



SQL Server 2025 Launch

**SQL Server 2025 -
- co nowego w tematach
wydajności i skalowalności**

Jakub Wawrzyniak

Paweł Potasiński

Jakub Wawrzyniak

Senior Solutions Engineer @ Microsoft

- Skilled data solution architect
- Active member of Data Community Poland
- Former MVP

in [in/jakub-piotr-wawrzyniak/](https://www.linkedin.com/in/jakub-piotr-wawrzyniak/)

FAVORITE STUFF:



Paweł Potasiński

CTO @ InfiniteDATA Services

- Founder of Polish SQL Server User Group
- Former Microsofter and MVP
- SQL developer at heart

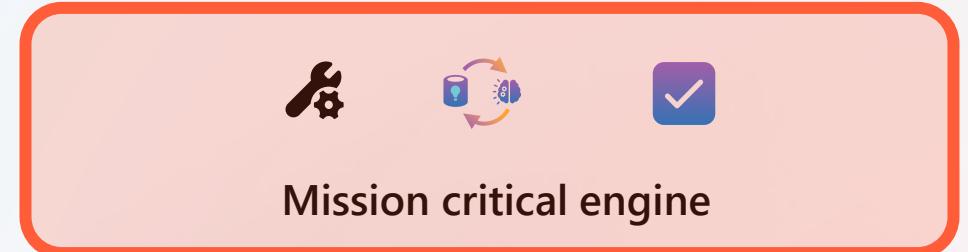
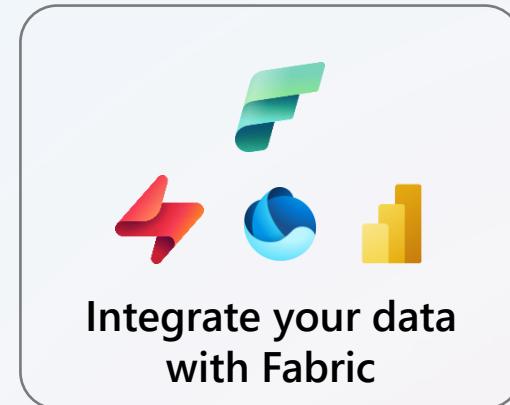
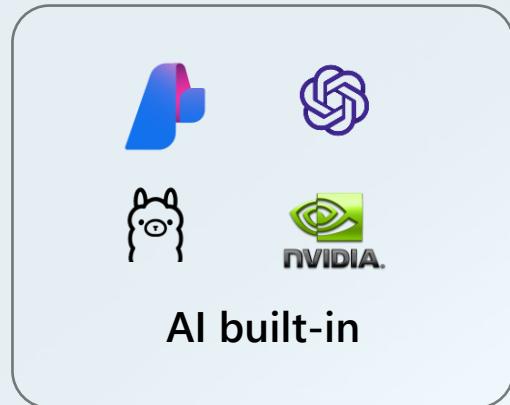
in linkedin.com/in/PawelPotasinski

FAVORITE STUFF:



SQL Server 2025

The AI-ready enterprise database from ground to cloud



Enhancing the Industry Proven Engine



Security

- Security cache improvements
- OAEP support for encryption
- PBKDF password hashing
- Authentication using system-assigned managed identity
- Backup to URL with managed identity
- Managed identity support for EKM
- Managed identity for AI models
- Entra logins with nonunique display names
- Custom password policy on Linux
- TDS 8.0/TLS 1.3 support for tools and features



Performance

- Optimized Locking
- Tempdb space resource governance
- ADR in tempdb
- Persisted stats for readable secondaries
- Change tracking cleanup
- Columnstore index maintenance
- CE feedback for expressions
- Optional parameter plans optimization
- DOP feedback on by default
- Query store for readable secondaries
- ABORT_QUERY_EXECUTION query hint
- Optimized sp_executesql
- Batch mode optimizations
- Remove In-Memory OLTP from a database
- tmpfs support for tempdb in Linux



HADR

- Fast failover for persistent AG Health
- Async page request dispatching
- Improved health diagnostics
- Communication control flow tuning
- Switching to resolving state
- Remove listener IP address
- NONE for routing
- AG group commit waiting tuning
- Contained AG support for DAG
- DAG sync improvements
- Backups on secondary replicas
- ZSTD Backup compression
- Backup to Azure immutable storage

