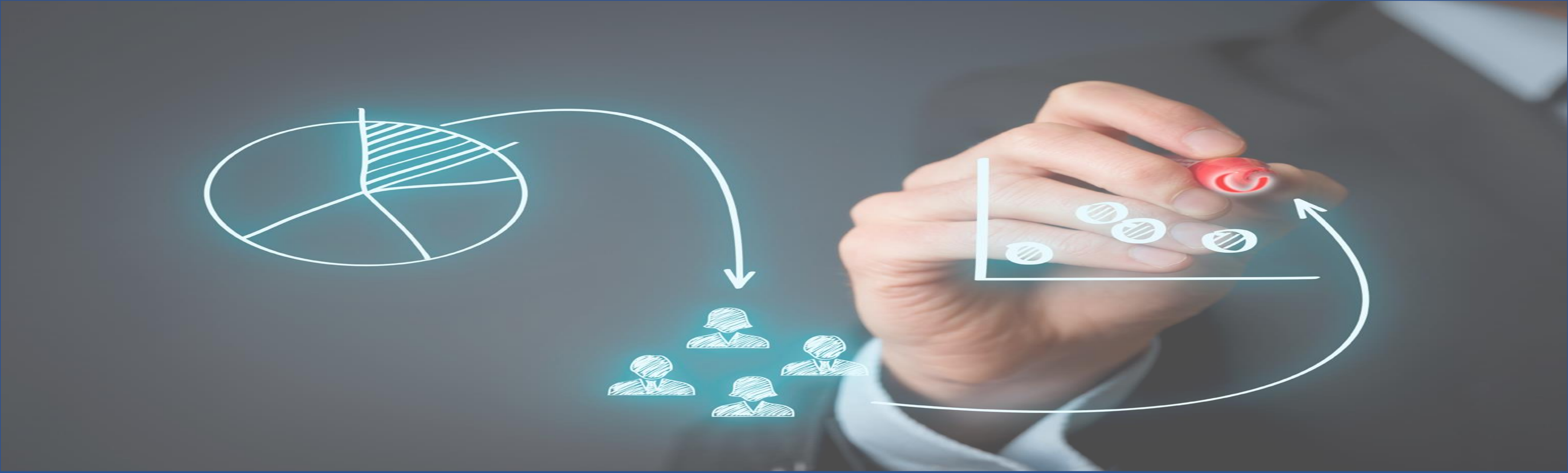


PROCESSING BIG DATA

with Azure Data Lake Analytics

Sean Forgatch
Business Intelligence Consultant
Sean.Forgatch@talavant.com

TALAVANT

About Me

Sean Forgatch

- Milwaukee, WI
- Business Intelligence Consultant
  - Healthcare, Insurance, SaaS
  - Integration and Analytics
  - Microsoft Big Data Certified
- PASS
  - Industry Speaker
  - FoxPASS President
- Running, Craft Beers, Reading

About Talavant

There is a better way to make data work for companies. Better resources, strategy, sustainability, inclusion of the organization as a whole, understanding of client needs, tools, outcomes, better ROI.

STRATEGY

ARCHITECTURE

IMPLEMENTATION

VALUE WE PROVIDE

- Accelerated planning, implementation and results
- Sustainable
- Increased

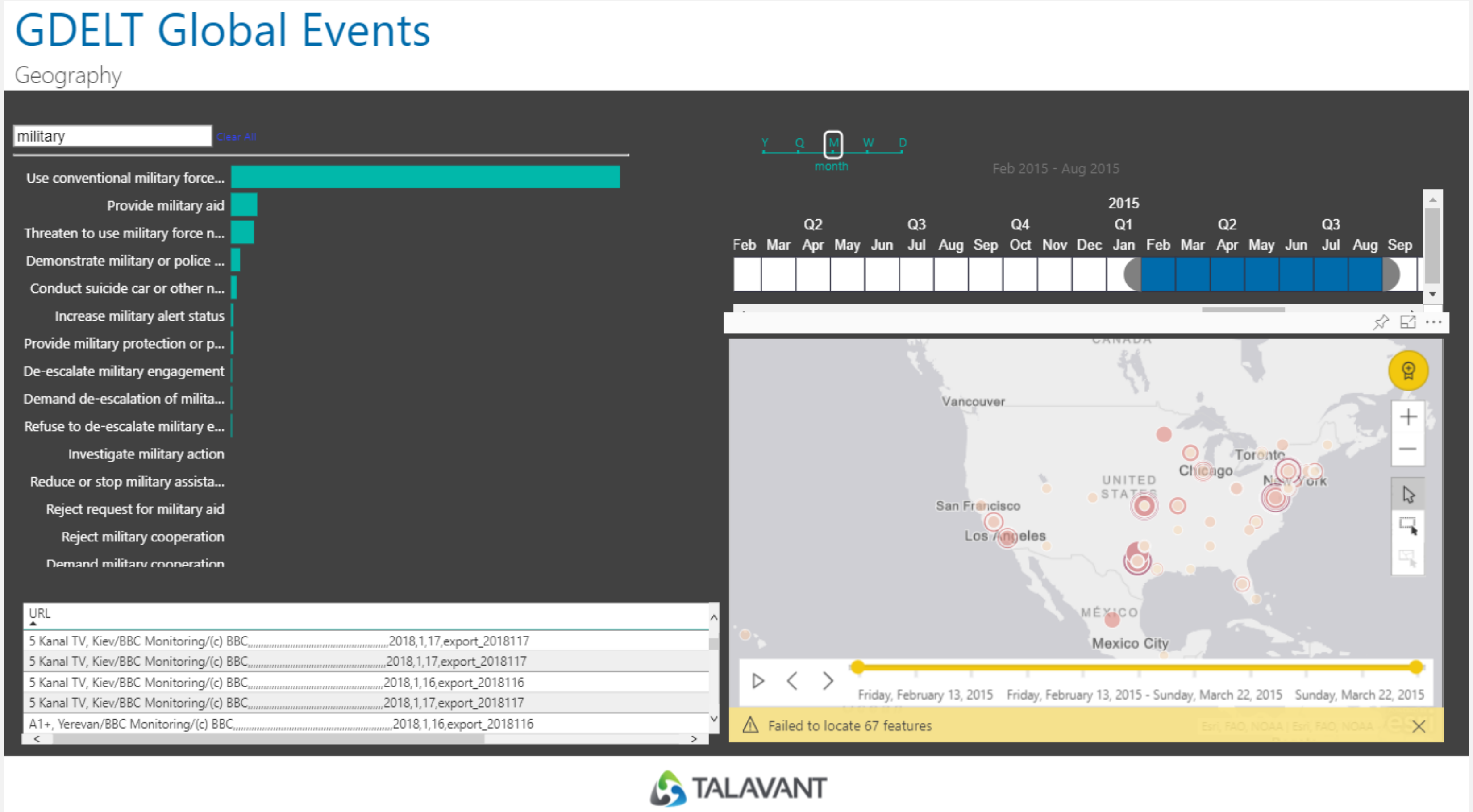
HOW WE DO IT

By providing a holistic approach inclusive of a client’s people, processes and technologies - built on investment in our own employees and company growth.

TALAVANT

1. **Big Data Overview**
2. Data Lake Concepts
3. Azure Data Lake Store
4. Azure Data Lake Analytics
5. U-SQL

GDELT Analysis in PowerBI



GDELT Global Events

Geography

military

Clear All

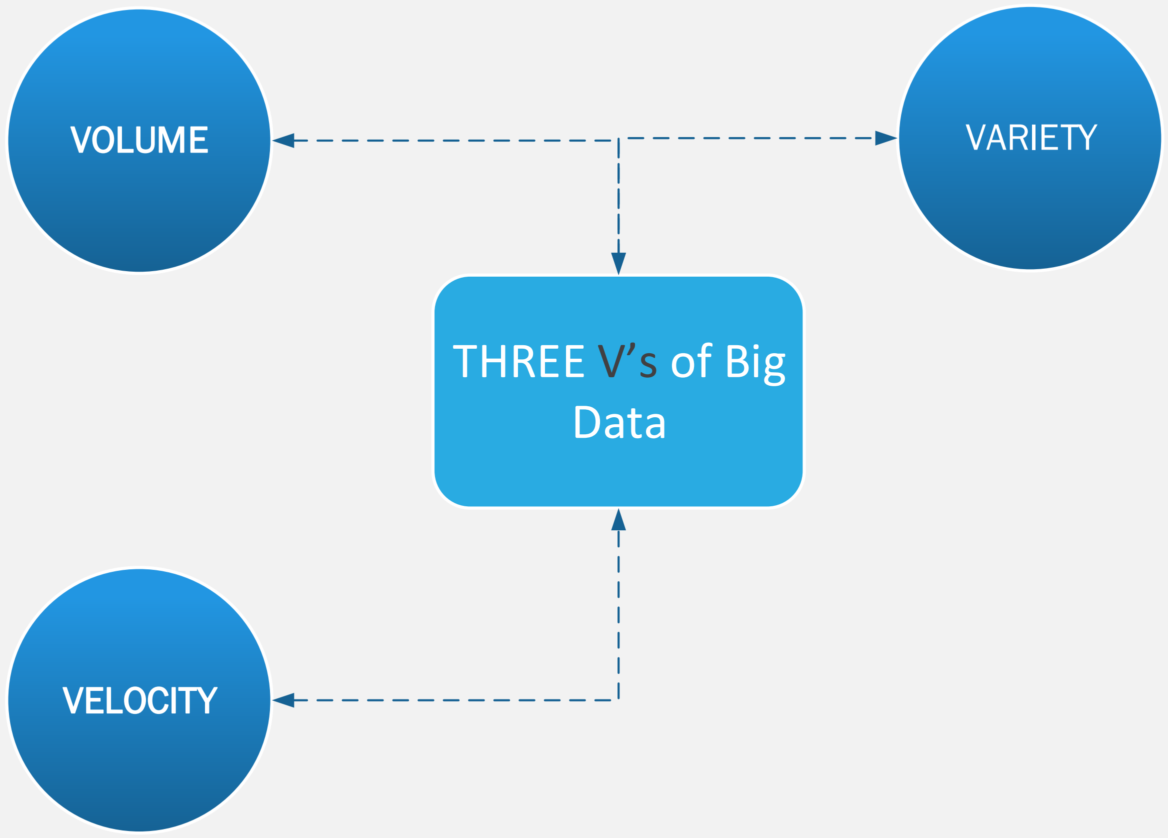
Use conventional military force...
Provide military aid
Threaten to use military force n...
Demonstrate military or police ...
Conduct suicide car or other n...
Increase military alert status
Provide military protection or p...
De-escalate military engagement
Demand de-escalation of milita...
Refuse to de-escalate military e...
Investigate military action
Reduce or stop military assista...
Reject request for military aid
Reject military cooperation
Demand military cooperation

