

Designing a Modern Data Warehouse + Data Lake

*Strategies & architecture options for implementing a
modern data warehousing environment*

Melissa Coates

Solution Architect,
BlueGranite



Blog: sqlchick.com

Twitter: [@sqlchick](https://twitter.com/sqlchick)



Designing a Modern Data Warehouse + Data Lake

Agenda

Discuss strategies & architecture options for:

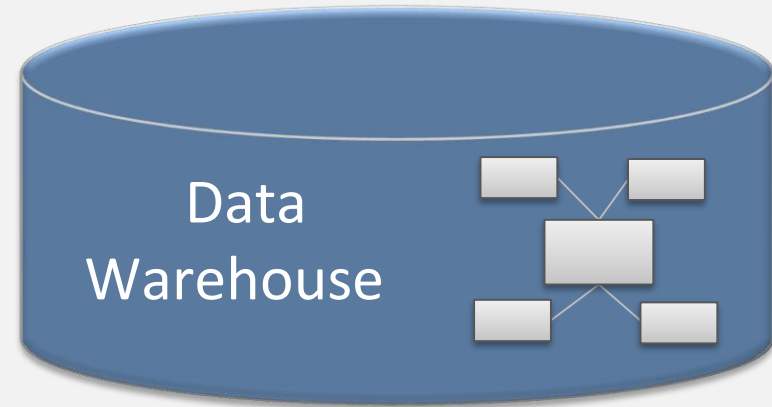
- 1) Evolving to a Modern Data Warehouse
- 2) Data Lake Objectives, Challenges, & Implementation Options
- 3) The Logical Data Warehouse & Data Virtualization

Evolving to a Modern Data Warehouse

Data Warehousing

Data is inherently more **valuable** once it is integrated from multiple systems. Full view of a customer:

- Sales activity +
- Delinquent invoices +
- Support/help requests



A DW is designed to be user-friendly, utilizing business terminology.

A DW is frequently built with a denormalized (star schema) data model. Data modeling + ETL processes consume most of the time & effort.

Transaction System vs. Data Warehouse

OLTP

Focus:

- ✓ Operational transactions
- ✓ "Writes"

Scope:

One database system

Ex. Objectives:

- ✓ Process a customer order
- ✓ Generate an invoice

Data Warehouse

Focus:

- ✓ Informational and analytical
- ✓ "Reads"

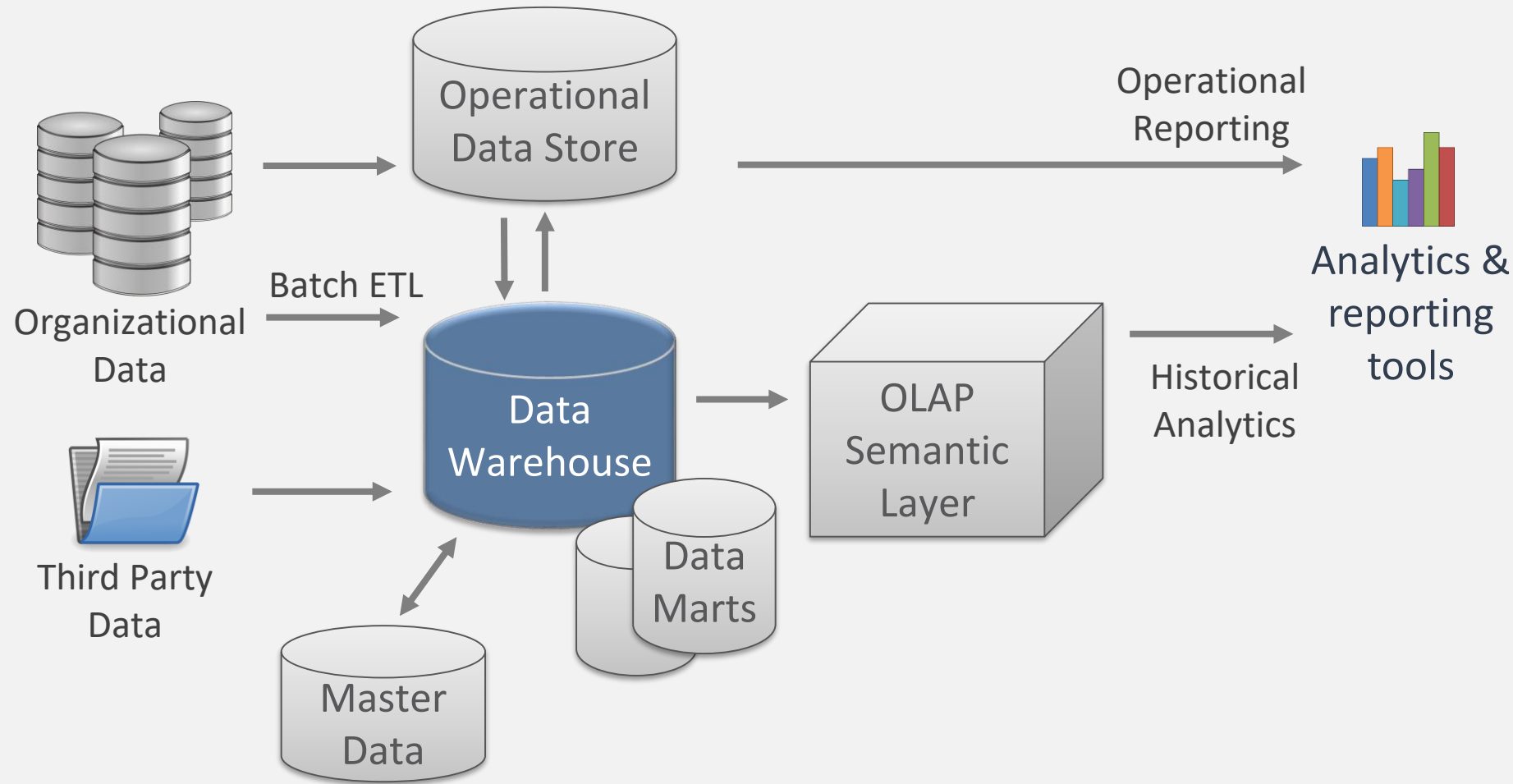
Scope:

Integrate data from multiple systems

Ex. Objectives:

- ✓ Identify lowest-selling products
- ✓ Analyze margin per customer

Traditional Data Warehousing



Loan Repayment Scenario:

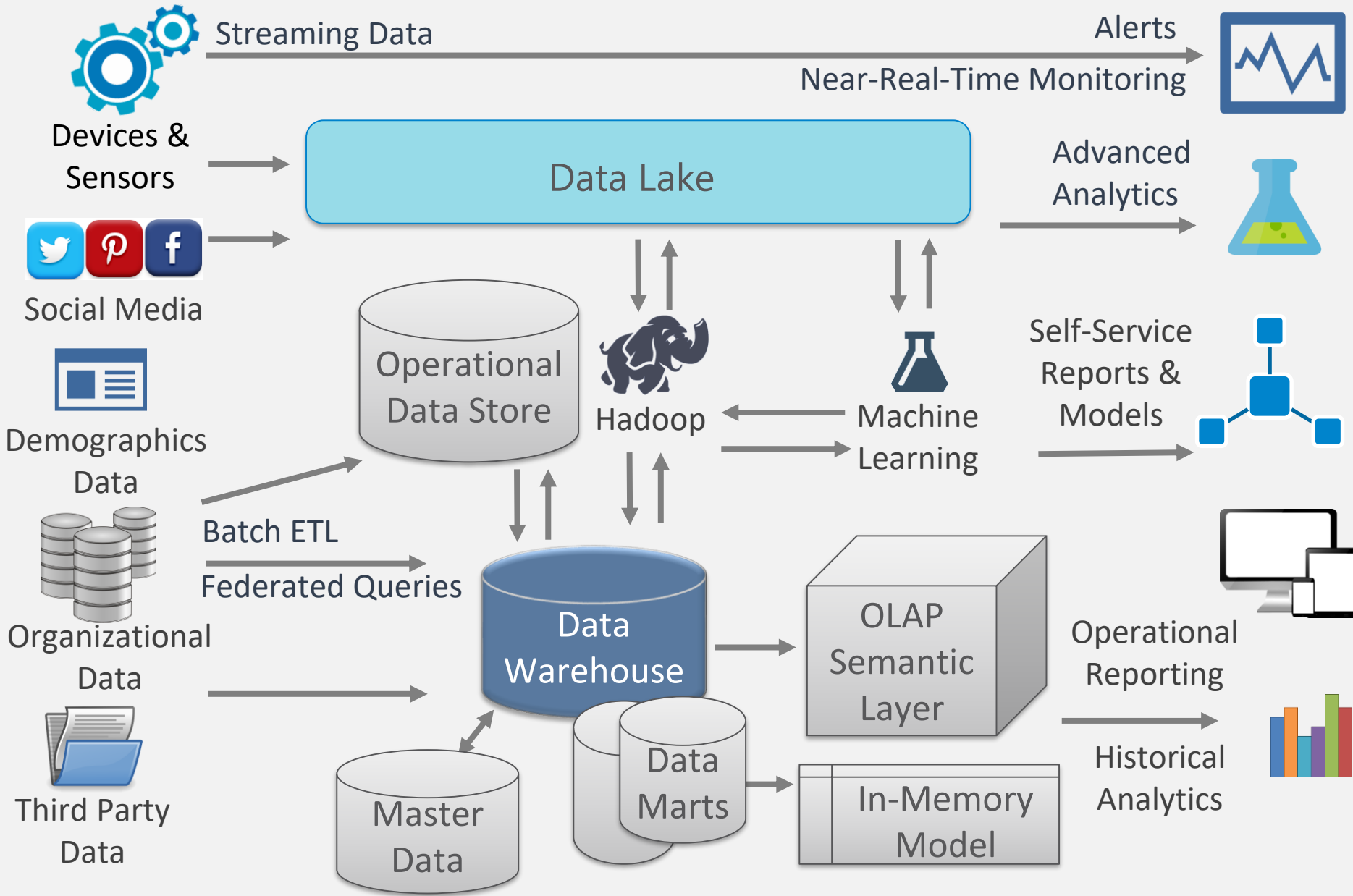
Organizational data:

- Customer application (income, assets)
- Loan history
- Payment activity

Third party data:

- Credit history

Modernizing an Existing DW



Loan Repayment Scenario:

Predictive Analytics:

- Model to predict repayment ability

Phone Records:

- Sentiment analysis

E-mail Records:

- Text analytics

Social Media:

- Personal comments

What Makes a Data Warehouse “Modern”

Variety of data sources;
multistructured

Coexists with
Data lake

Coexists with
Hadoop

Larger data
volumes; MPP

Multi-platform
architecture

Data
virtualization +
integration

Support all
user types &
levels

Flexible
deployment

Deployment
decoupled
from dev

Governance
model & MDM

Promotion of
self-service
solutions

Near real-time
data; Lambda
arch

Advanced
analytics

Agile
delivery

Cloud
integration;
hybrid env

Automation &
APIs

Data catalog;
search ability

Scalable
architecture

Analytics
sandbox w/
promotability

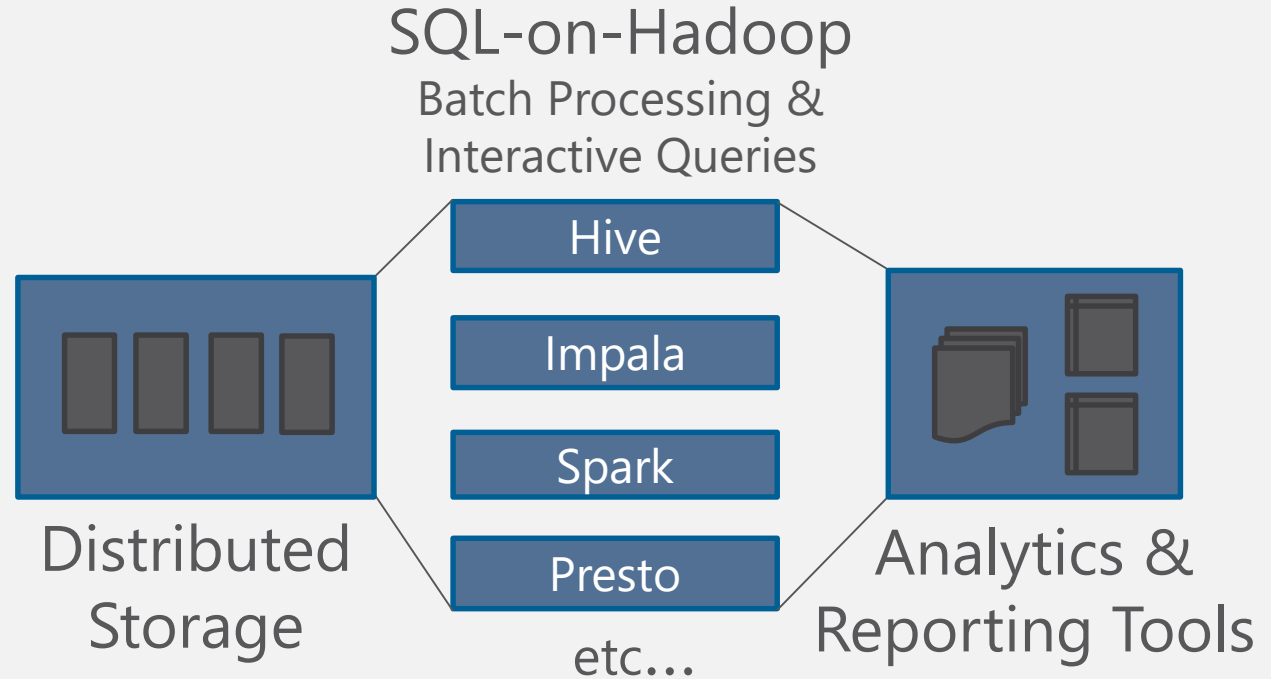
Bimodal
environment

Ways to Approach Data Warehousing

The DW **is** Hadoop:

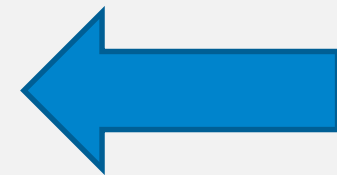
- ✓ Many open source projects
- ✓ Can utilize distributions (Hortonworks, Cloudera, MapR)
- ✓ Challenging implementation

OR



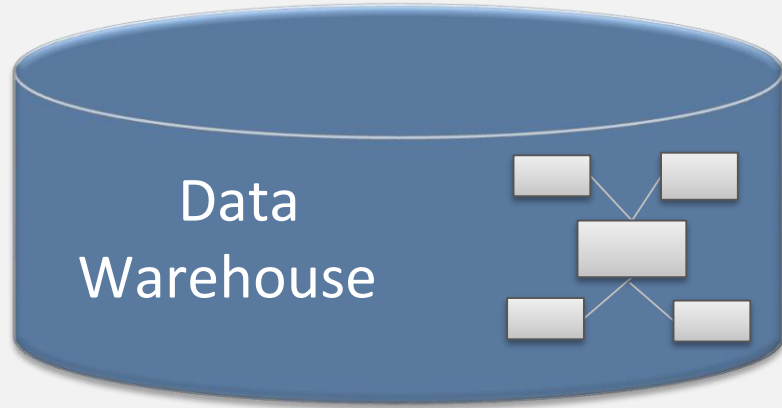
The DW & Hadoop co-exist & complement each other:

- ✓ Generally an easier path
- ✓ Augment an existing DW environment
- ✓ Additional value to existing DW investment