Enhancing the Industry Proven Engine



Security

- Security cache improvements
- OAEP support for encryption
- PBKDF password hashing
- Authentication using system-assigned managed identity
- Backup to URL with managed identity
- Managed identity support for EKM
- Managed identity for AI models
- Entra logins with nonunique display names
- Custom password policy on Linux
- TDS 8.0/TLS 1.3 support for tools and features



Performance

- Optimized Locking
- Tempdb space resource governance
- ADR in tempdb
- Persisted stats for readable secondaries
- Change tracking cleanup
- Columnstore index maintenance
- CE feedback for expressions
- Optional parameter plans optimization
- DOP feedback on by default
- Query store for readable secondaries
- ABORT_QUERY_EXECUTION query hint
- Optimized sp_executesql
- Batch mode optimizations
- Remove In-Memory OLTP from a database
- tmpfs support for tempdb in Linux



HADR

- Fast failover for persistent AG Health
- Async page request dispatching
- Improved health diagnostics
- Communication control flow tuning
- Switching to resolving state
- Remove listener IP address
- NONE for routing
- AG group commit waiting tuning
- Contained AG support for DAG
- DAG sync improvements
- Backups on secondary replicas
- ZSTD Backup compression
- Backup to Azure immutable storage

Mission Critical Engine

Performance and Availability for your modern database



Improve Concurrency

- Optimized Locking
- Abort Query Hint
- Tempdb resource governance

Accelerate Performance

- IQP enhancements
- Columnstore indexes
- Query store on read replicas

Increase HADR

- Reliable Failover for AGs
- AG Tuning and diagnostics
- Backup enhancements



Industry proven engine

Mission Critical Engine

Performance and Availability for your modern database



Improve Concurrency

Optimized Locking

Abort Query Hint

Tempdb resource governance

Accelerate Performance

IQP enhancements

Columnstore indexes

Query store on read replicas

Increase HADR

Reliable Failover for AGs

AG Tuning and diagnostics

Backup enhancements



Industry proven engine

Improve Concurrency



Optimized Locking

Eliminate lock escalation
Lock After Qualification (LAQ)
Requires ADR, RCSI, and db option