URL
5 Kanal TV, Kiev/BBC Monitoring/(c) BBC,,,2018,1,17,export_2018117
5 Kanal TV, Kiev/BBC Monitoring/(c) BBC,,2018,1,17,export_2018117
5 Kanal TV, Kiev/BBC Monitoring/(c) BBC,,,2018,1,16,export_2018116
5 Kanal TV, Kiev/BBC Monitoring/(c) BBC,,,2018,1,17,export_2018117
A1+, Yerevan/BBC Monitoring/(c) BBC,,,2018,1,16,export_2018116

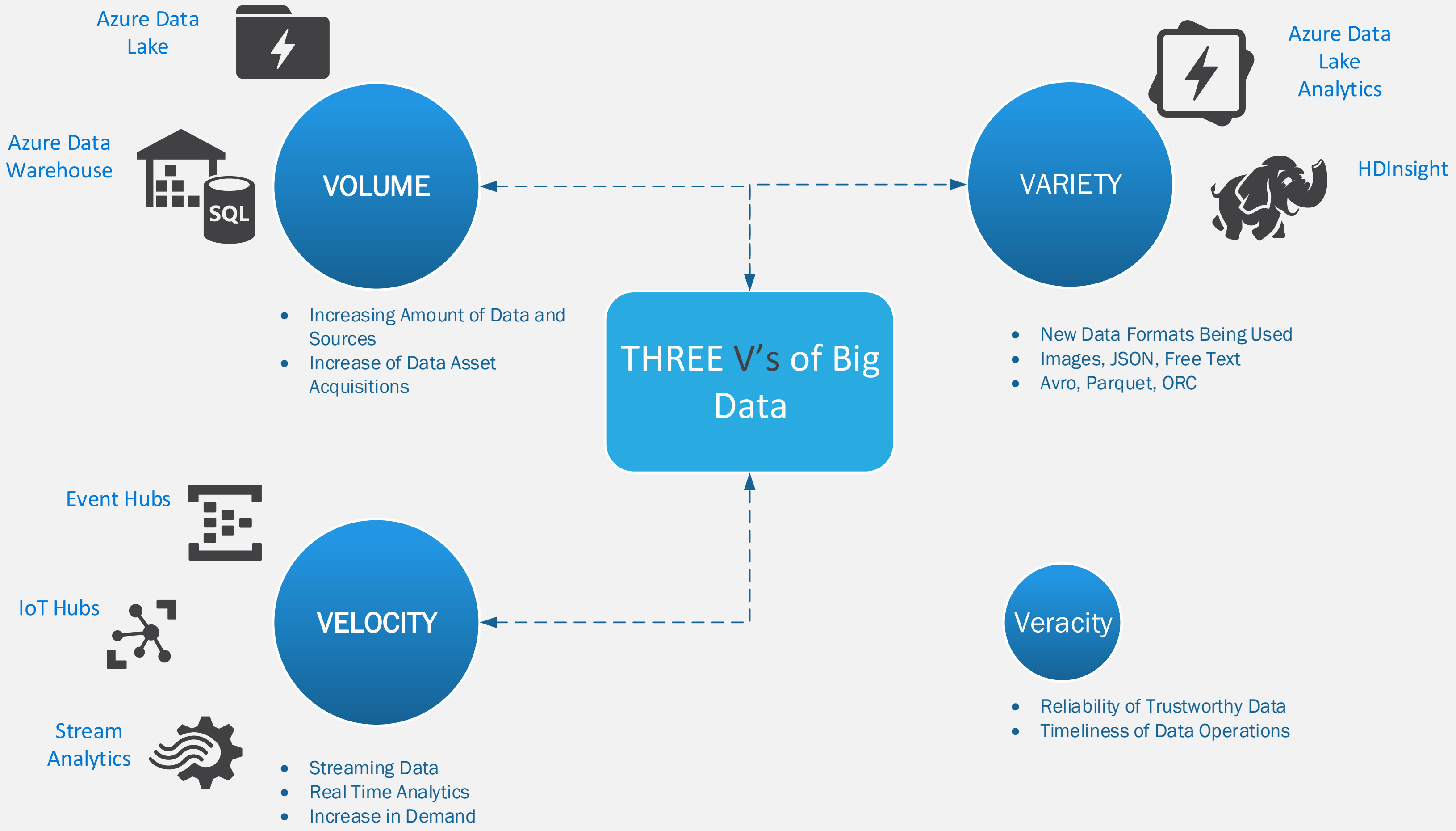
Y
Q
M
W
D
month
Feb 2015 - Aug 2015
2015
Q2
Q3
Q4
Q1
Q2
Q3
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
CANADA
Vancouver
UNITED STATES
Chicago
Toronto
New York
San Francisco
Los Angeles
MÉXICO
Mexico City
Friday, February 13, 2015
Friday, February 13, 2015 - Sunday, March 22, 2015
Sunday, March 22, 2015
Esri, FAO, NOAA | Esri, FAO, NOAA

⚠ Failed to locate 67 features

Big Data Primer



Big Data Primer: Azure Tools



- Increasing Amount of Data and Sources
- Increase of Data Asset Acquisitions

- New Data Formats Being Used
- Images, JSON, Free Text
- Avro, Parquet, ORC

- Streaming Data
- Real Time Analytics
- Increase in Demand

- Reliability of Trustworthy Data
- Timeliness of Data Operations

A Big Data Trends

“Variety, not volume or velocity, drives big-data investments”

A Big Data Trends

Variety, not volume or velocity, drives big-data investments"

"Big data grows up: Hadoop adds to enterprise standards"

A Big Data Trends

“Variety, not volume or velocity, drives big-data investments”

“Big data grows up: Hadoop adds to enterprise standards”

“Rise of metadata catalogs helps people find analysis-worthy big data.”