Focus of the
remainder of this
presentation

Growing an Existing DW Environment



Growing a DW:

- ✓ Data modeling strategies
- ✓ Partitioning
- ✓ Clustered columnstore index
- ✓ In-memory structures
- ✓ MPP (massively parallel processing)

Larger Scale Data Warehouse: MPP

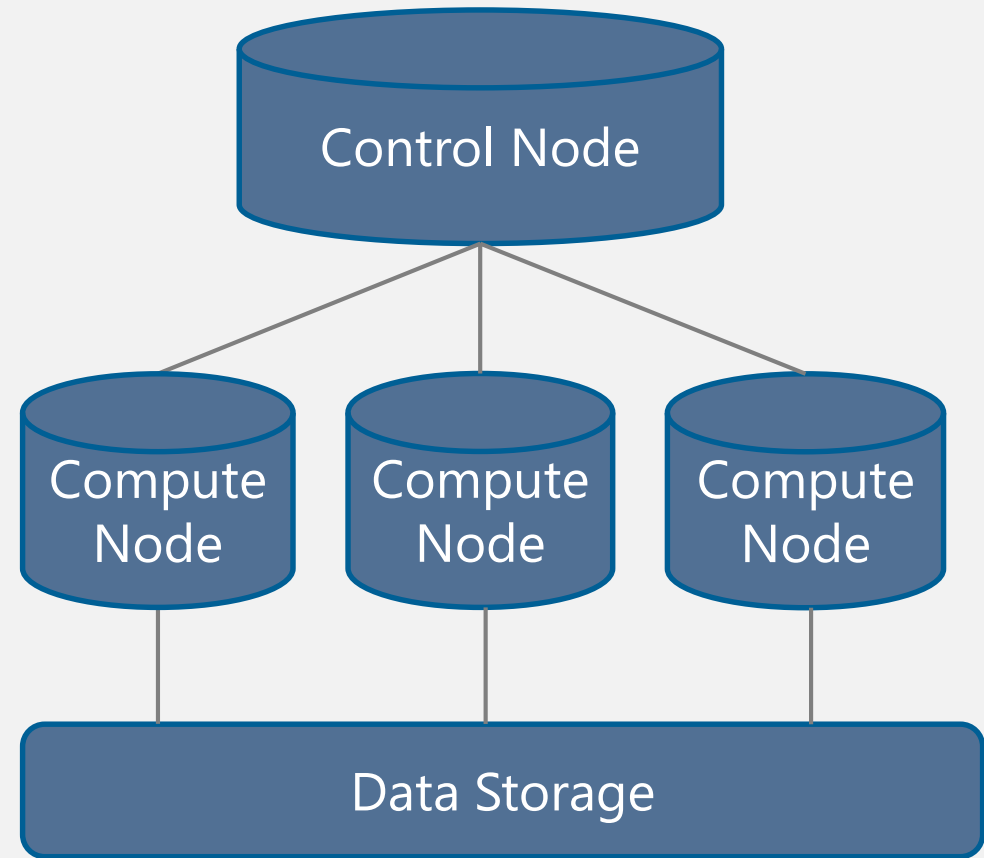
Massively Parallel Processing (MPP) operates on **high volumes of data** across **distributed nodes**

Shared-nothing architecture: each node has its own disk, memory, CPU

Decoupled storage and compute

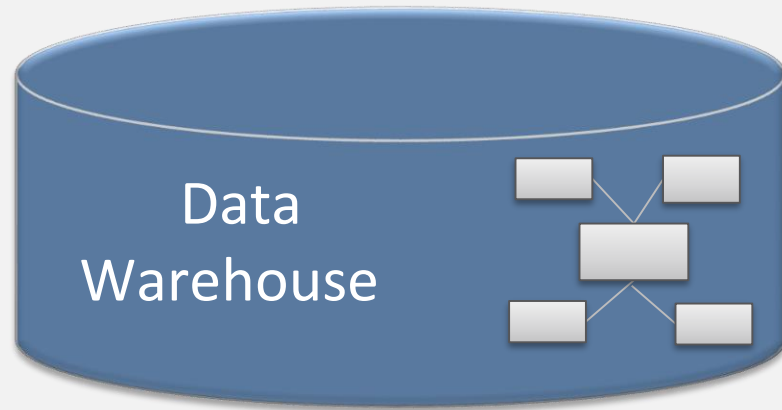
Scale up compute nodes to increase **parallelism**

Integrates with **relational & non-relational data**



Examples: Azure SQL DW, APS, Amazon Redshift, Snowflake

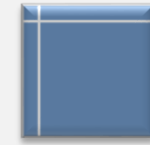
Growing an Existing DW Environment



Data Lake



Hadoop



In-Memory
Model



NoSQL

Growing a DW:

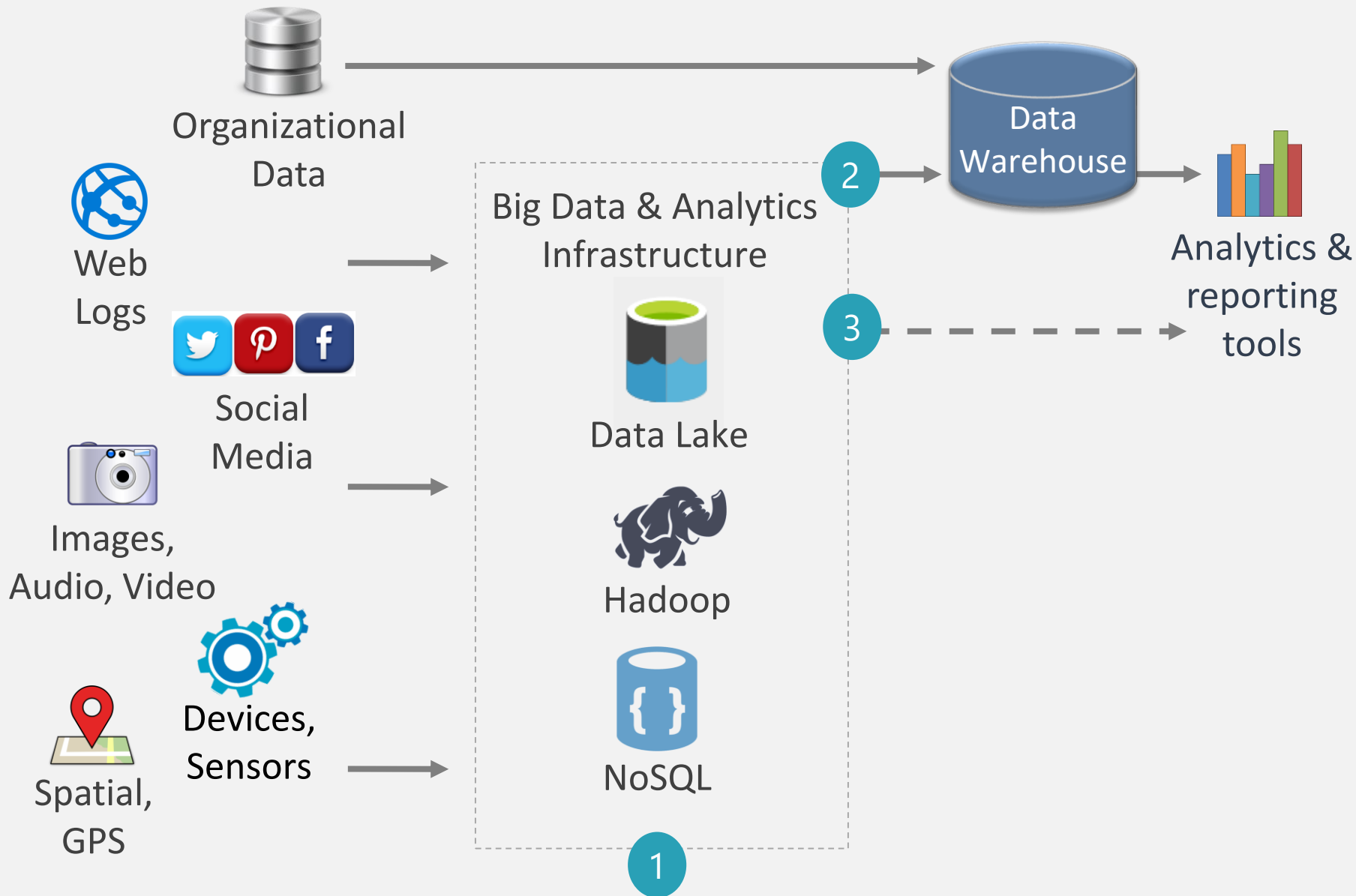
- ✓ Data modeling strategies
- ✓ Partitioning
- ✓ Clustered columnstore index
- ✓ In-memory structures
- ✓ MPP (massively parallel processing)

Extending a DW:

- ✓ Complementary data storage & analytical solutions
- ✓ Cloud & hybrid solutions
- ✓ Data virtualization (virtual DW)

