

When and how to ... migrate to Azure Synapse?

Kamil Nowinski

Principal Microsoft Consultant



altius



Kamil Nowiński



Microsoft Data Platform MVP
Speaker, blogger, data enthusiast
Principal Microsoft Consultant at Altius (www.altiusdata.com)
20 years of experience as DEV/BI/(DBA)
Member of the Data Community PL
Project member of „SCD Merge Wizard”
„azure.datafactory.tools” & „azure.datafactory.devops”
Founder of blog SQLPlayer (www.SQLplayer.net)

SQL Server Certificates:
MCITP, MCP, MCTS, MCSA, MCSE Data Platform,
MCSE Data Management & Analytics
Moreover: Bicycle, Running, Digital photography
@NowinskiK, @SQLPlayer

SQLBITS
2020

Blog



Scan me

www.SQLPlayer.net

- Technical posts
- Various skill level
- Cheet sheets
- Recommended books
- Many useful other links
- YouTube Channel ←
- Interviews (Podcast)



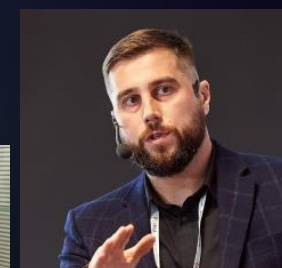
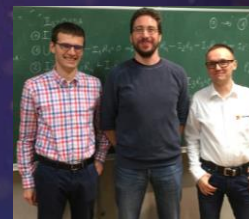
LBITS
2020

“Ask SQL Family” – interviews with...