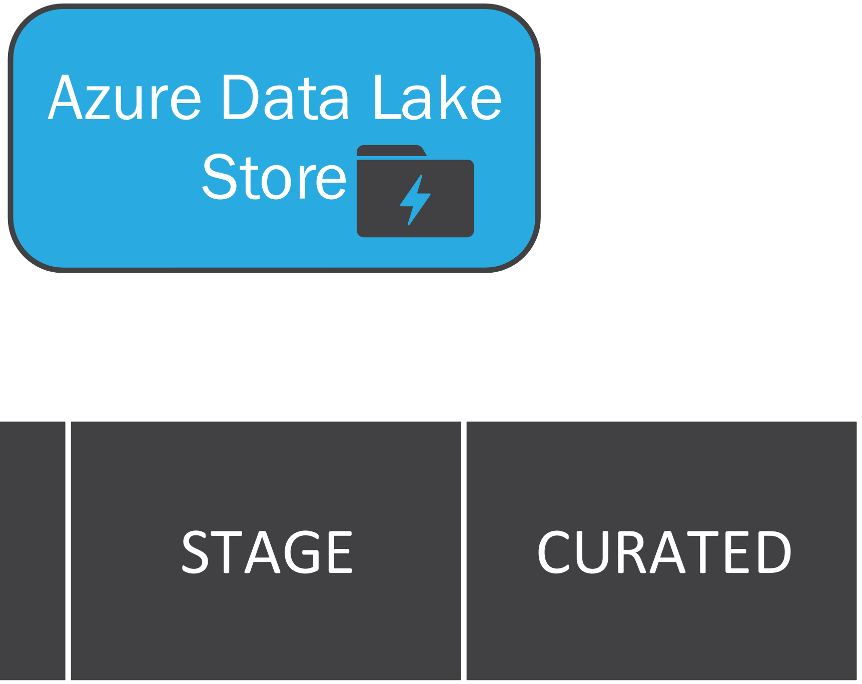
-TDWI: Top Ten Big Data Trends for 2017

1. Big Data Overview
2. **Data Lake Concepts**
3. Azure Data Lake Store
4. Azure Data Lake Analytics
5. U-SQL

Data Lake Concepts

TALAVANT

Data Lake Operations



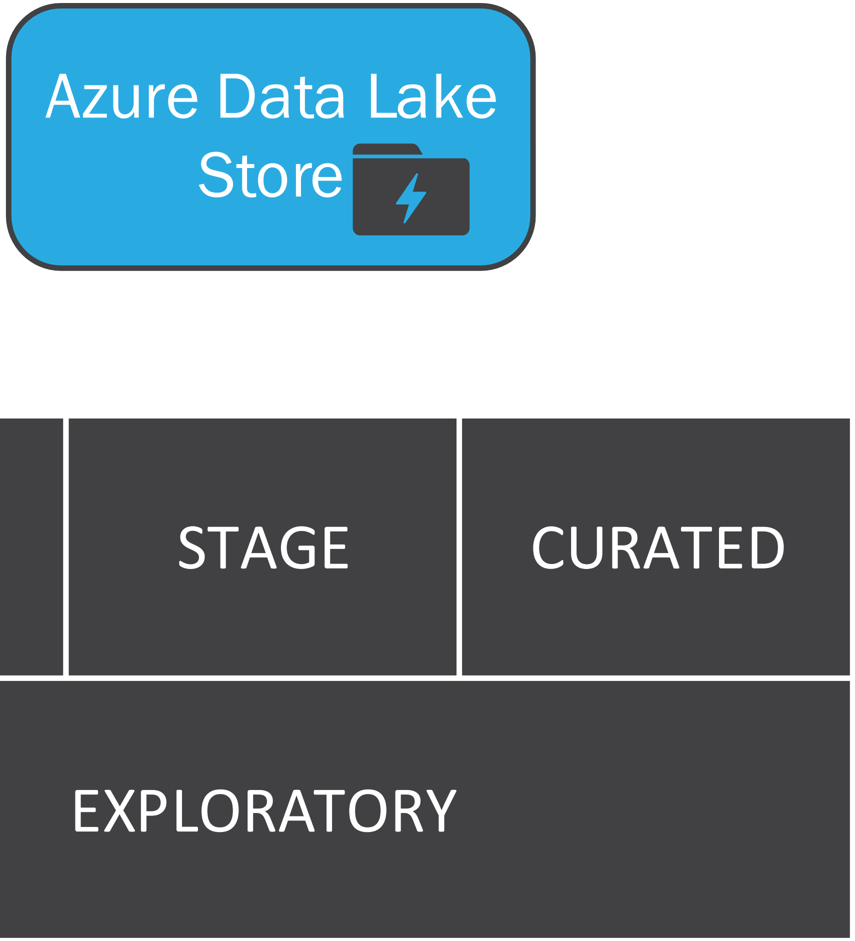
Azure Data Lake Store

RAW

STAGE

CURATED

Data Lake Operations



Azure Data Lake Store

RAW

STAGE

CURATED

EXPLORATORY

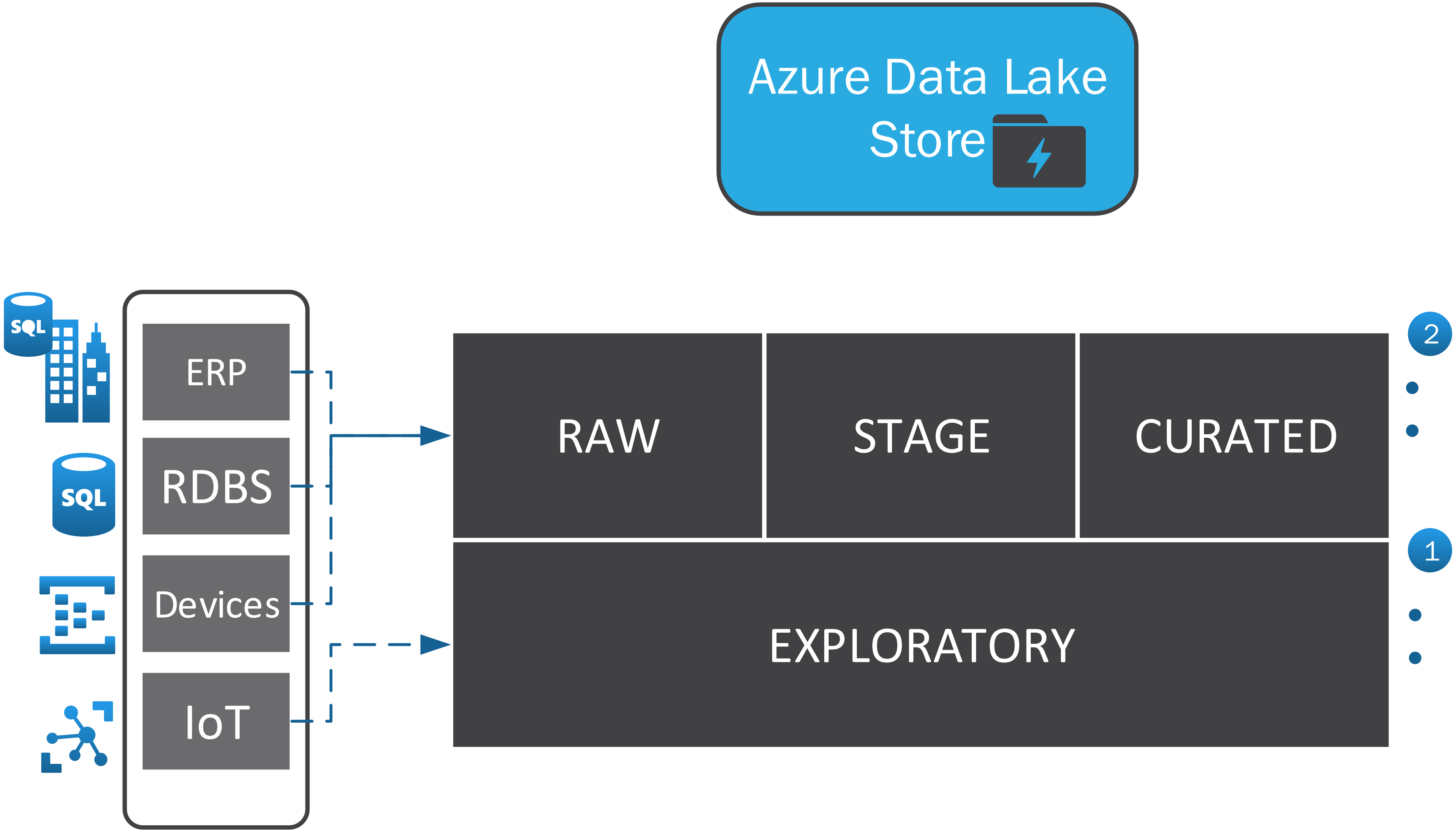
2

- **Operational**
- Value has been identified

1

- **Explorational**
- Value is being discovered

Data Lake Operations



Azure Data Lake Store

ERP

RDBS

Devices

IoT

RAW

STAGE

CURATED

EXPLORATORY

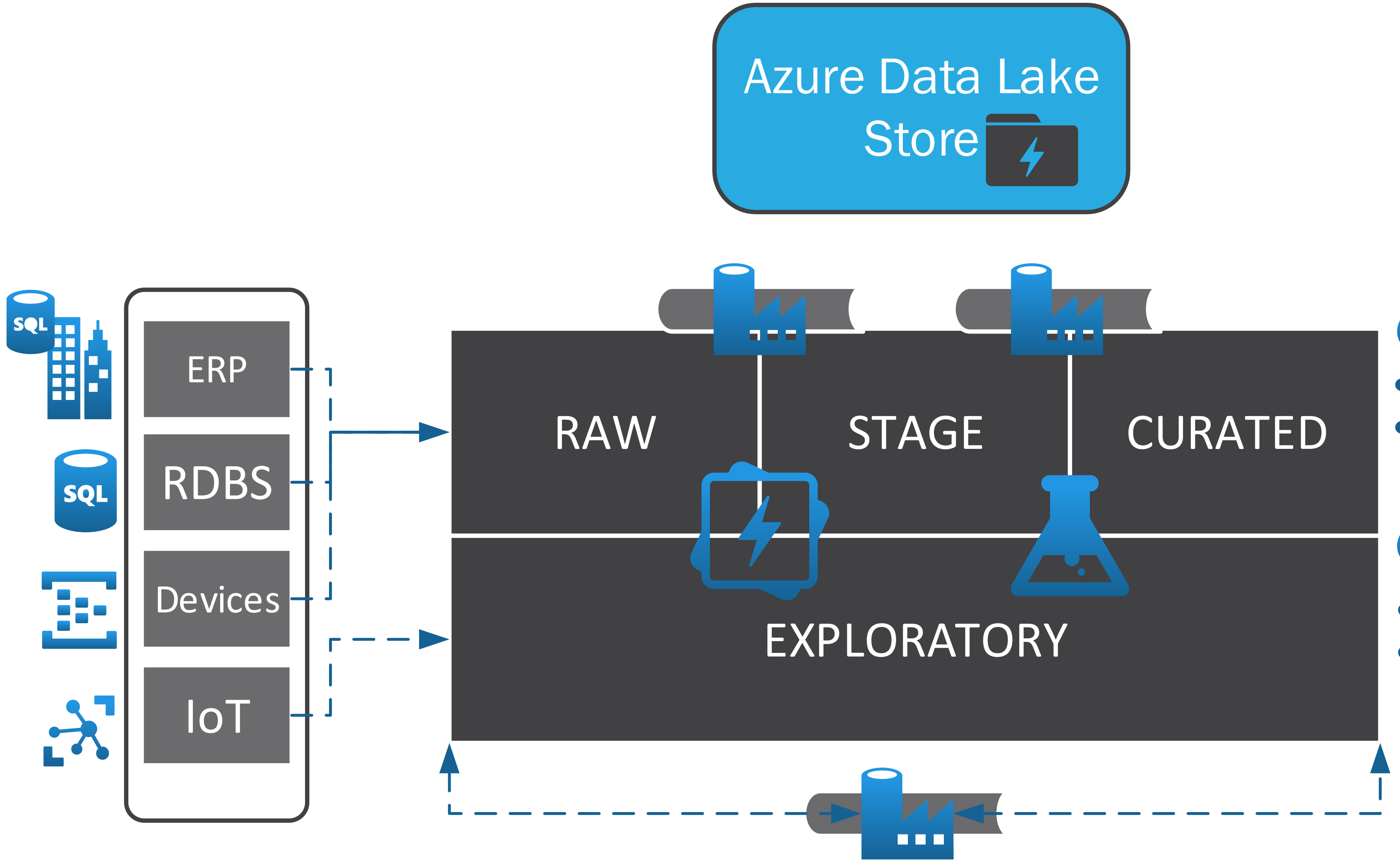
2

- **Operational**
- Value has been identified

1

- **Explorational**
- Value is being discovered

Data Lake Operations



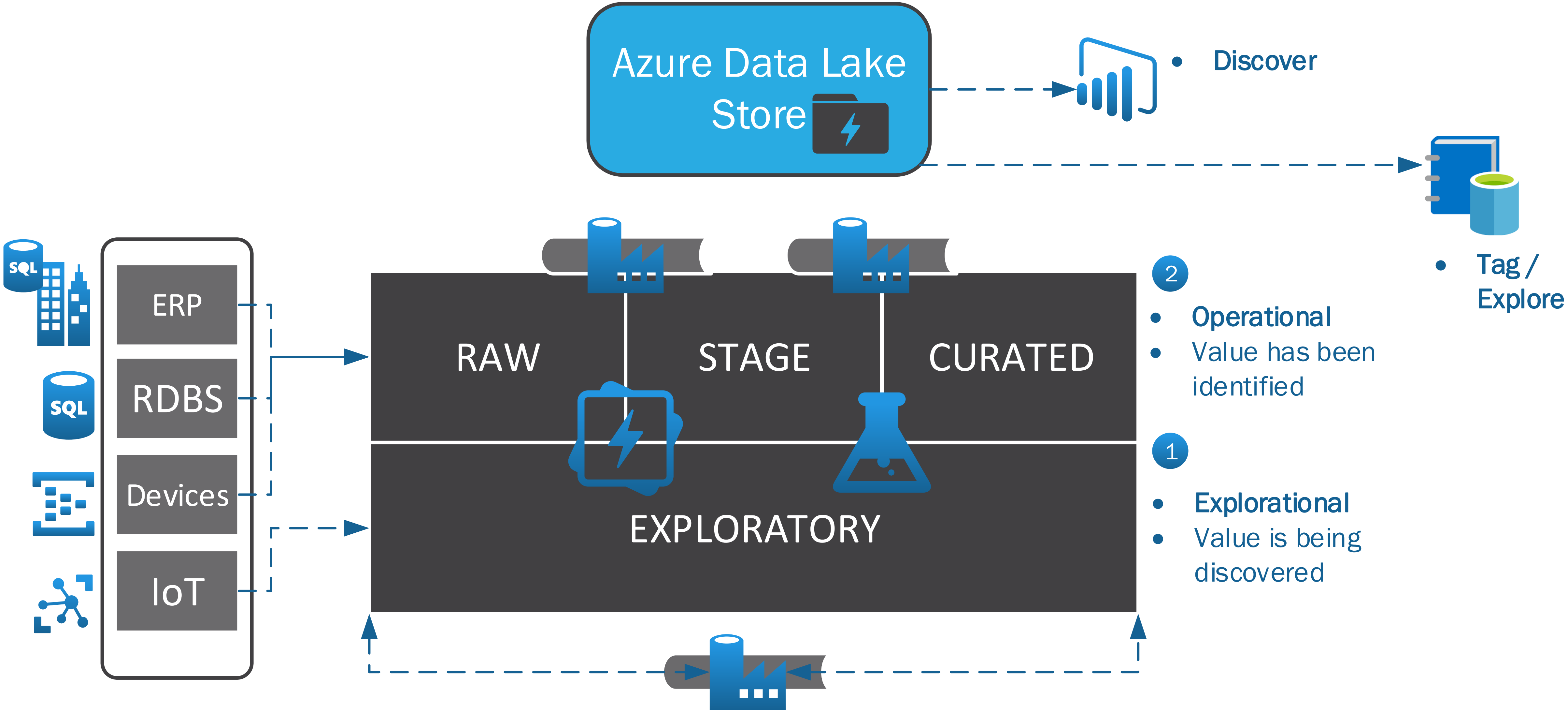
2

- **Operational**
- Value has been identified

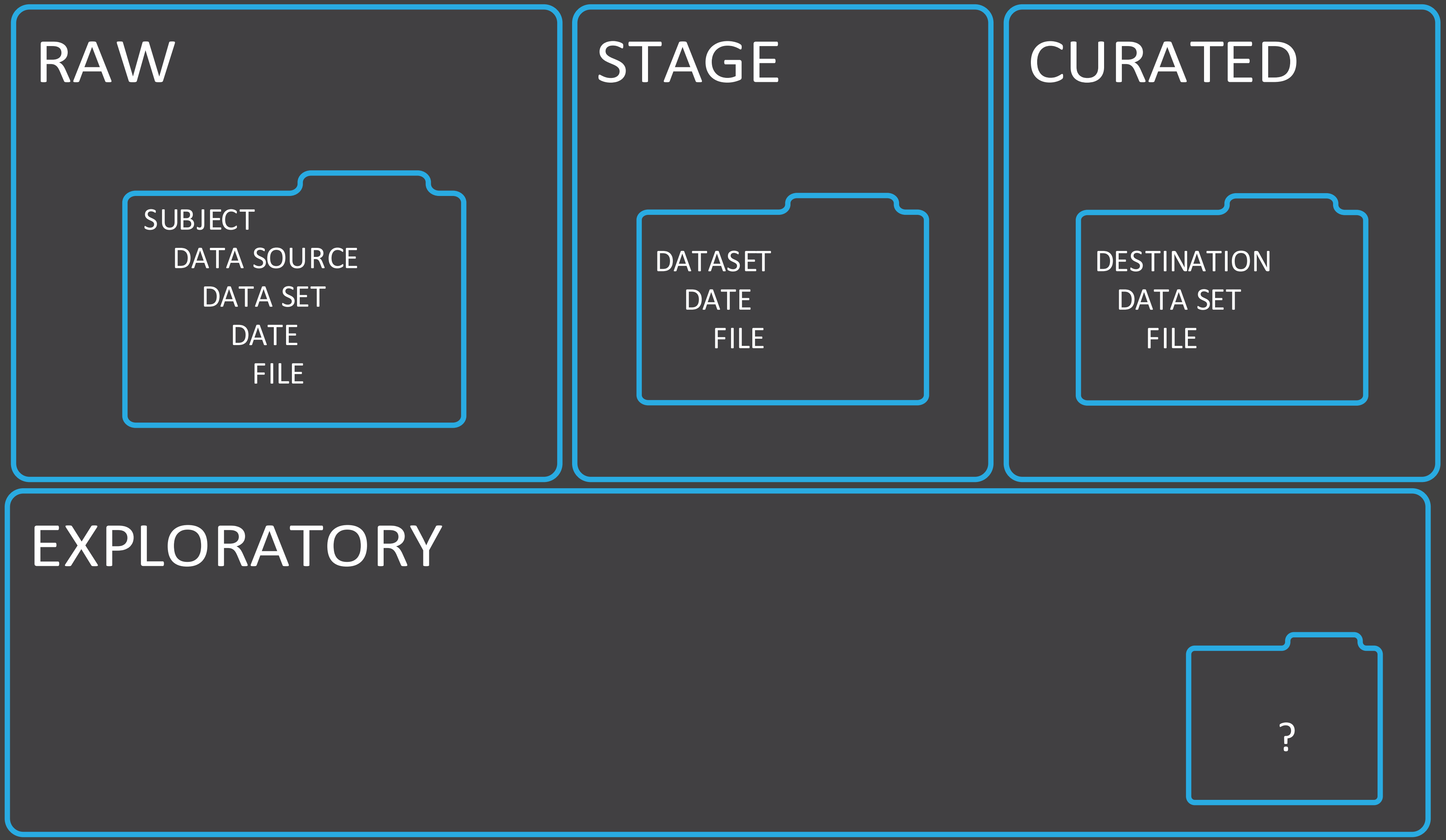