-- Grow around your existing data warehouse --

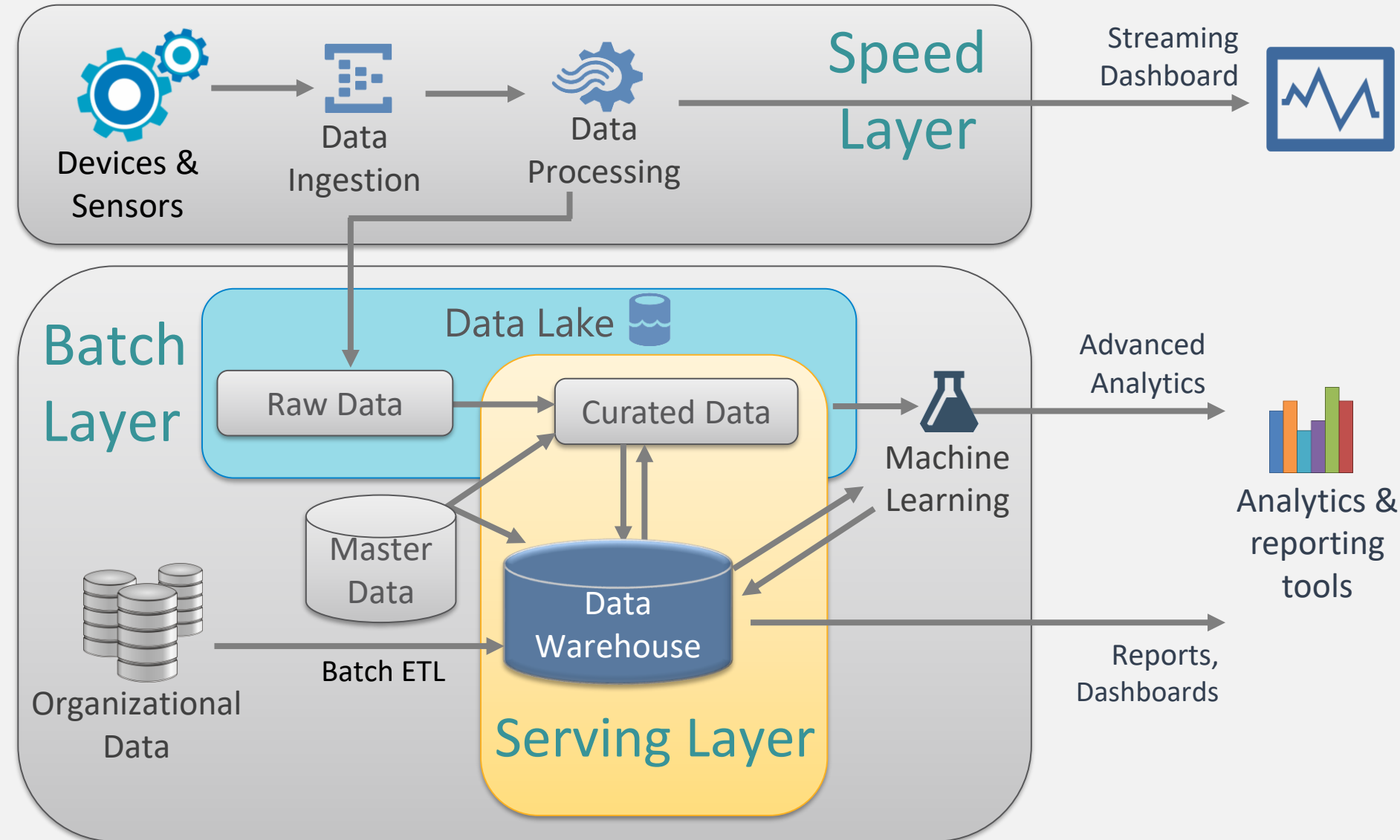
Multi-Structured Data



Objectives:

- 1** Storage for multi-structured data (json, xml, csv...) with a 'polyglot persistence' strategy
- 2** Integrate portions of the data into data warehouse
- 3** Federated query access (data virtualization)

Lambda Architecture



Speed Layer:
Low latency data

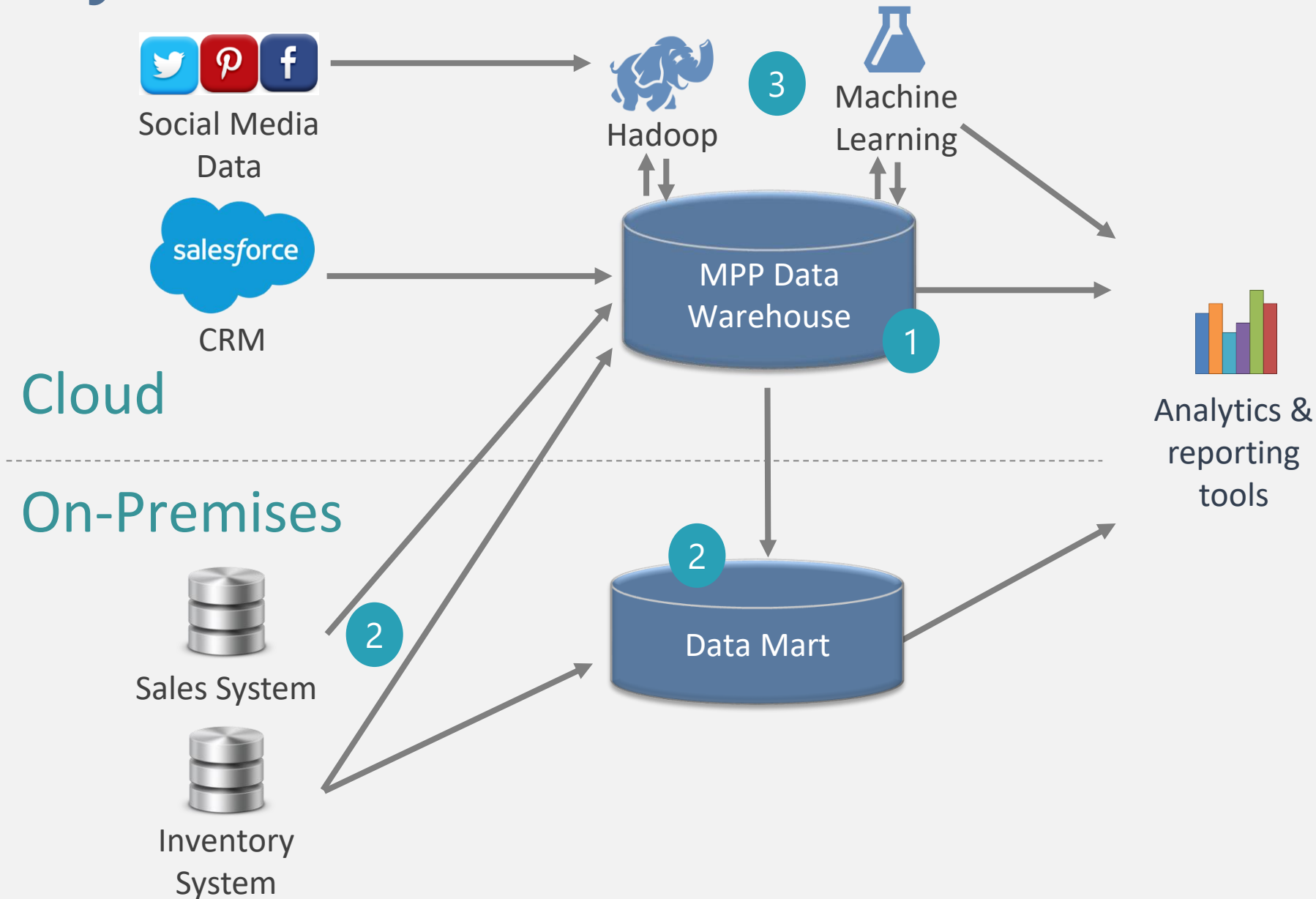
Batch Layer:
Data processing to support complex analysis