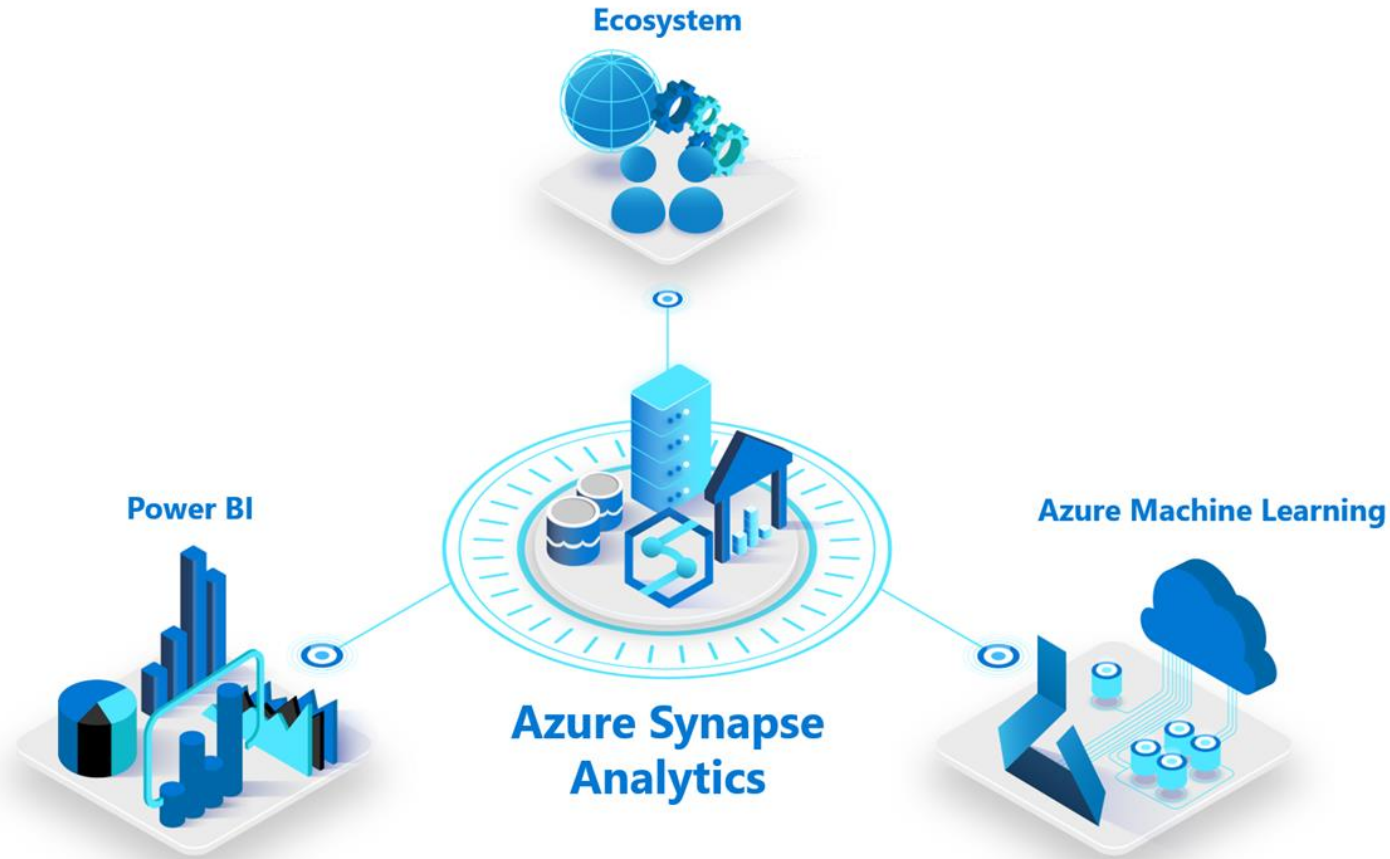


Scan me

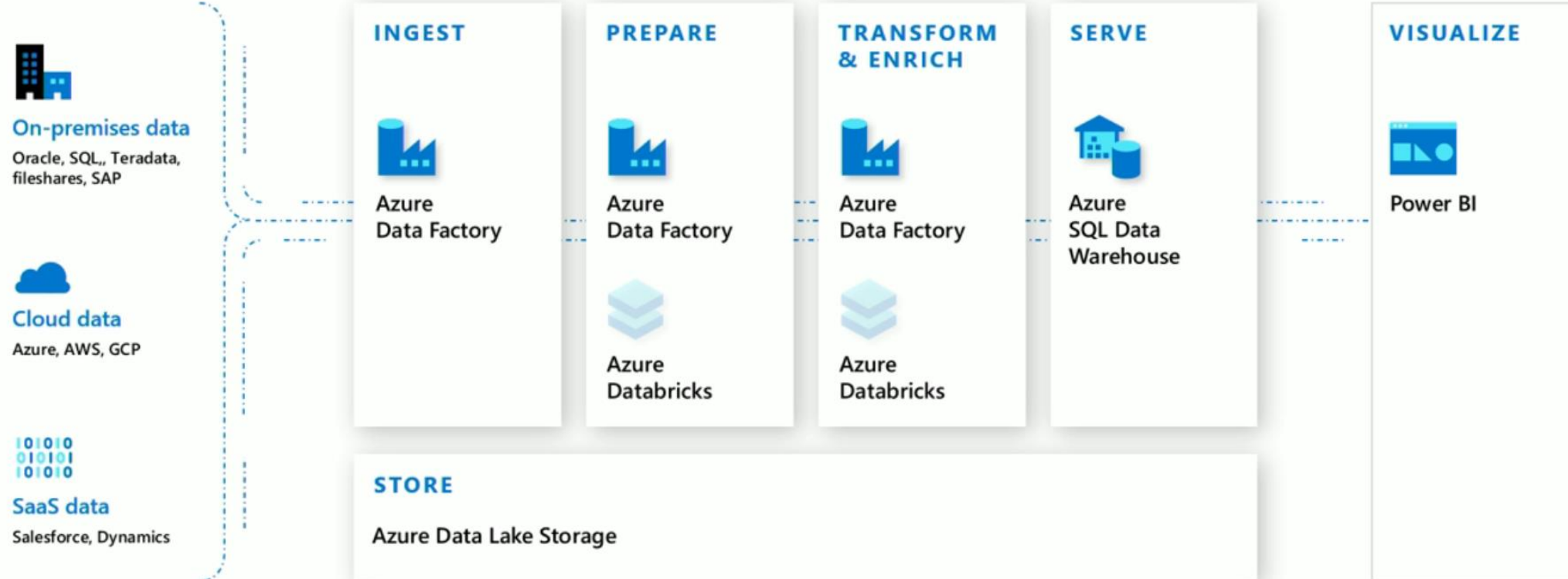
SQLBITS 2020



Azure Synapse is Azure SQL Data Warehouse evolved



Modern Data Warehouse



Azure Synapse Analytics - *Data Lakehouse*



On-premises data

Oracle, SQL, Teradata,
fileshares, SAP



Cloud data

Azure, AWS, GCP



SaaS data

Salesforce, Dynamics



Azure Synapse Analytics

STORE

Azure Data Lake Storage

VISUALIZE



Power BI

Multiple analytics platforms

Big Data



Data
Lake

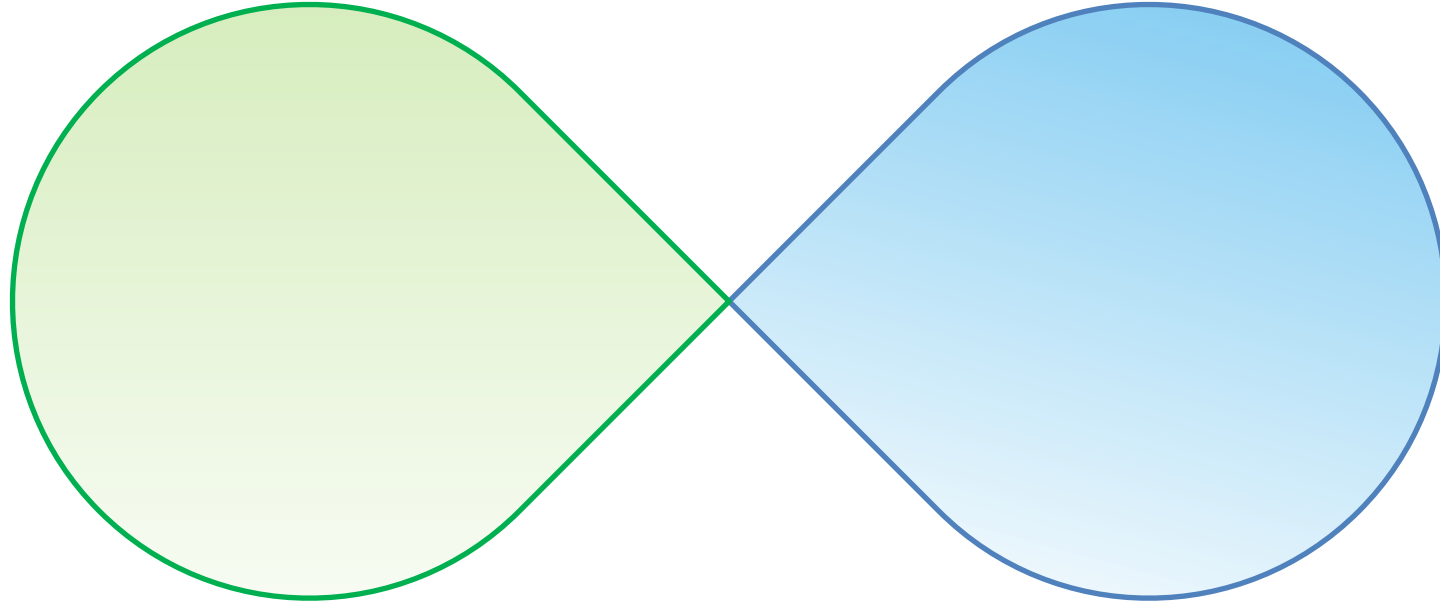
OR

Relational Data



Data
Warehouse

Azure brings these two worlds together



Welcome to Azure Synapse Analytics

Data warehousing & big data analytics - all in one service

Parallelism

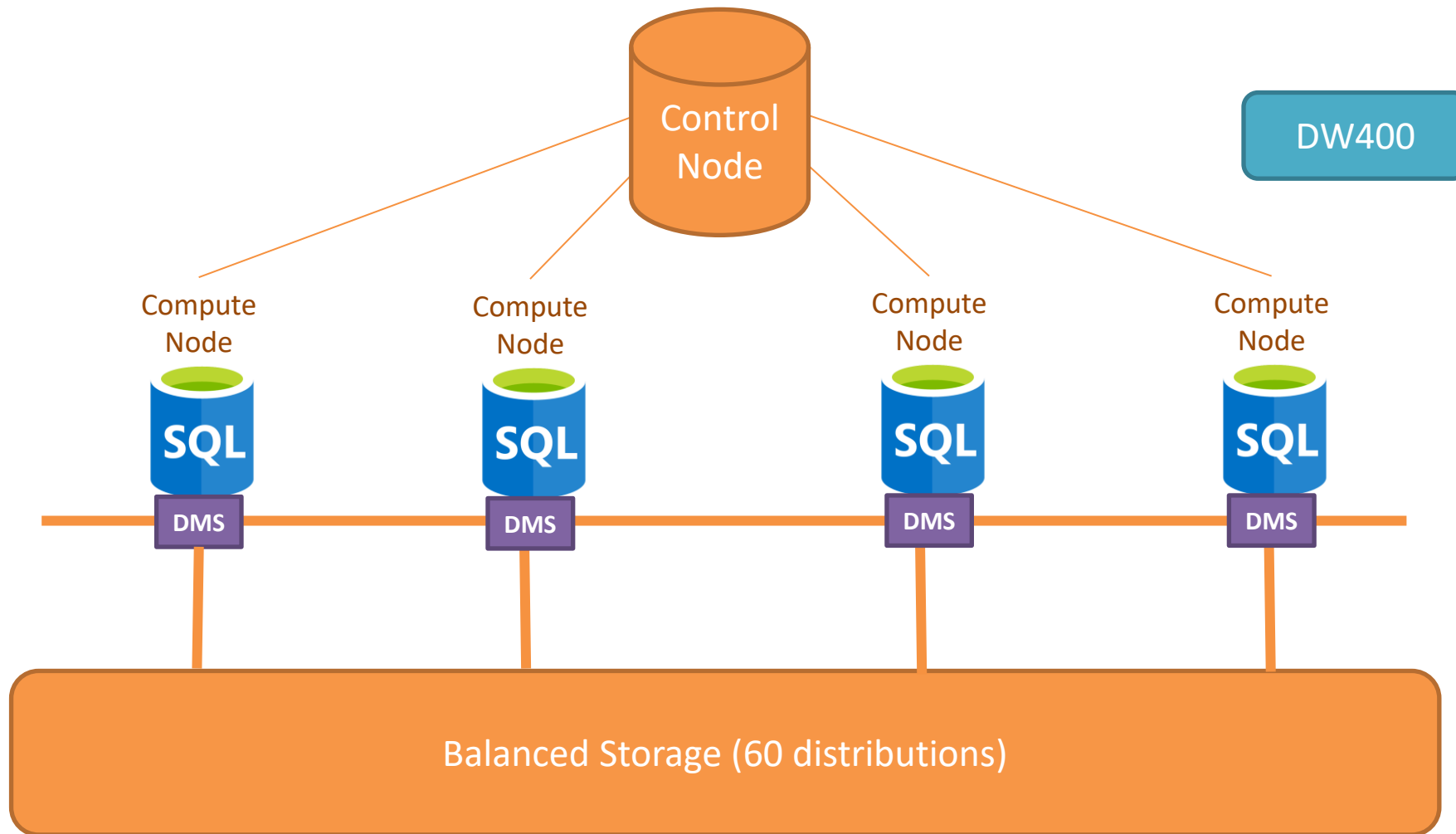
SMP – Symmetric Multiprocessing

- Multiple CPUs used to complete individual processes simultaneously
- All CPUs share the same memory, disks, and network controllers (scale-up)
- All SQL Server implementations up until now have been SMP