ABORT_QUERY_EXECUTION query hint

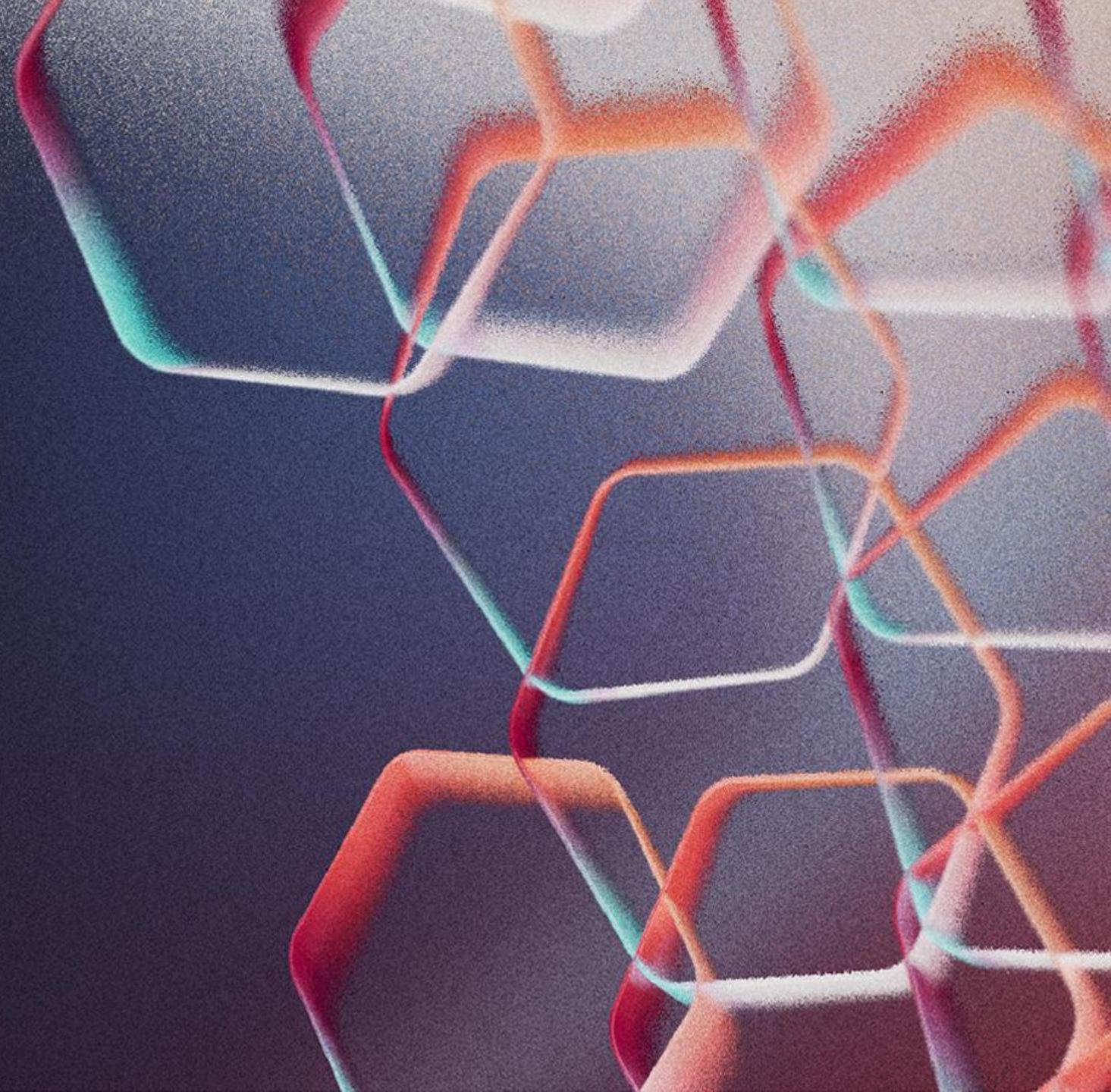
Abort future query execution
Stop rogue queries from harming other users
Query Store Hint



Tempdb Resource Governance

Tempdb space usage can grow out of control
Need to control and isolate users
Flexibility to change policy over time.
Fixed size or %

DEMO:
Optimized locking



DEMO: ABORT_QUERY_EXECUTION



DEMO: Tempdb Resource Governance



Mission Critical Engine

Performance and Availability for your modern database



Improve Concurrency

- Optimized Locking
- Abort Query Hint
- Tempdb resource governance

Accelerate Performance

- IQP enhancements
- Columnstore indexes
- Query store on read replicas

Increase HADR

- Reliable Failover for AGs
- AG Tuning and diagnostics
- Backup enhancements



Industry proven engine



Backup Enhancements

ZSTD compression
algorithm for backups

```
BACKUP DATABASE [AdventureWorks2025]
TO DISK = N'D:\Temp\AW-ZSTD.bak'
WITH COMPRESSION (ALGORITHM = ZSTD);
```

Compression algorithm
server configuration

```
EXECUTE sp_configure 'backup compression algorithm', 3;
RECONFIGURE;
```