1

- **Explorational**
- Value is being discovered

Data Lake Operations



Data Lake Organization



RAW

SUBJECT
DATA SOURCE
DATA SET
DATE
FILE

STAGE

DATASET
DATE
FILE

CURATED

DESTINATION
DATA SET
FILE

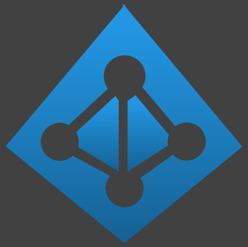
EXPLORATORY

?

Data Lake Security

RAW (1)	STAGE (2)	CURATED (3)	EXPLORATION (0)
Data Experts/Engineers	Data Experts/Engineers	ETL and BI Engineers / SME’s / Analysts	Data Scientist / Analysts

TOOLS



Data Lake Tagging

RAW (1)	STAGE (2)	CURATED (3)	EXPLORATION (0)
Data Experts/Engineers	Data Experts/Engineers	ETL and BI Engineers / SME's / Analysts	Data Scientist / Analysts
AUTOMATED		**SME**	**N/A**

TOOLS

TALAVANT

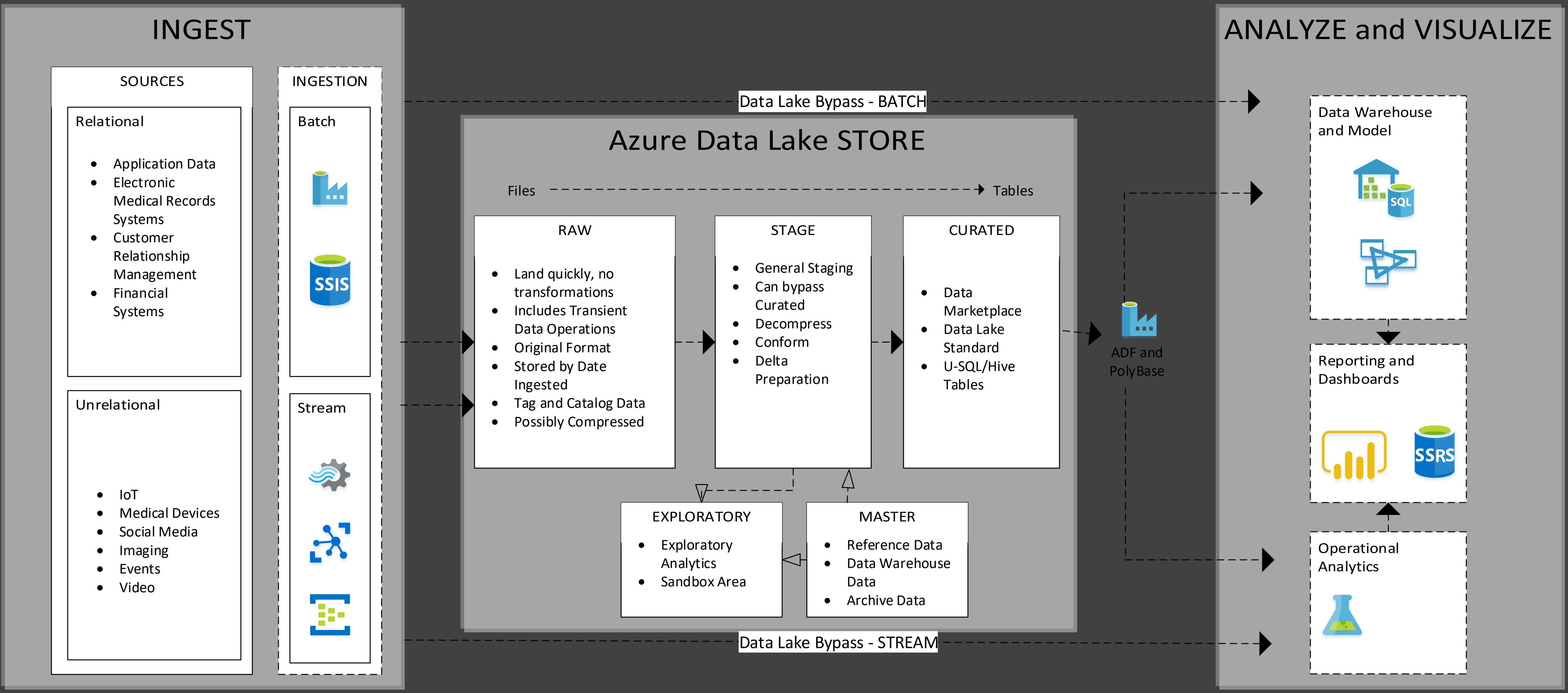
Data Lake Processing

RAW (1)	STAGE (2)	CURATED (3)	EXPLORATION (0)
Data Experts/Engineers	Data Experts/Engineers	ETL and BI Engineers / SME's / Analysts	Data Scientist / Analysts
AUTOMATED		SME	N/A
INGESTION	**CLEANSING**	**DISTRIBUTION**	