- Mostly, the solution is housed on a shared SAN

MPP – Massively Parallel Processing

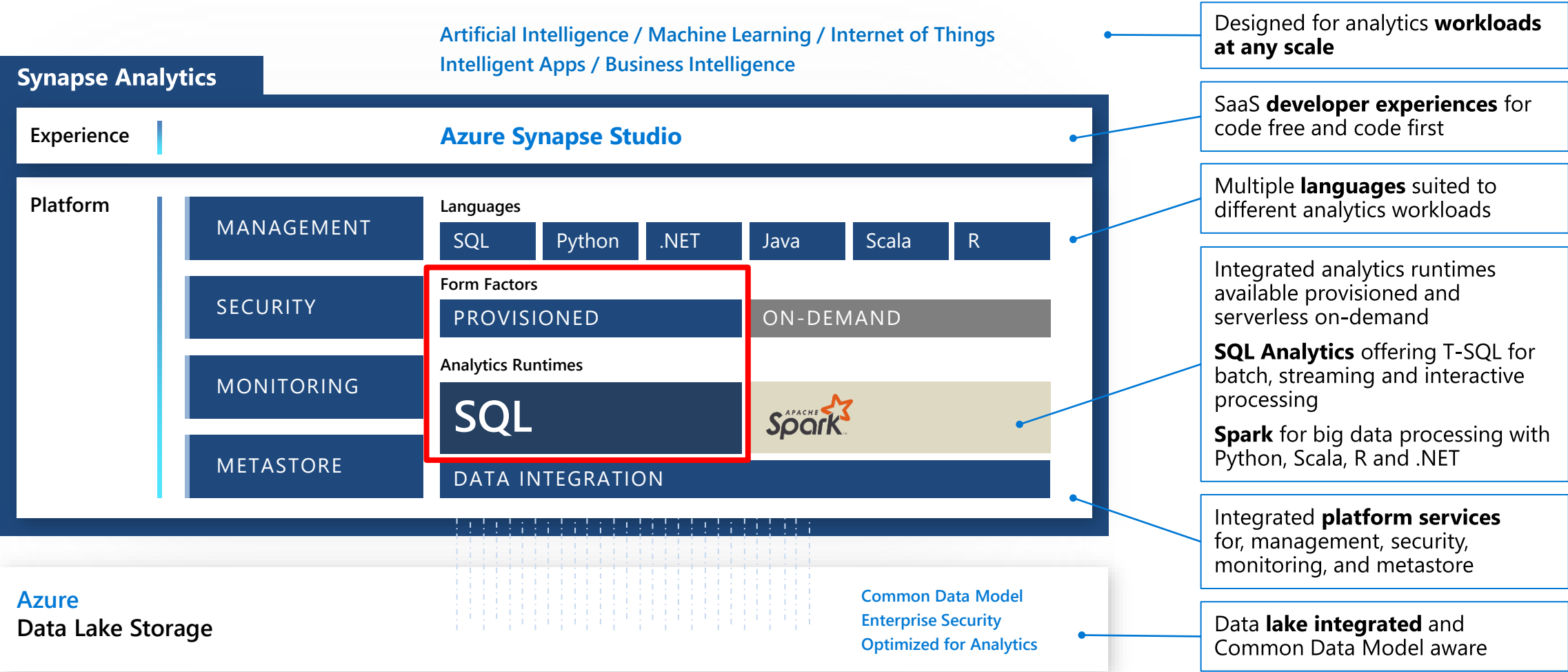
- Uses many separate CPUs running in parallel to execute a single program
- Shared Nothing: Each CPU has its own memory and disk (scale-out)
- Segments communicate using high-speed network between nodes

Synapse Analytics Logical Architecture



Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Key features

Rich surface area

- T-SQL language for data analytics
- Supporting large number of languages and tools
- Enterprise-grade security

SQL Provisioned

- Modern Data Warehouse
- Indexing and caching
- Import and query external data
- Workload management

SQL Serverless

- Querying external data
- Model raw files as virtual tables and views
- Easy data transformation

STORAGE



Table Distribution Options

- ROUND ROBIN
- HASH
- REPLICATED

Tables – Distributions

Round-robin distributed

Distributes table rows evenly across all distributions at random.