DEMO: ZSTD Backup Compression



Mission Critical Engine

Performance and Availability for your modern database



Improve Concurrency

- Optimized Locking
- Abort Query Hint
- Tempdb resource governance

Accelerate Performance

- IQP enhancements
- Columnstore indexes
- Query store on read replicas

Increase HADR

- Reliable Failover for AGs
- AG Tuning and diagnostics
- Backup enhancements



Industry proven engine

Columnstore Indexes

Ordered nonclustered columnstore indexes

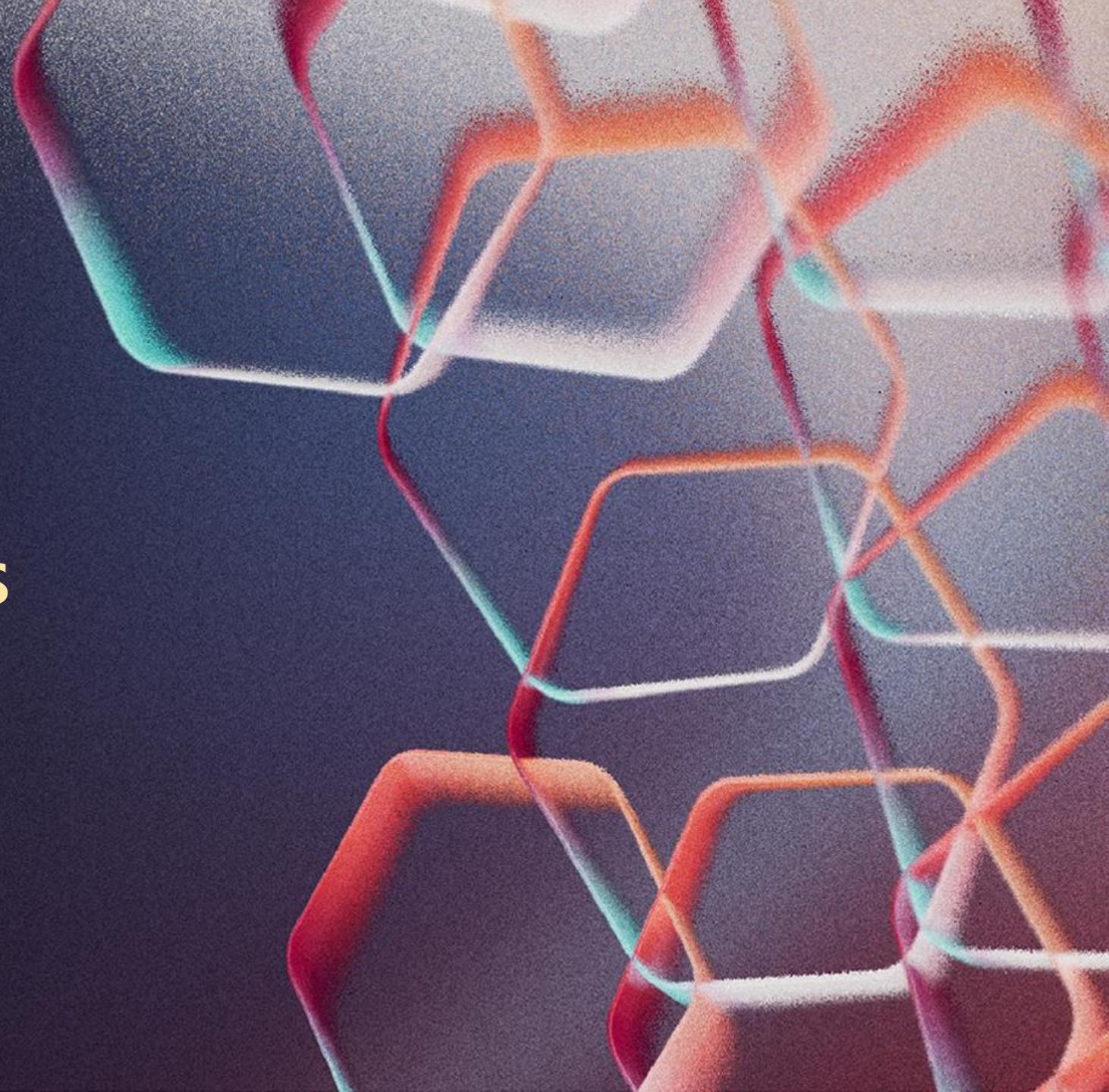
```
CREATE NONCLUSTERED COLUMNSTORE INDEX NCCI_SalesOrdersBIG  
ON Sales.OrdersBIG (OrderDate, ExpectedDeliveryDate, BackorderOrderID)  
ORDER (OrderDate);
```

Online operations on ordered columnstore index

```
CREATE NONCLUSTERED COLUMNSTORE INDEX NCCI_SalesOrdersBIG  
ON Sales.OrdersBIG (OrderDate, ExpectedDeliveryDate, BackorderOrderID)  
ORDER (OrderDate)  
WITH (DROP_EXISTING = ON, MAXDOP = 1, ONLINE = ON);
```

DEMO:

Columnstore indexes



Other Performance Improvements

Optimized `sp_executesql`

Honor Compile locks and prevent plan cache bloat

Optional parameter plan optimization (OPPO)