Serving Layer:
Responds to queries

Objectives:

- Support large volume of high-velocity data
- Near real-time analysis + persisted history

Hybrid Architecture



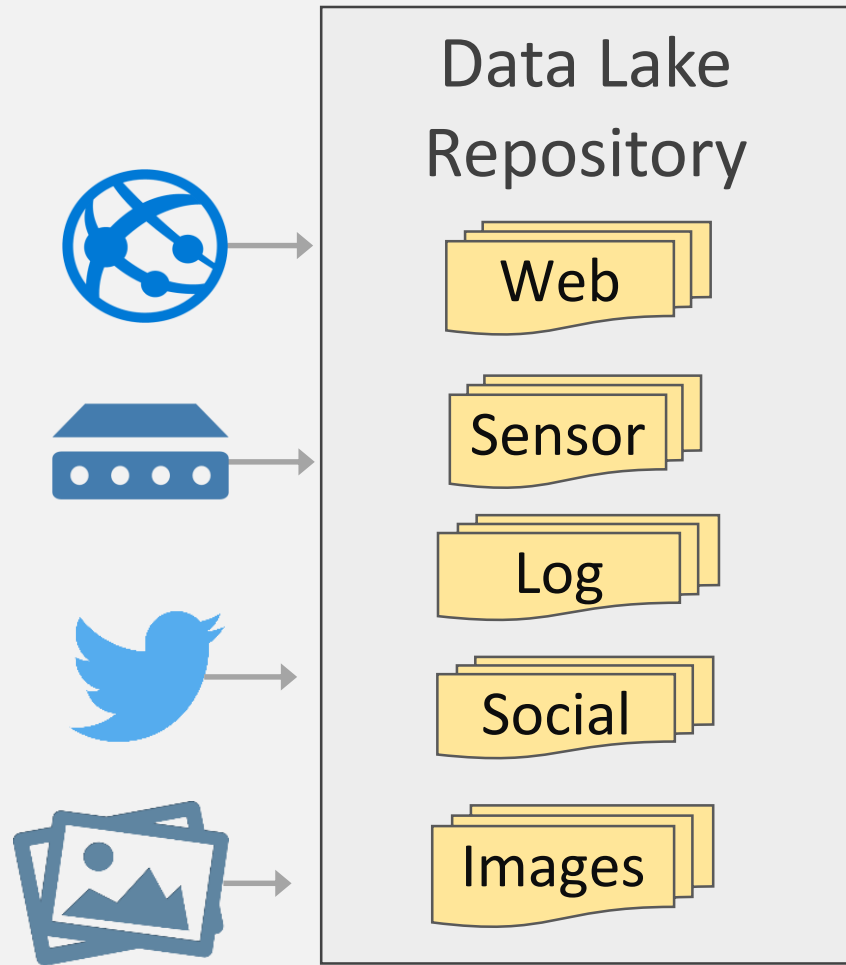
Objectives:

- 1 Scale up MPP compute nodes during:
 - Peak ETL data loads, or
 - High query volumes
- 2 Utilize existing on-premises data structures
- 3 Take advantage of cloud services for advanced analytics