Hash distributed

Distributes table rows across the Compute nodes by using a deterministic hash function to assign each row to one distribution.

Replicated

Full copy of table accessible on each Compute node.

```
CREATE TABLE dbo.OrderTable
(
    OrderId INT NOT NULL,
    Date DATE NOT NULL,
    Name VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]) |
                                ROUND ROBIN |
                                REPLICATED
);
```

Tables – Partitions

Overview

Table partitions divide data into smaller groups

In most cases, partitions are created on a date column

Supported on all table types

RANGE RIGHT – Used for time partitions

RANGE LEFT – Used for number partitions

Benefits

Improves efficiency and performance of loading and querying by limiting the scope to subset of data.

Offers significant query performance enhancements where filtering on the partition key can eliminate unnecessary scans and eliminate IO.

```
CREATE TABLE partitionedOrderTable
(
    OrderId INT NOT NULL,
    Date DATE NOT NULL,
    Name VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]),
    PARTITION (
        [Date] RANGE RIGHT FOR VALUES (
            '2000-01-01', '2001-01-01', '2002-01-01',
            '2003-01-01', '2004-01-01', '2005-01-01'
        )
    )
);
```


Tables – Distributions & Partitions

Logical table structure

OrderId	Date	Name	Country
85016	11-2-2018	V	UK
85018	11-2-2018	Q	SP
85216	11-2-2018	Q	DE
85395	11-2-2018	V	NL
82147	11-2-2018	Q	FR
86881	11-2-2018	D	UK
93080	11-3-2018	R	UK
94156	11-3-2018	S	FR
96250	11-3-2018	Q	NL
98799	11-3-2018	R	NL
98015	11-3-2018	T	UK
98310	11-3-2018	D	DE
98979	11-3-2018	Z	DE
98137	11-3-2018	T	FR
...

Physical data distribution

(Hash distribution (OrderId), Date partitions)

Distribution1

(OrderId 80,000 – 100,000)

11-2-2018 partition

OrderId	Date	Name	Country
85016	11-2-2018	V	UK
85018	11-2-2018	Q	SP
85216	11-2-2018	Q	DE
85395	11-2-2018	V	NL
82147	11-2-2018	Q	FR
86881	11-2-2018	D	UK
...

11-3-2018 partition

OrderId	Date	Name	Country
93080	11-3-2018	R	UK
94156	11-3-2018	S	FR
96250	11-3-2018	Q	NL
98799	11-3-2018	R	NL
98015	11-3-2018	T	UK
98310	11-3-2018	D	DE
98979	11-3-2018	Z	DE
98137	11-3-2018	T	FR
...

...

x 60 distributions (shards)

Common table distribution methods

Table Category	Recommended Distribution Option
Fact	<p>Use hash-distribution with clustered columnstore index. Performance improves because hashing enables the platform to localize certain operations within the node itself during query execution.</p> <p>Operations that benefit:</p> <p>COUNT(DISTINCT(<hashed_key>))</p> <p>OVER PARTITION BY <hashed_key></p> <p>most JOIN <table_name> ON <hashed_key></p> <p>GROUP BY <hashed_key></p>
Dimension	<p>Use replicated for smaller tables. If tables are too large to store on each Compute node, use hash-distributed.</p>
Staging	<p>Use round-robin for the staging table. The load with CTAS is faster. Once the data is in the staging table, use INSERT...SELECT to move the data to production tables.</p>

Table Distribution Options: ROUND ROBIN

1	Poland
2	Germany
8	UK
...	
66	Switzerland
70	Ireland

DB1



DB2



DB3



...

DB60



altius

Table Distribution Options: HASH

1	Poland
2	Germany
8	UK
...	
66	Switzerland
70	Ireland

DB1



DB2



DB3



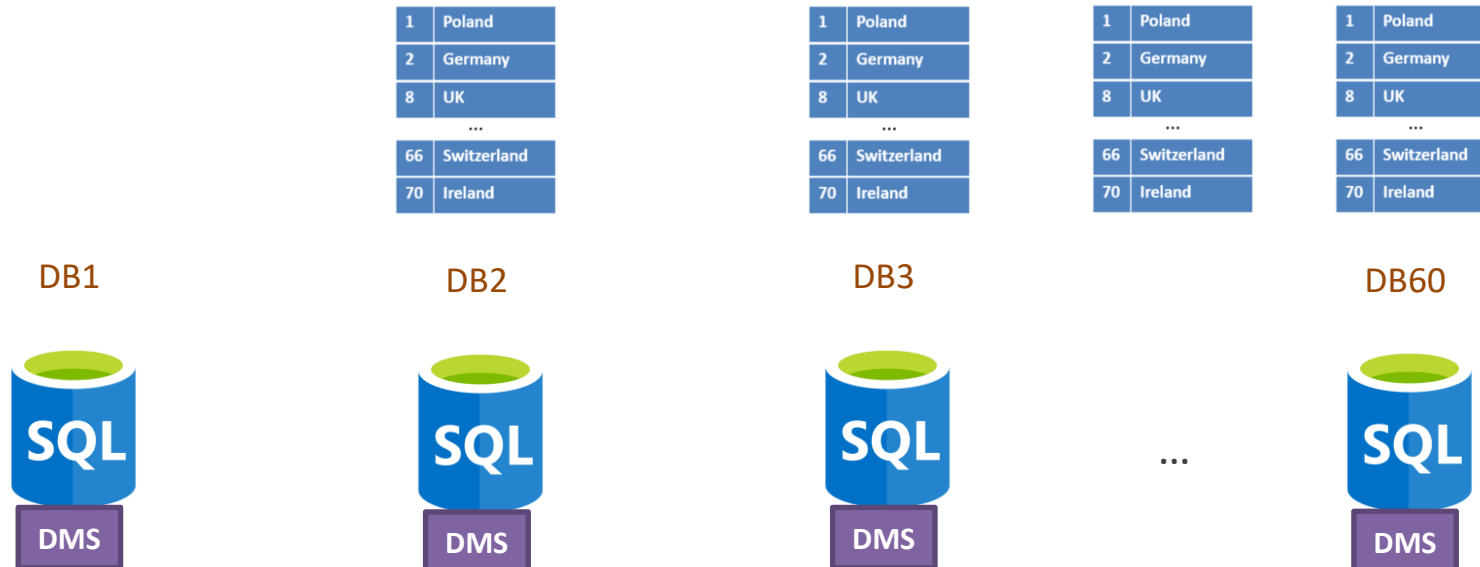
...