TOOLS

TALAVANT

Conceptual Architecture



INGEST

SOURCES

Relational

- Application Data
- Electronic Medical Records Systems
- Customer Relationship Management
- Financial Systems

Unrelational

- IoT
- Medical Devices
- Social Media
- Imaging
- Events
- Video

INGESTION

Batch

Stream

Data Lake Bypass - BATCH

Azure Data Lake STORE

Files

Tables

RAW

- Land quickly, no transformations
- Includes Transient Data Operations
- Original Format
- Stored by Date Ingested
- Tag and Catalog Data
- Possibly Compressed

STAGE

- General Staging
- Can bypass Curated
- Decompress
- Conform
- Delta Preparation

CURATED

- Data Marketplace
- Data Lake Standard
- U-SQL/Hive Tables

EXPLORATORY

- Exploratory Analytics
- Sandbox Area

MASTER

- Reference Data
- Data Warehouse Data
- Archive Data

Data Lake Bypass - STREAM

ADF and PolyBase

ANALYZE and VISUALIZE

Data Warehouse and Model

Reporting and Dashboards

Operational Analytics

TALAVANT

Data Lake Questions

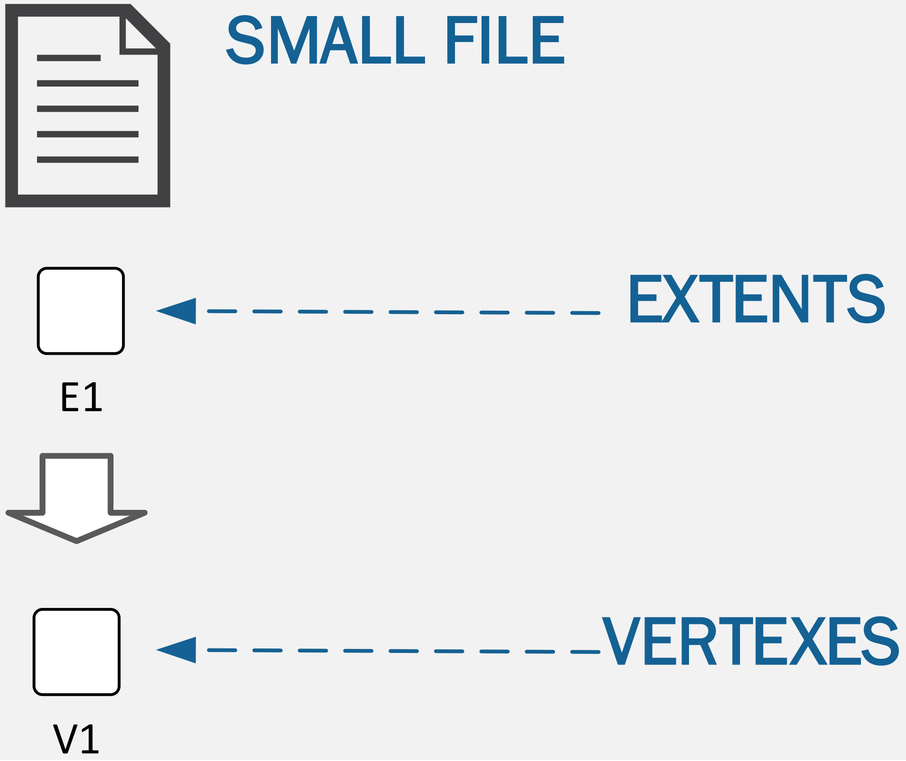
?

1. Big Data Overview
2. Data Lake Concepts
3. **Azure Data Lake Store**
4. Azure Data Lake Analytics
5. U-SQL

Azure Data Lake Store

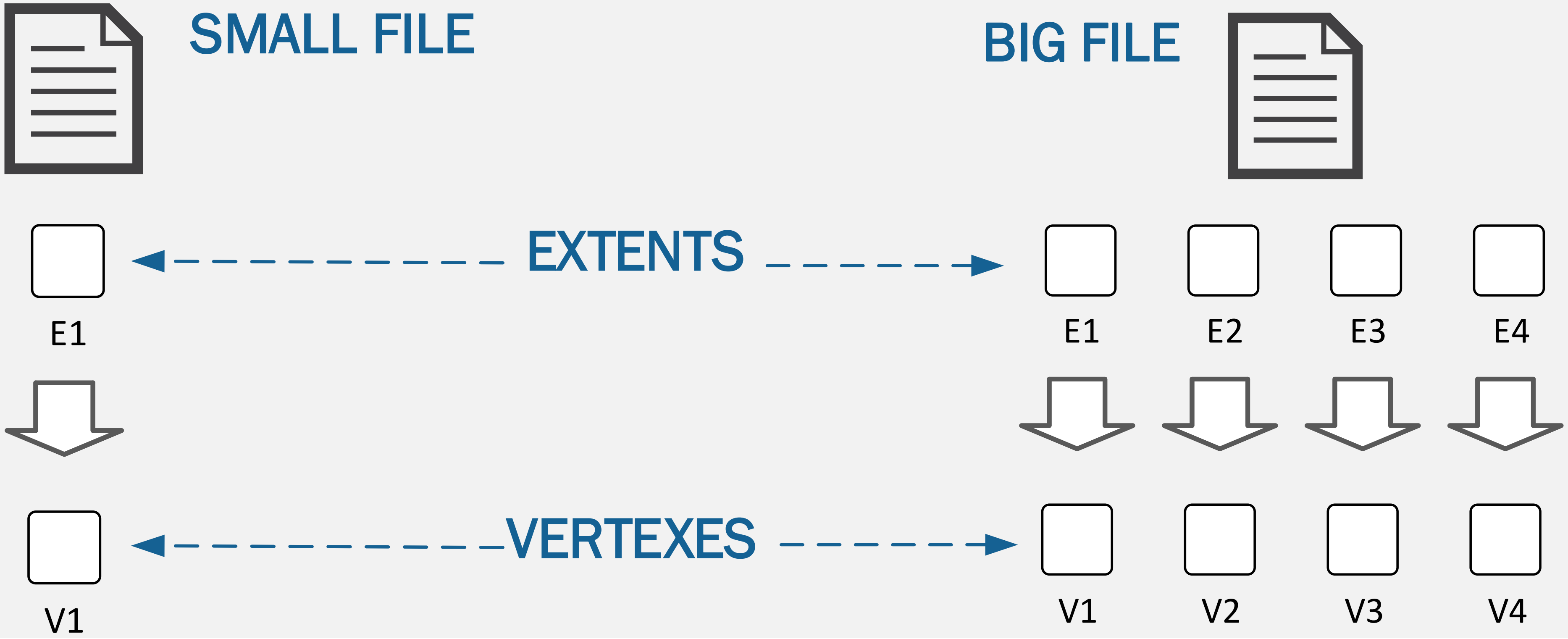
- **HDFS** Data Store for housing data in it’s Native **Raw** Format
  - Built on Apache YARN
- Process and store **Petabyte** size files
- Enterprise **Security** through Azure Active Directory
- No storage limit

