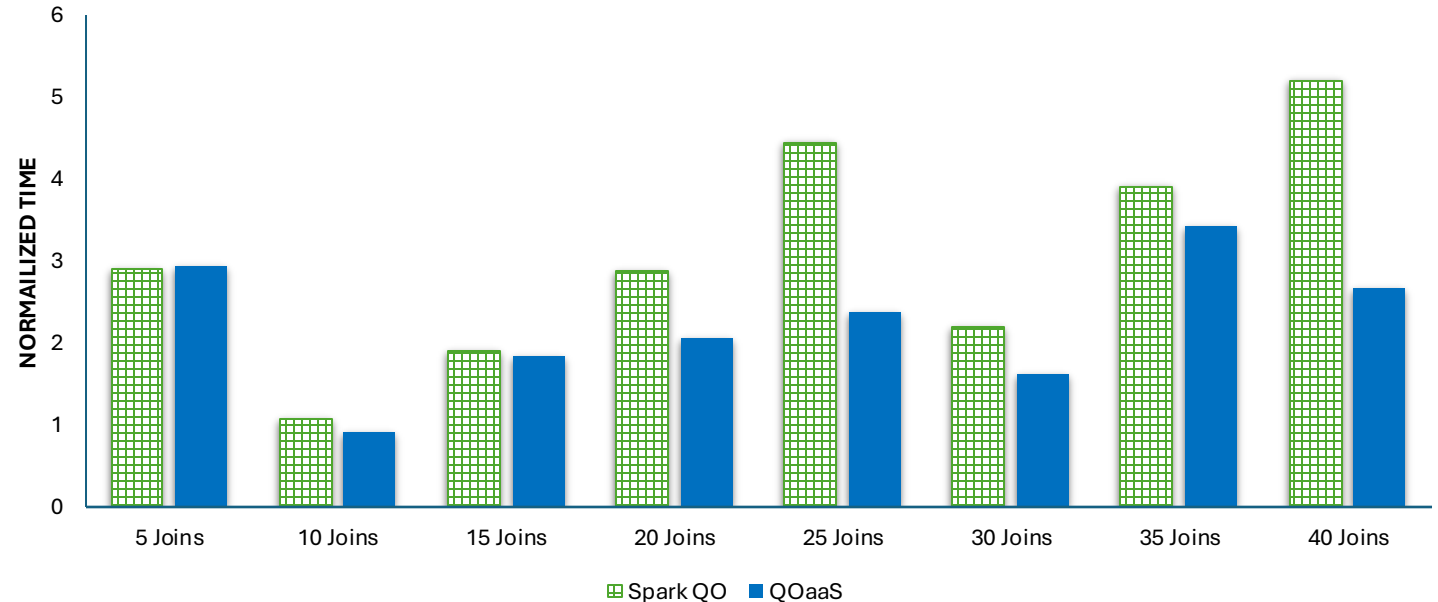


A Simple QOaaS Prototype

- A simple QOaaS prototype for DW and Spark on Fabric Ecosystem
 - Adapting the QO of Fabric DW to QOaaS
 - Adopting Substrait as the cross-engine plan specification

MSSales Workload

- 627 tables on OneLake
- Highly templated queries, join heavy



Conclusion

- Today's DBs are still using the QO frameworks from 4 decades ago
- But QO is not a solved problem: we need to work on the important problems, not to polish the round ball
- Three trends in practice
 - Collaboration between QO and QE
 - Converging QO with WO
 - From customized to composable QO