DB60



Table Distribution Options: REPLICATED

1	Poland
2	Germany
8	UK
...	
66	Switzerland
70	Ireland



altius

Execution Plan – DMS Operations

DMS Operation	Description
ShuffleMoveOperation	Distribution → Hash algorithm → New distribution Changing the distribution column in preparation for join.
PartitionMoveOperation	Distribution → Control Node Aggregations - count(*) is count on nodes, sum of count
BroadcastMoveOperation	Distribution → Copy to all distributions Changes distributed table to replicated table for join.
TrimMoveOperation	Replicated table → Hash algorithm → Distribution When a replicated table needs to become distributed. Needed for outer joins.
MoveOperation	Control Node → Copy to all distributions Data moved from Control Node back to Compute Nodes resulting in a replicated table for further processing.
RoundRobinMoveOperation HadoopRoundRobinMoveOperation	Source → Round robin algorithm → Distribution Redistributes data to Round Robin Table.

What Workloads are Suitable?

Analytics

- Store large volumes of data.
- Consolidate disparate data into a single location.
- Shape, model, transform and aggregate data.
- Batch/Micro-batch loads.
- Perform query analysis across large datasets.
- Ad-hoc reporting across large data volumes.
- All using simple SQL constructs.

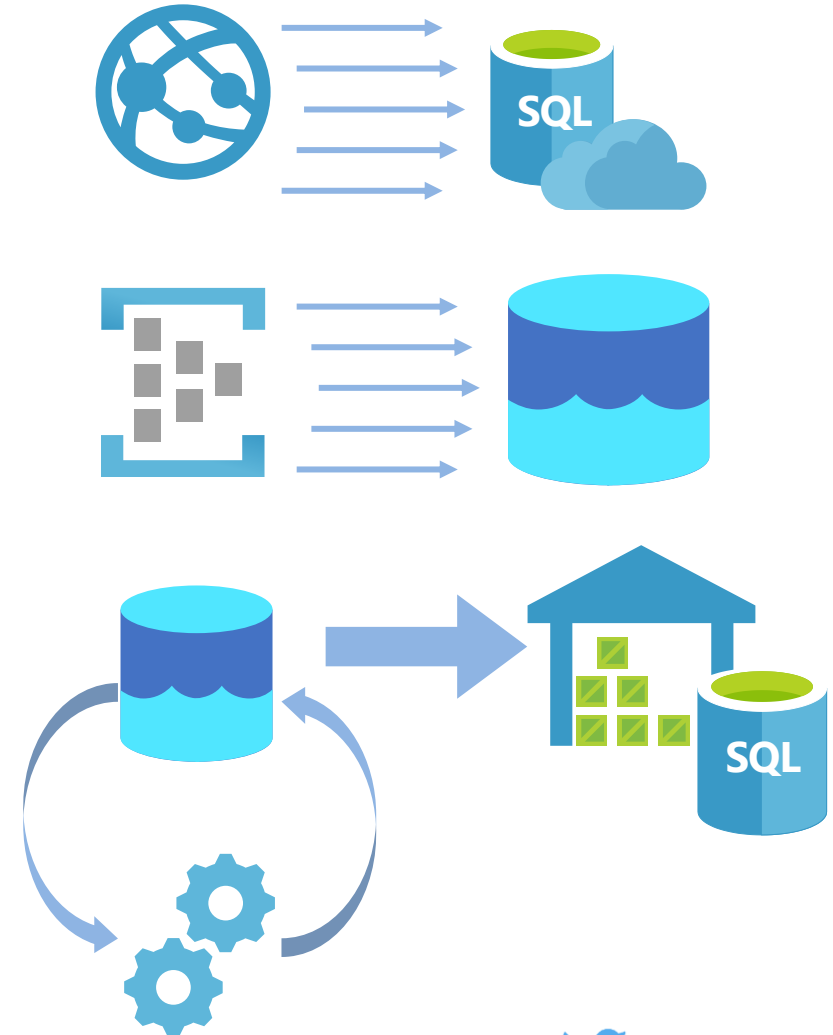
What workloads are NOT suitable?

Operational workloads (OLTP)

- High frequency reads and writes
- Large numbers of singleton selects
- High volumes of single row inserts

Data Preparations

- Row by row processing needs
- Incompatible formats (XML)



Important things

- SQL DW is based on an MPP architecture (not SMP)
 - The same engine under hood, but scale and concurrency are vary
- SIZE does really matter
- Individual table size and rowcount are important
- OLTP reporting type workloads are usually poor candidates
- Proper schema design – **important** in SQL Server
- Right schema desing – **CRITICAL** in SQL DW