Data Lake Store

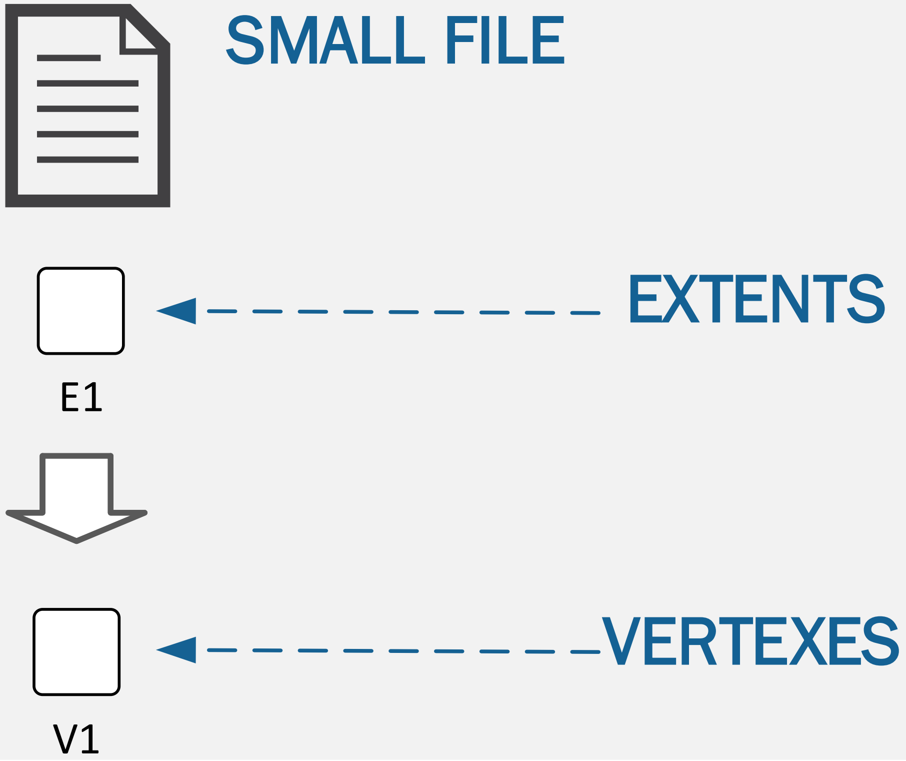


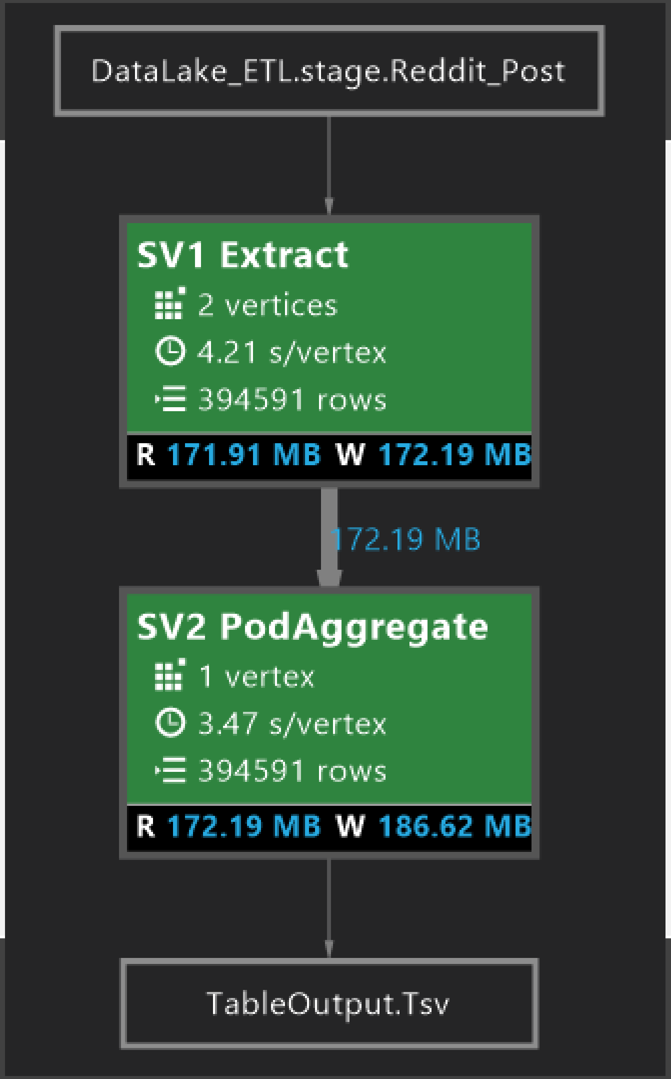
TALAVANT

Data Lake Store



Data Lake Store





DataLake_ETL.stage.Reddit_Post

SV1 Extract

- 2 vertices
- 4.21 s/vertex
- 394591 rows

R 171.91 MB W 172.19 MB

172.19 MB

SV2 PodAggregate

- 1 vertex
- 3.47 s/vertex
- 394591 rows

R 172.19 MB W 186.62 MB

TableOutput.Tsv

Data Lake Ingestion

- Visual Studio
- Azure Portal (Limit)
- SSIS Data Lake Destination and SSIS Data Lake Task
- Azure Data Factory
- Powershell
- ADLCopy
- Sqoop (HDInsight)
- Other Apache tools

1. Big Data Overview
2. Data Lake Concepts
3. Azure Data Lake Store
4. **Azure Data Lake Analytics**
5. U-SQL

Azure Data Lake Analytics

- **Big Data Queries** as a **Service**
- Analytics **Federation**
- Develop in **U-SQL**, **.NET**, **R**, and **Python**
- **Cognitive Services**
- **Scale** Instantly
- Pay **Per Job**

1. Big Data Overview
2. Data Lake Concepts
3. Azure Data Lake Store
4. Azure Data Lake Analytics
5. **U-SQL**

Intro to U-SQL

KEY FEATURES

- Combines **SQL** and **C#**
- **Patterned File Processing**
- Extensions: **Python**, **R**, **Cognitive**
- Query Data where it Lives (**Federated Querying**)
- **Partition** and **Distribution** of Data for **Massive Parallelism**
- Manage Structure and Shared Programming through **U-SQL Catalog**
- U-SQL Procedures

U-SQL : Extract Query

U-SQL

1

```
@MyExtract =
EXTRACT
  Field1 string,
  Field2 int,
  Field 3 int?
FROM “/datalake/01_RAW/{*}.csv”
USING Extractors.Csv();
```

T-SQL

```
  CREATE TABLE myTable
  (
  Field1 VARCHAR(100),
  Field2 INT,
  Field3 INT NOT NULL
  );

INSERT INTO myTable
( Field1, Field2, Field3)
SELECT
  CAST(Field1 as varchar(100) as Field1,
  CAST(Field2 AS INT) as Field2,
  CONVERT(INT, Field3) as Field 3
FROM myTable
```

U-SQL : Extract Query

U-SQL

1
```
@MyExtract =
EXTRACT
  Field1 string,
  Field2 int,
  Field 3 int?
FROM "/datalake/01_RAW/{*}.csv"
USING Extractors.Csv();
```

2
```
@MyAgg =
SELECT
  Field1,
  MAX(Field2) A
FROM @MyExtract
GROUP BY Field1;
```

T-SQL

```
  CREATE TABLE myTable
  (
  Field1 VARCHAR(100),
  Field2 INT,
  Field3 INT NOT NULL
  );

INSERT INTO myTable
( Field1, Field2, Field3)
SELECT
  CAST(Field1 as varchar(100) as Field1,
  CAST(Field2 AS INT) as Field2,
  CONVERT(INT, Field3) as Field 3
FROM myTable
```

U-SQL : Extract Query

U-SQL

1

```
@MyExtract =
EXTRACT
  Field1 string,
  Field2 int,
  Field 3 int?
FROM “/datalake/01_RAW/{*}.csv”
USING Extractors.Csv();
```

2

```
@MyAgg =
SELECT
  Field1,
  MAX(Field2) AS Field2
FROM @MyExtract
GROUP BY Field1;
```

3

```
OUTPUT @MyAgg
TO     datalake/02_STAGE/MyOutput.csv”
USING Outputters.Csv();
```

T-SQL

```
 CREATE TABLE myTable
 (
 Field1 VARCHAR(100),
 Field2 INT,
 Field3 INT NOT NULL
 );

INSERT INTO myTable
( Field1, Field2, Field3)
SELECT
  CAST(Field1 as varchar(100) as Field1,
  CAST(Field2 AS INT) as Field2,
  CONVERT(INT, Field3) as Field 3
FROM myTable
```

U-SQL : Extractors and Outputters

CURRENT EXTRACTORS

- Csv()
- Tsv()
- Txt()

CURRENT OUTPUTTERS

- Csv()
- Tsv()
- Txt()

U-SQL : Extractor and Outputter Parameters

```
EXTRACT
…
FROM “/datalake/01_RAW/{*}.CSV
USING Extractors.Csv(silent : true , delimiter : “,”)
```

(); PARAMETERS

- Delimiter
- Encoding
- escapeCharecter
- nullEscape
- Quoting
- rowDelimiter
- Silent
- skipFirstNRows
- charFormat

U-SQL : Extractors and Outputters

CURRENT EXTRACTORS

- Csv()
- Tsv()
- Txt()

CURRENT OUTPUTTERS

- Csv()
- Tsv()
- Txt()

CUSTOM EXTRACTORS and OUTPUTTERS

- FlexExtractor()
- XML()
- JSON()
- Avro()

U-SQL : Virtual Columns

```
DECLARE        =“/datalake/01_stage/2017/06/{FileName}.csv

@MyExtract =
EXTRACT
  Field1 string
  Field2 int,
  Field 3 int?,
  FileName
FROM @IN
USING Extractors.Csv()
WHERE FileName == “MyRedditFile_20170602”;
```

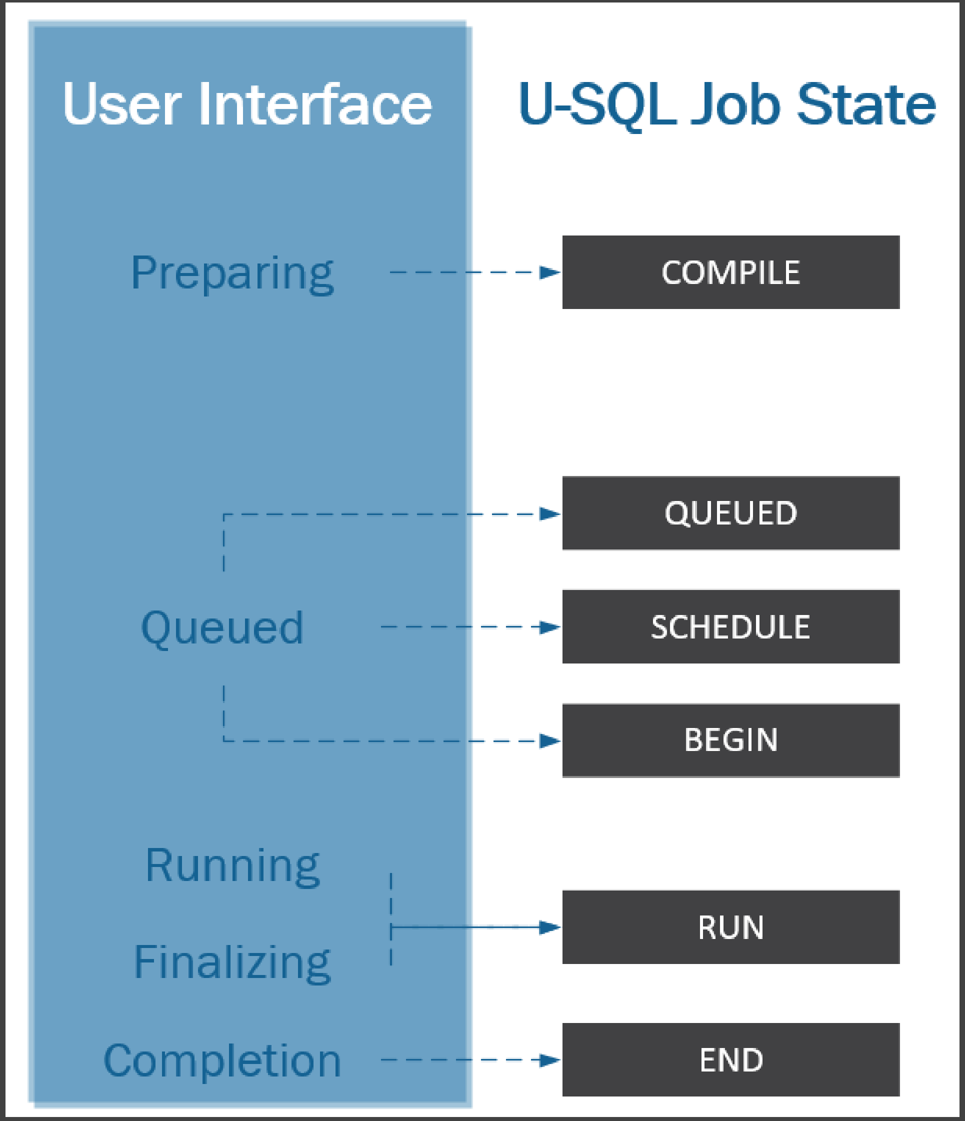
File Names :