Extend PSPO to handle NULLs

Accelerated Database Recovery (ADR) in tempdb

Get all the benefits of ADR for temp object transactions

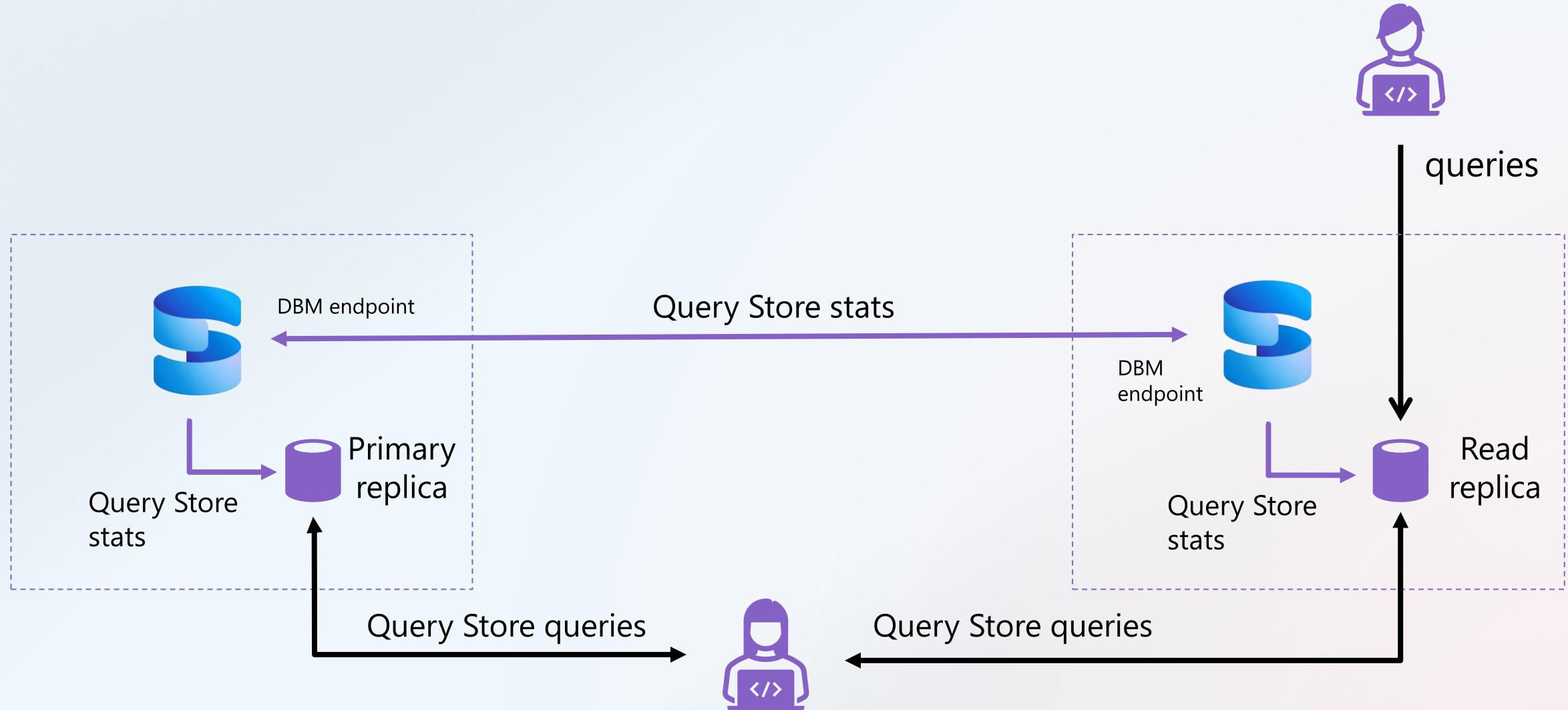
Query store for readable secondaries

Empower performance analysis on all replicas

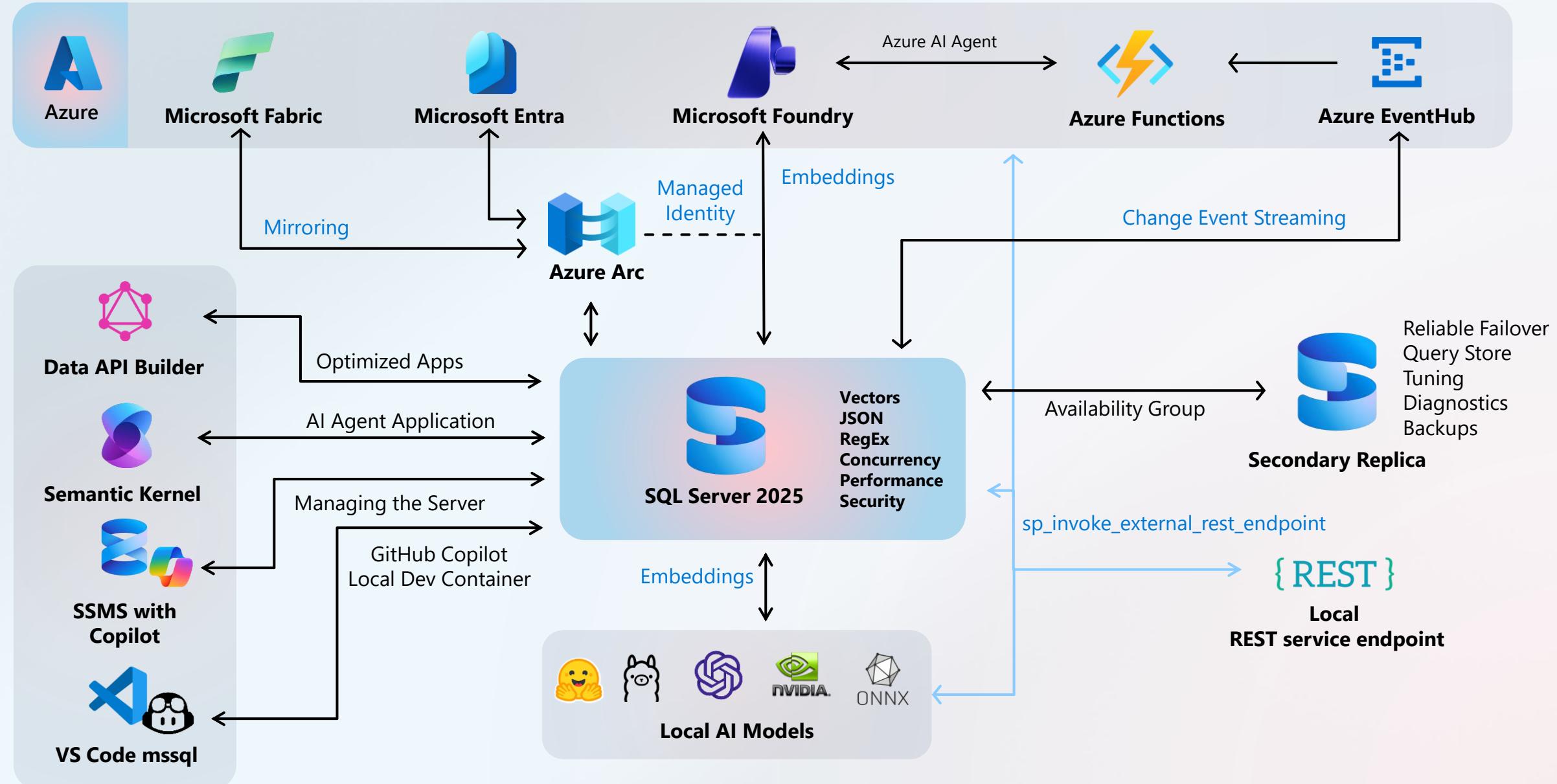
Persisted stats for readable secondaries

Predictable stats on all readable secondaries using query store channel

Query Store for Read Replicas



The SQL Server 2025 Platform Architecture



Call to Action



Check out blogs

aka.ms/sqlserver2025

aka.ms/Build/sql2025blog



Download it today

aka.ms/getsqlserver2025



Read the docs

aka.ms/sqlserver2025docs



Get the decks

aka.ms/sqlserver2025decks



Get the demos

aka.ms/sqlserver2025demos



Share your feedback

aka.ms/sqlfeedback

https://aka.ms/ssms-feedback



Learn from videos

aka.ms/Build/sql2025mechanics

aka.ms/SQL2025videos



Upgrade to the new SSMS 22 and Copilot

aka.ms/ssms



Be more efficient with GitHub Copilot

aka.ms/vscode-mssql-copilot