Azure Synapse Analytics

Data Distribution DEMO





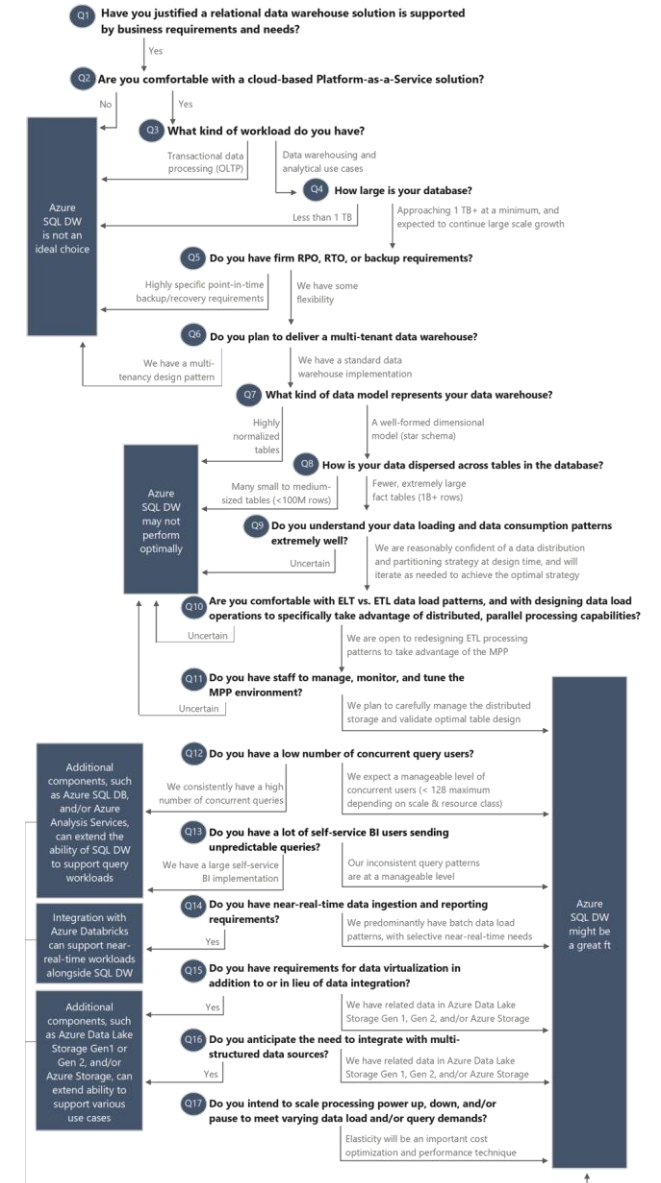
Azure Synapse Analytics Migration



Is Azure Synapse a good fit?

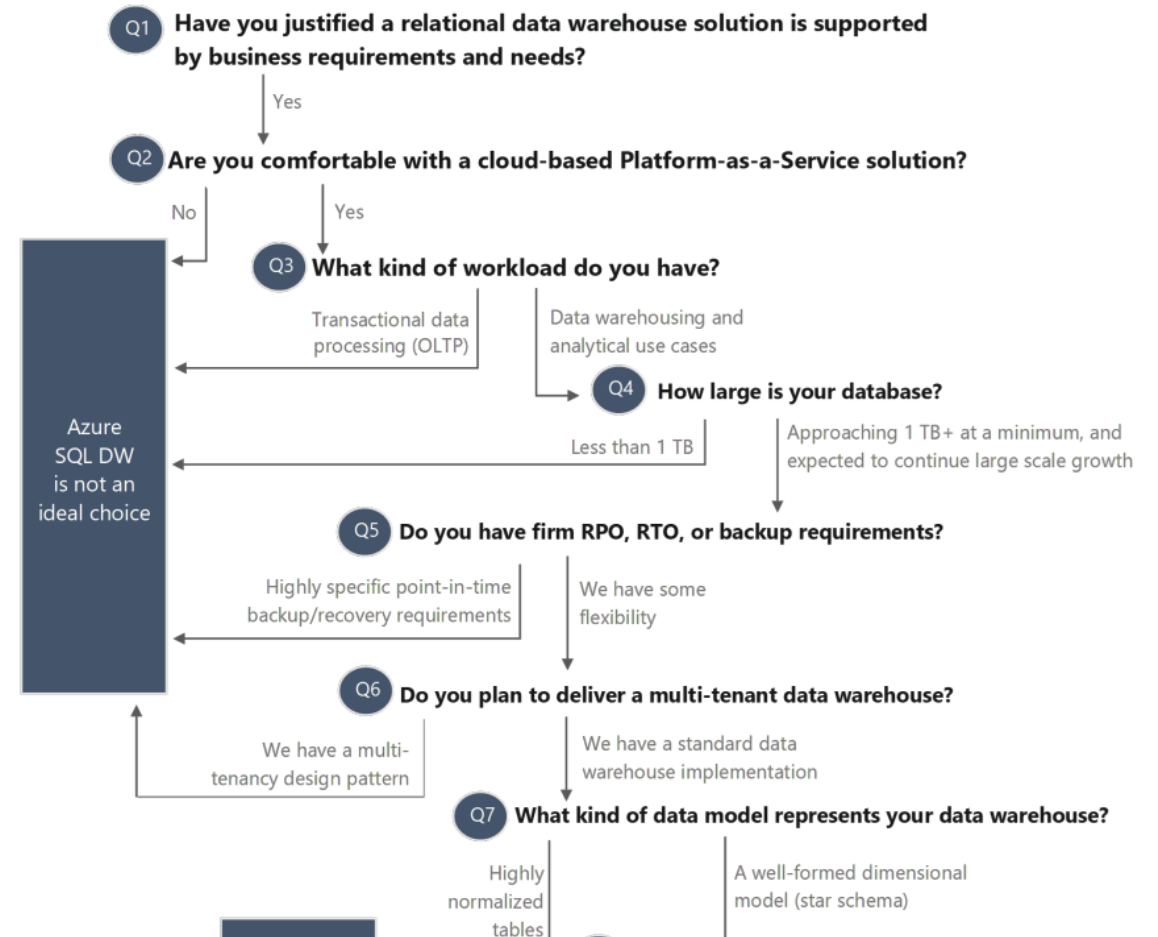
- Verify your source database in many aspects
- Do answer for many questions
- Use form from more experienced
- Questions' diagram
- Ask **Melissa Coates**

<https://www.blue-granite.com/blog/is-azure-sql-data-warehouse-a-good-fit-updated>