MyRedditFile_20170601.csv

MyRedditFile_20170602.csv

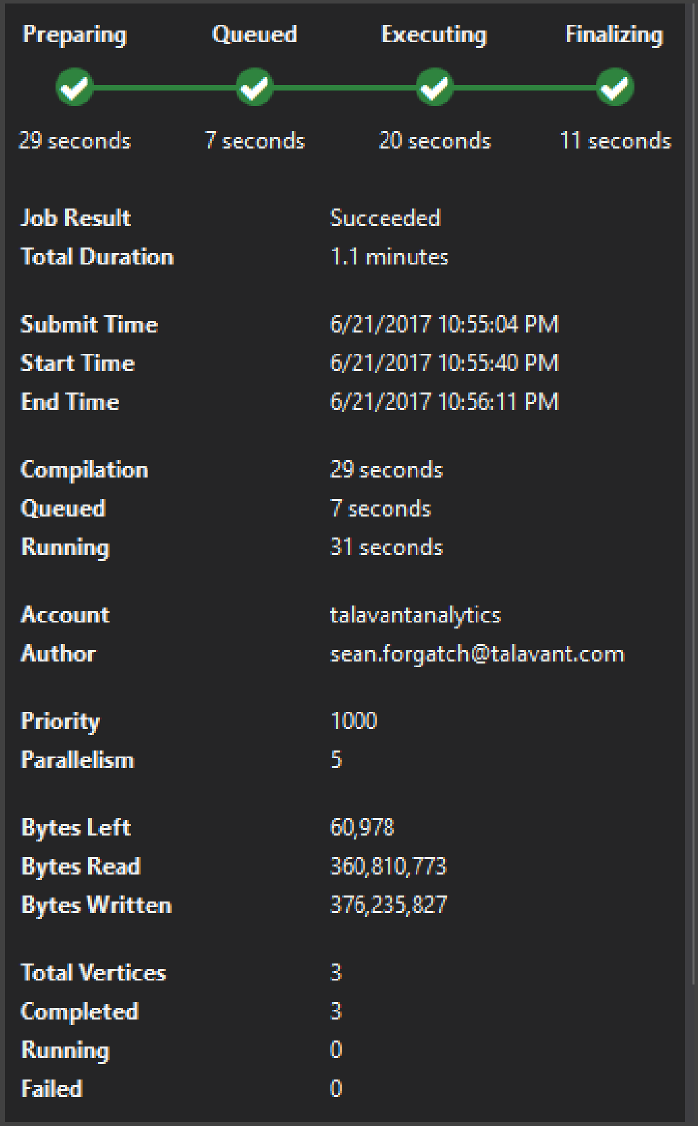
etc…

U-SQL: Job Execution



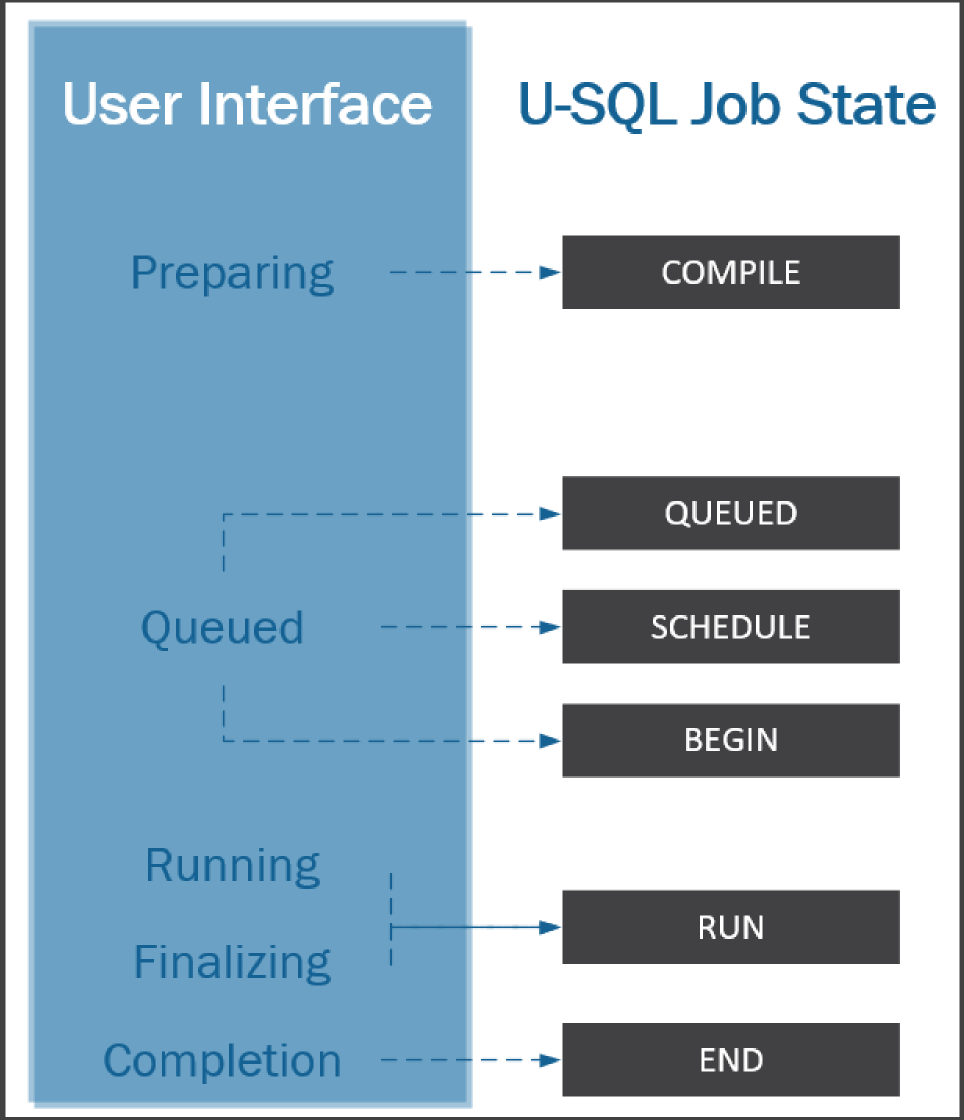
TALAVANT

U-SQL: Job Execution



Preparing	Queued	Executing	Finalizing
✓	✓	✓	✓
29 seconds	7 seconds	20 seconds	11 seconds

Job Result	Succeeded
Total Duration	1.1 minutes
Submit Time	6/21/2017 10:55:04 PM
Start Time	6/21/2017 10:55:40 PM
End Time	6/21/2017 10:56:11 PM
Compilation	29 seconds
Queued	7 seconds
Running	31 seconds
Account	talavantanalytics
Author	sean.forgatch@talavant.com
Priority	1000
Parallelism	5
Bytes Left	60,978
Bytes Read	360,810,773
Bytes Written	376,235,827
Total Vertices	3
Completed	3
Running	0
Failed	0



TALAVANT

U-SQL : Tables

GUIDELINES

1. Must Have **Clustered Index**
2. Utilize When **Improving Performance** with Distribution/Partitioning
3. You have **Multiple Large Files**
4. Don’t Use when:
    - No Filtering, Joining, Grouping

U-SQL : Tables

```
DROP TABLE IF EXISTS <adla>.<database>.<schema>.tableName;
CREATE TABLE <adla>.<database>.<schema>.tableName
(
Field1 int,
Field2 string,
Field3 int?

INDEX idx_1 CLUSTERED(Field1)
DISTRIBUTED BY HASH(Field2)
);
```

U-SQL : Tables

```
DROP TABLE IF EXISTS
<adla>.<database>.<schema>.tableName;
CREATE TABLE <adla>.<database>.<schema>.tableName
(
Field1 int,
Field2 string,
Field3 int?

INDEX idx_1 CLUSTERED(Field1)
DISTRIBUTED BY HASH(Field2)
)
AS SELECT ...
```

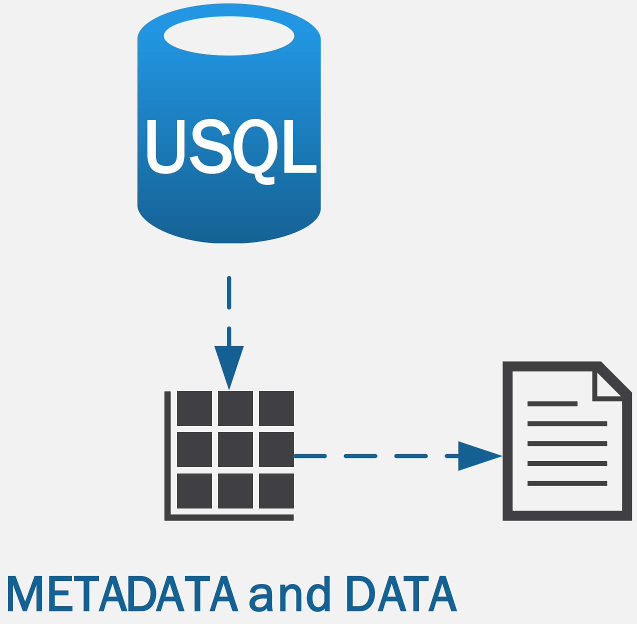
U-SQL : Tables

```
DROP TABLE IF EXISTS
<adla>.<database>.<schema>.tableName;
CREATE TABLE <adla>.<database>.<schema>.tableName
(
Field1 int,
Field2 string,
Field3 int?

INDEX idx_1 CLUSTERED(Field1)
DISTRIBUTED BY HASH(Field2)
)
AS EXTRACT ...
```

U-SQL : Tables

```
DROP TABLE IF EXISTS
<adla>.<database>.<schema>.tableName;
CREATE TABLE <adla>.<database>.<schema>.tableName
(
Field1 int,
Field2 string,
Field3 int?

INDEX idx_1 CLUSTERED(Field1)
DISTRIBUTED BY HASH(Field2)
)
AS TVF ...
```