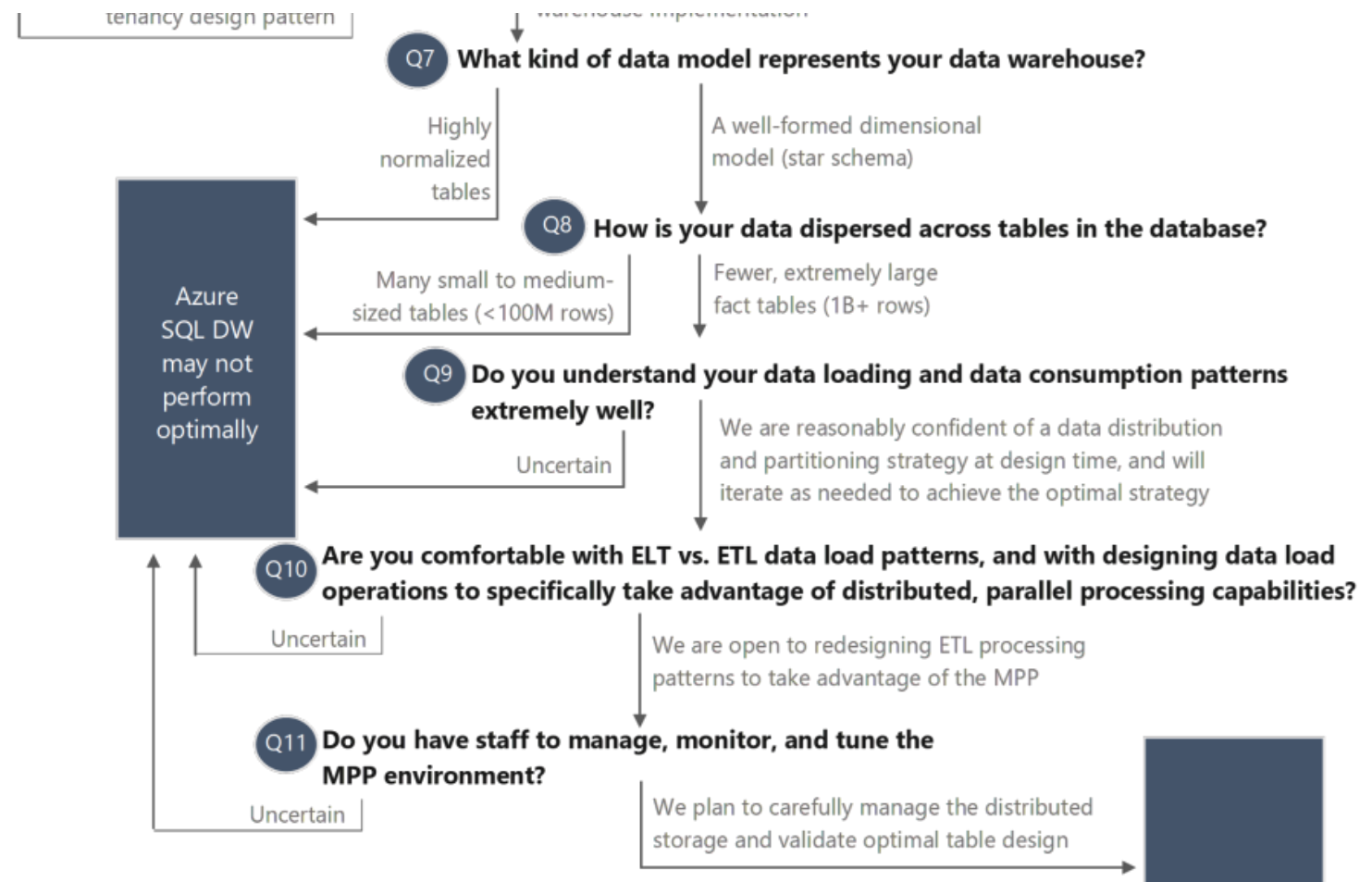
Is Azure SQL Data Warehouse a good fit? technology choice for your implementation?

- Q3
 - OLTP?
 - DW / Analytical workload?
- Q4
 - <1 TB?
 - >1 TB



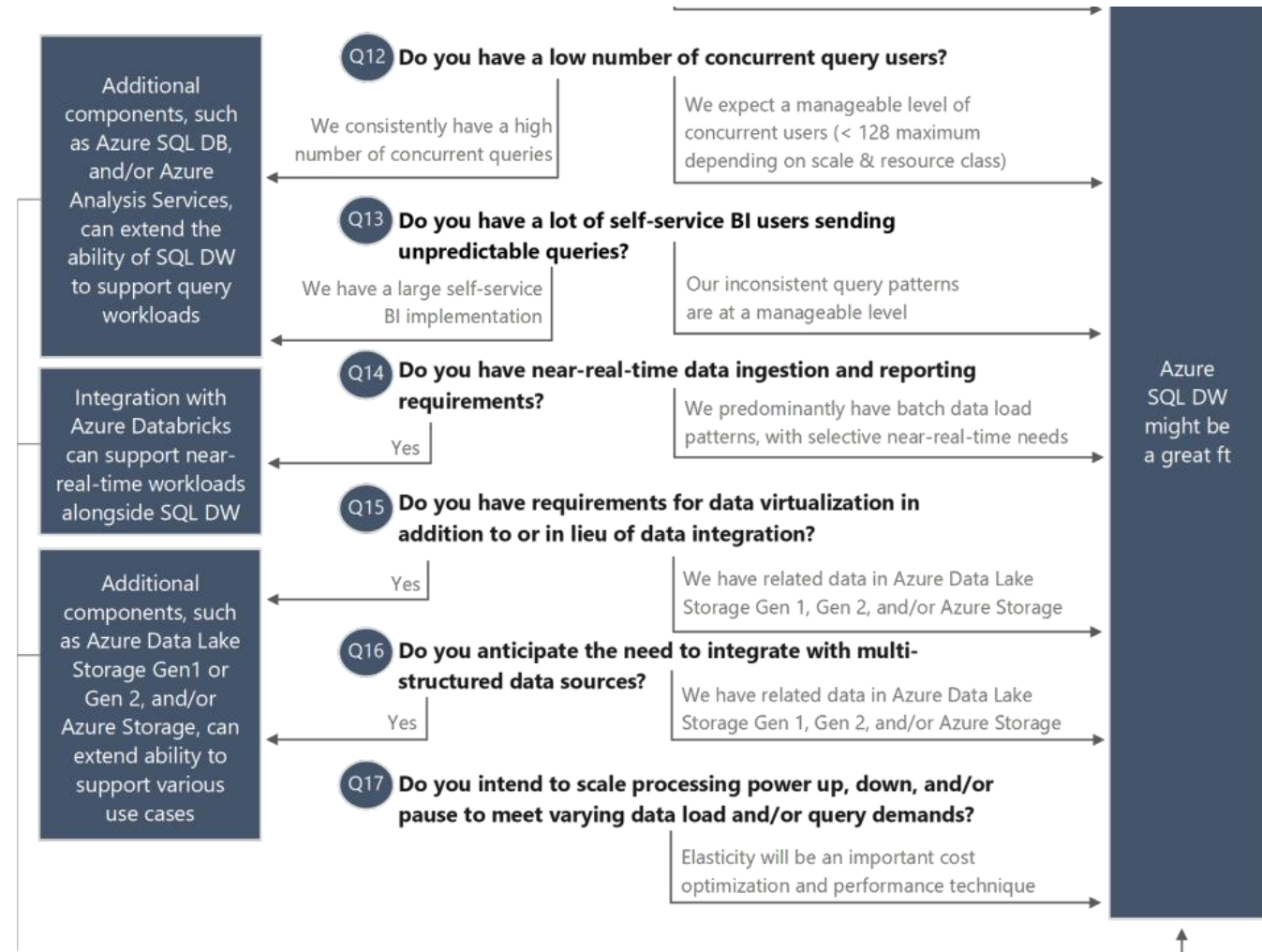
Is Azure SQL Data Warehouse the best technology choice for your implementation?