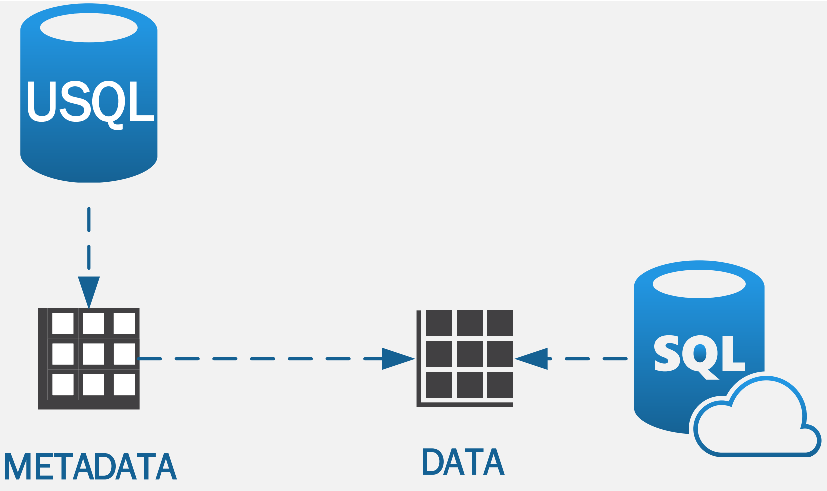
U-SQL : Tables



MANAGED

- Own Their Data
- No Heaps
- INSERT only

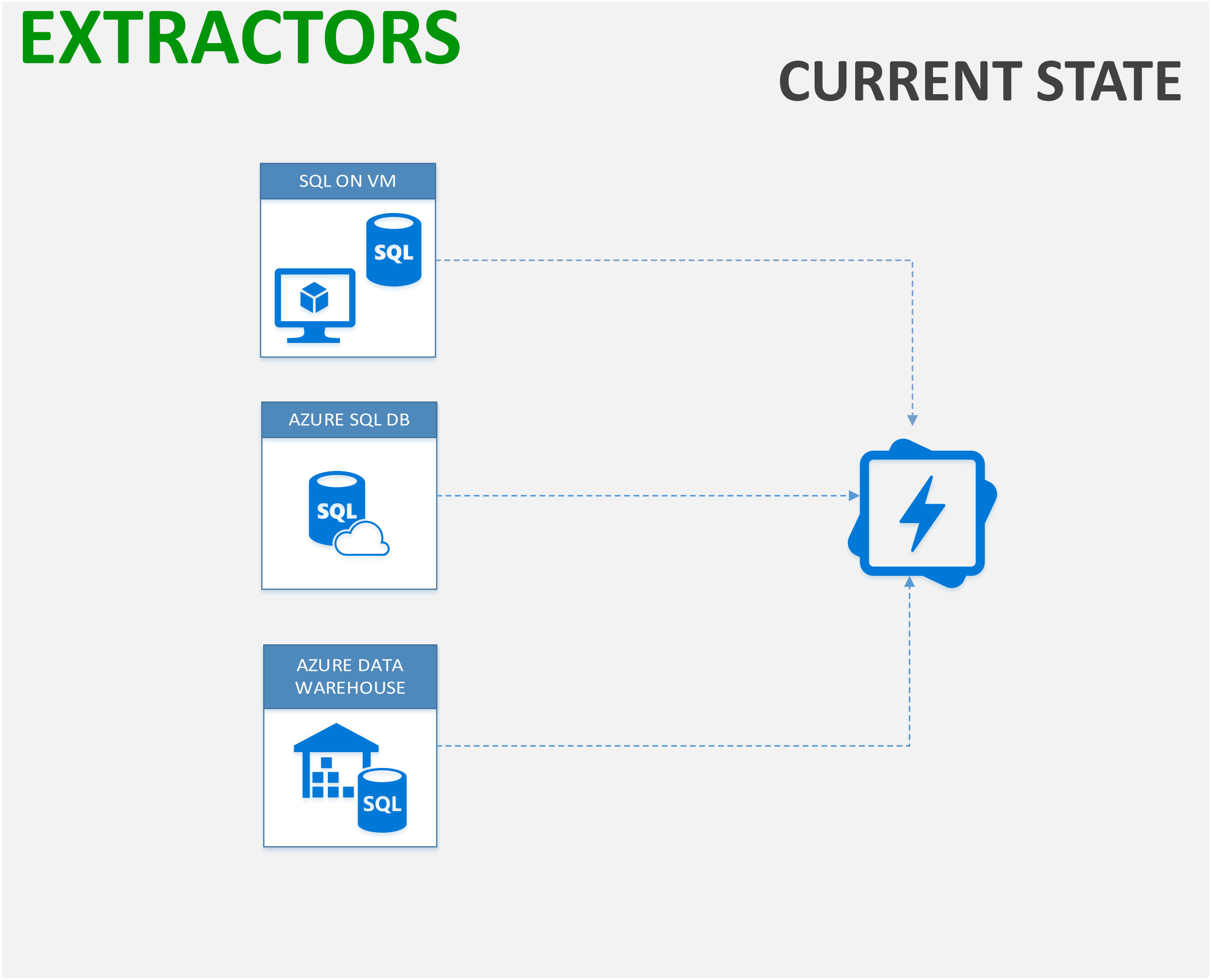
U-SQL : Tables



EXTERNAL

- Stored Metadata
- Use VIEW or TVF over EXTRACT
- Data Lives on Source
  - Azure SQL DB
  - Azure SQL DW
  - Azure SQL VM

U-SQL : Data Federation



TALAVANT

U-SQL : Operators

COMPARISON OPERATORS	LOGICAL OPERATORS
IS NULL	AND
==	BETWEEN
>	IN, NOT IN
>=	LIKE, NOT LIKE
!=	NOT
	OR

U-SQL : Functions

REPORTING FUNCTIONS

- COUNT
- SUM
- MIN
- MAX
- AVG
- STDEV
- VAR

ANALYTIC FUNCTIONS

- CUME_DIST
- PERCENT_RANK
- PERCENTILE_CONT
- PERCENTILE_DISC

RANKING FUNCTIONS

- RANK
- DENSE_RANK
- NTILE
- ROW_NUMBER

U-SQL : C# Functions

MATH METHODS

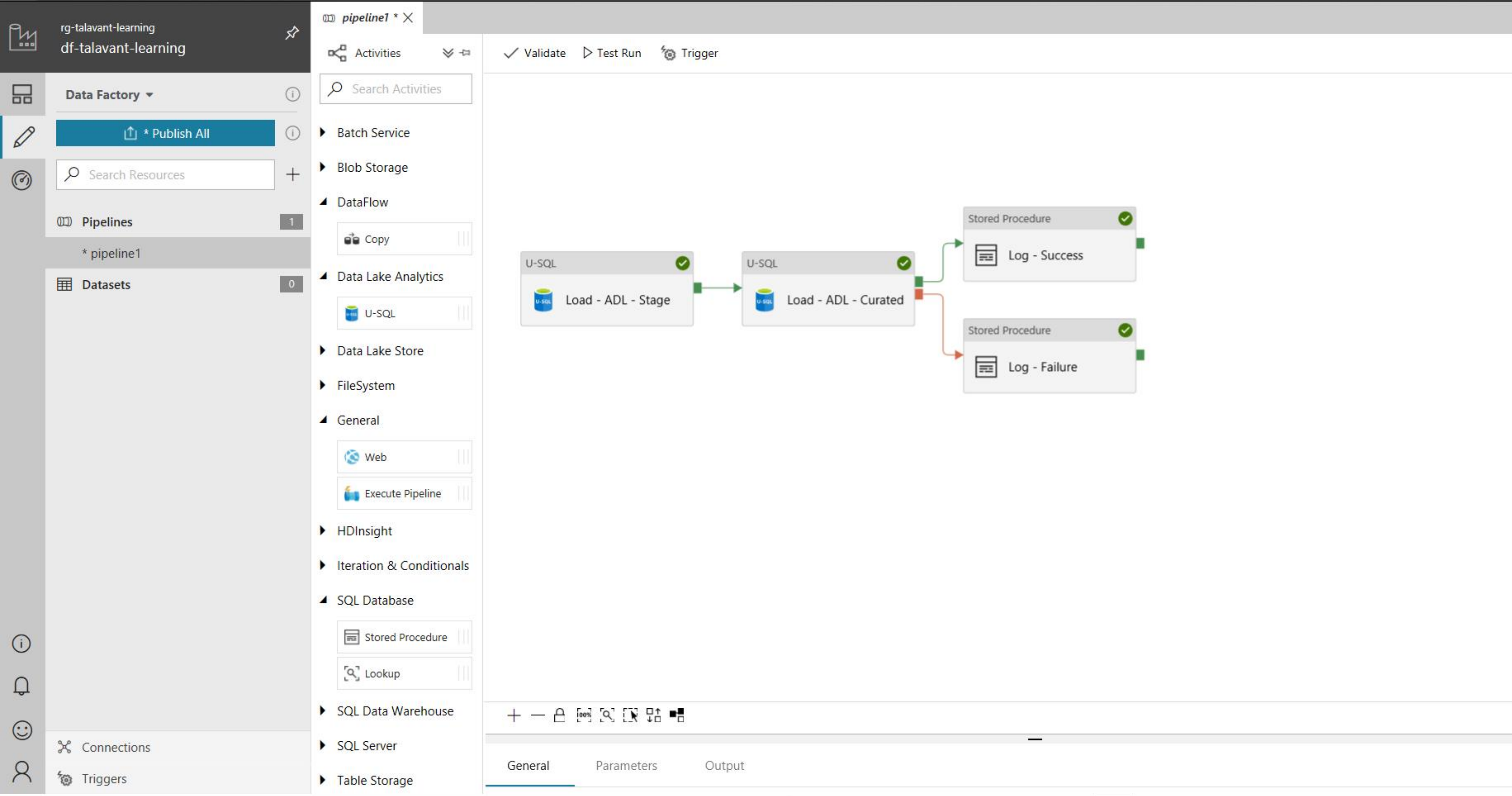
- Abs
- BigMul
- Floor
- Max/Min
- Round
- Sqrt
- ..plus many more!

STRING METHODS

- Compare
- Concat
- Contains
- Equals
- Replace
- Split
- ToUpper
- Trim
- ..plus many more!

Advice

- Identify Value of Data Lake Approach
- Data Lake: Invest Time and Strategy into Data Lake Design
- U-SQL: Utilize U-SQL Constructs before C#
- U-SQL: Understand and Control Data through Partitioning


rg-talavant-learning
df-talavant-learning
pipeline1 *
Activities
Validate
Test Run
Trigger
Data Factory
* Publish All
Search Resources
Pipelines 1
* pipeline1
Datasets 0
Connections
Triggers
Search Activities
Batch Service
Blob Storage
DataFlow
Copy
Data Lake Analytics
U-SQL
Data Lake Store
FileSystem
General
Web
Execute Pipeline
HDInsight
Iteration & Conditionals
SQL Database
Stored Procedure
Lookup
SQL Data Warehouse
SQL Server
Table Storage
U-SQL
Load - ADL - Stage
U-SQL
Load - ADL - Curated
Stored Procedure
Log - Success
Stored Procedure
Log - Failure
General
Parameters
Output

Learn U-SQL !

- **Michael Rys** – LinkedIn Slide Share's
- **GitHub** – U-SQL Repository
- SQL Server Central – **Stairway to U-SQL**
- **Azure** – Built in Example

Let’s Connect!

https://www.linkedin.com/in/seanforgatch/

Sean.Forgatch@Talavant.com

- @4gatchSQL

TALAVANT

PASS