- Q7
 - Highly normalized tables?
 - Well-formed / star schema
- Q8: Number of tables & rows?
 - Many small/medium tables?
 - Fewer / large fact tables (1B+ rows)
- Q11: Skilled stuff



Is Azure SQL Data Warehouse the best technology choice for your implementation?

- Q12: Concurrent queries
 - High number
 - <128 active sessions
- Q14: Frequency of ingestion?
 - Near-real-time





Azure Synapse Analytics

Preparation & Copy



Data Preparation: files

- Filter essential objects to migrate
- Create performant local storage to receive exported data
- Establish standard or dedicated connectivity to cloud
- Choose region nearest to you with Azure SQL DW
- PolyBase: One folder per table in storage container

Data Migration Recommendations

- Data Format Conversion
 - Data Format, Field delimiters, Escaping, Field order, encoding
- Compression
 - Use Gzip, ORC, parquet
- Export
 - BCP for fast export
 - Multiple files per large table, one folder per table
- Copy
 - AZCopy
 - Data Movement Library

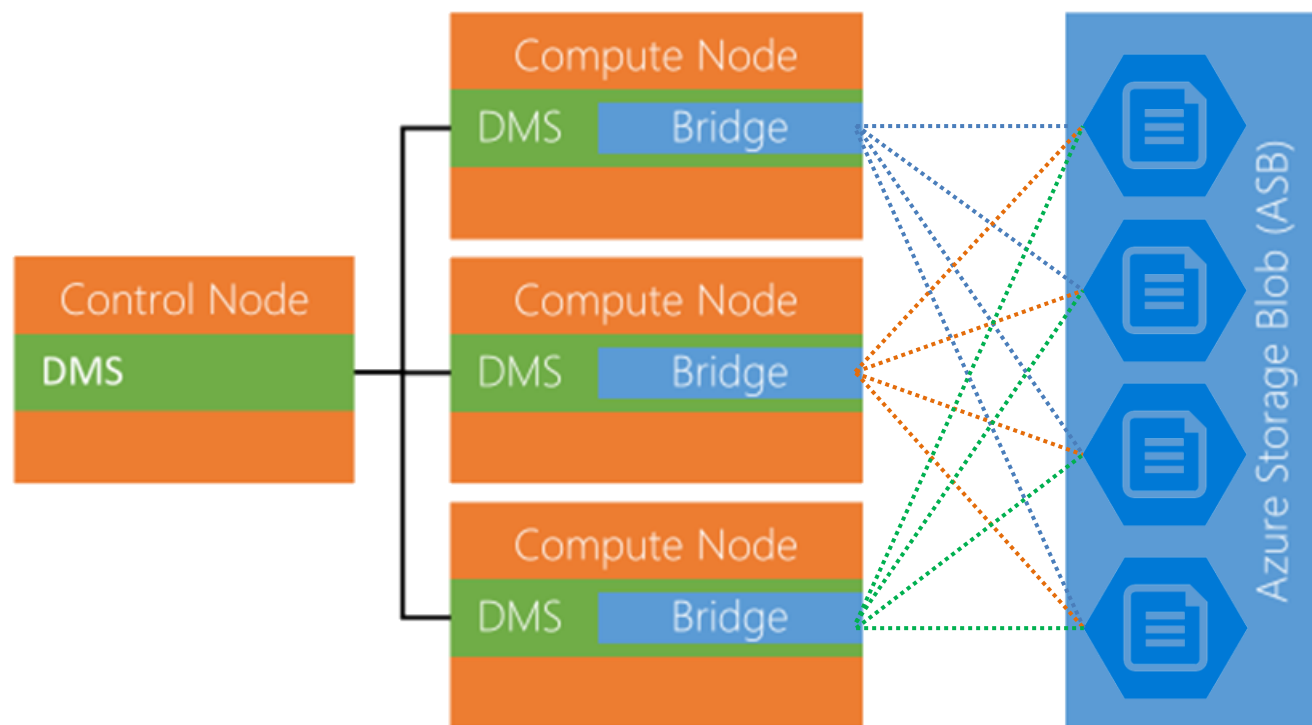
Data Migration Tips

- Incorrect format means migration needs to be entirely repeated
- Exploit bcp options, hints, parallelism
- Multiple compressed files, split files
- Parallel import, reliable transfer
- Don't use multiple files in the same gzipped file
- Efficient Copy
 - Parallel, Async, Resumable
 - Limit concurrent copies if low bandwidth
- Very large Data transfer
 - Express Route, Import/Export Service

Data Loading Recommendations

- PolyBase and SSIS (with 2017 Azure feature pack) the fastest method
 - Upload to BLOB via AZCOPY or PowerShell library
 - Historical load – use CTAS
 - Incremental – use INSERT...SELECT
 - UTF-8, UTF-16 also supports
- Use the highest resource class (without sacrificing concurrency)
- Increase DWU before load, decrease once done
- ADLS supported
- Doesn't support:
 - Extended ASCII
 - Custom multi-date format

Parallel Loading with PolyBase



PolyBase characteristics

- Single PolyBase load provides best performance for non-compressed files
- Load performance scales as you increase service level objective (SLO)
 - Number of files should be greater than or equal to the total number of readers of your service level objective (SLO)
- Automatically parallelizes data load process;
 - no need to manually break the input data into multiple files and issue concurrent loads
 - Each reader slice 512 MB block from data files
- Max throughput depends on number of readers available on the DWU level
- Multiple readers will not work against a compressed text file (gzip)
 - Only a single reader is used per compressed file since uncompressing the file in the buffer is single threaded
 - Alternatively, generate multiple compressed files



Azure Synapse Analytics

Parallel Loading with PolyBase

Resources



<http://sqlplayer.net/Synapse>

Thank you!



kamil@nowinski.net



@NowinskiK

@SQLPlayer



SQLPlayer.net



<https://github.com/NowinskiK/CommunityEvents>



Kamil Nowinski

Microsoft Data Platform MVP

MCSE Data Platform & MCSE Data Management and Analytics

Credits

Big credits to James Serra (Microsoft)
who shared and let me use some of his slides.

The background of the entire image is a dark blue gradient with a bokeh effect of golden light spots. Golden confetti, including small rectangular pieces and wavy streamers, is scattered across the top half of the image. The main title 'SQLBITS 2020' is centered in the upper half. 'SQLBITS' is in a large, bold, serif font with a golden gradient and a subtle drop shadow. Below it, '2020' is in a smaller, similar font, with a decorative flourish under the '2'.

SQLBITS 2020

THE GREATEST DATA SHOW