Data Lake

Objectives, Challenges & Implementation Options

Data Lake



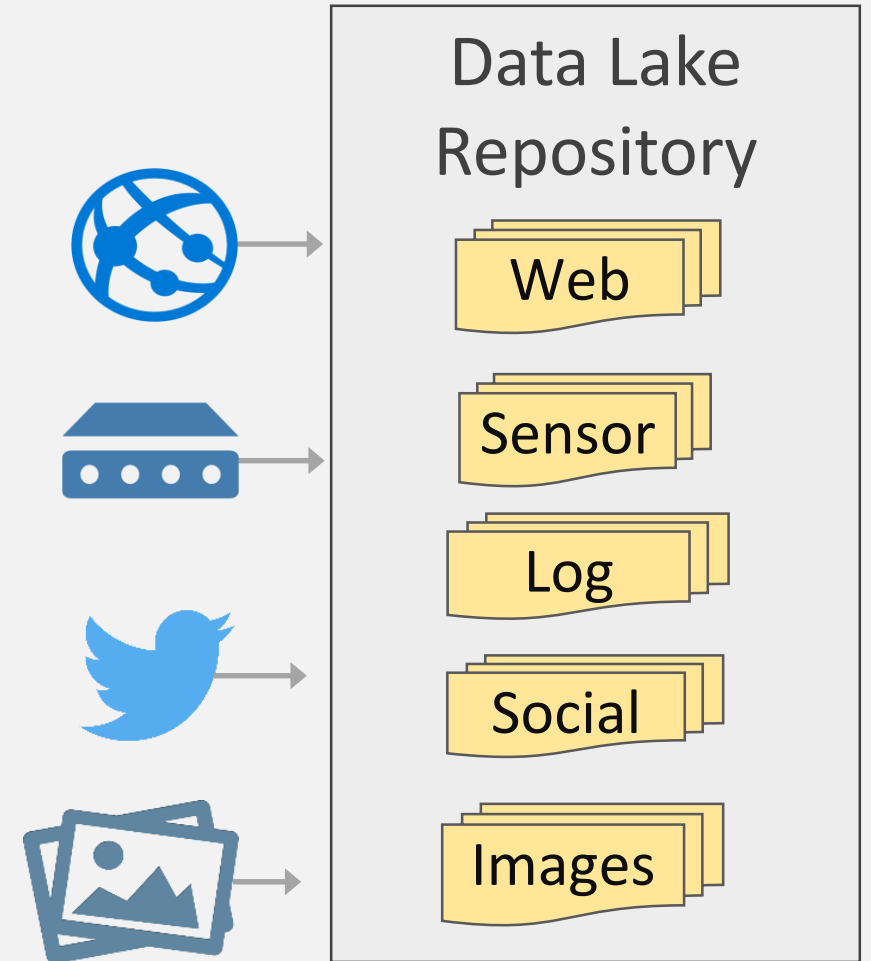
A **repository** for analyzing large quantities of disparate sources of data in its native format

One **architectural platform** to house all types of data:

- Machine-generated data (ex: IoT, logs)
- Human-generated data (ex: tweets, e-mail)
- Traditional operational data (ex: sales, inventory)

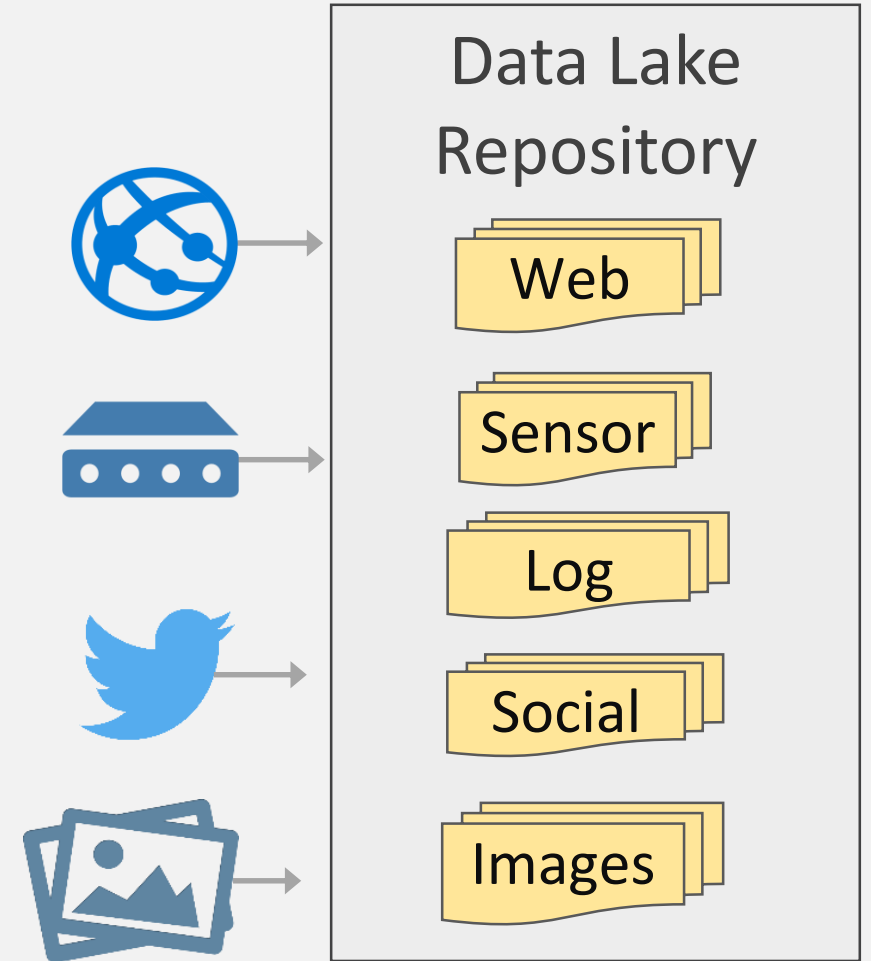
Objectives of a Data Lake

- ✓ Reduce up-front effort by ingesting data in any format without requiring a schema initially
- ✓ Make acquiring new data easy, so it can be available for data science & analysis quickly
- ✓ Store large volume of multi-structured data in its native format

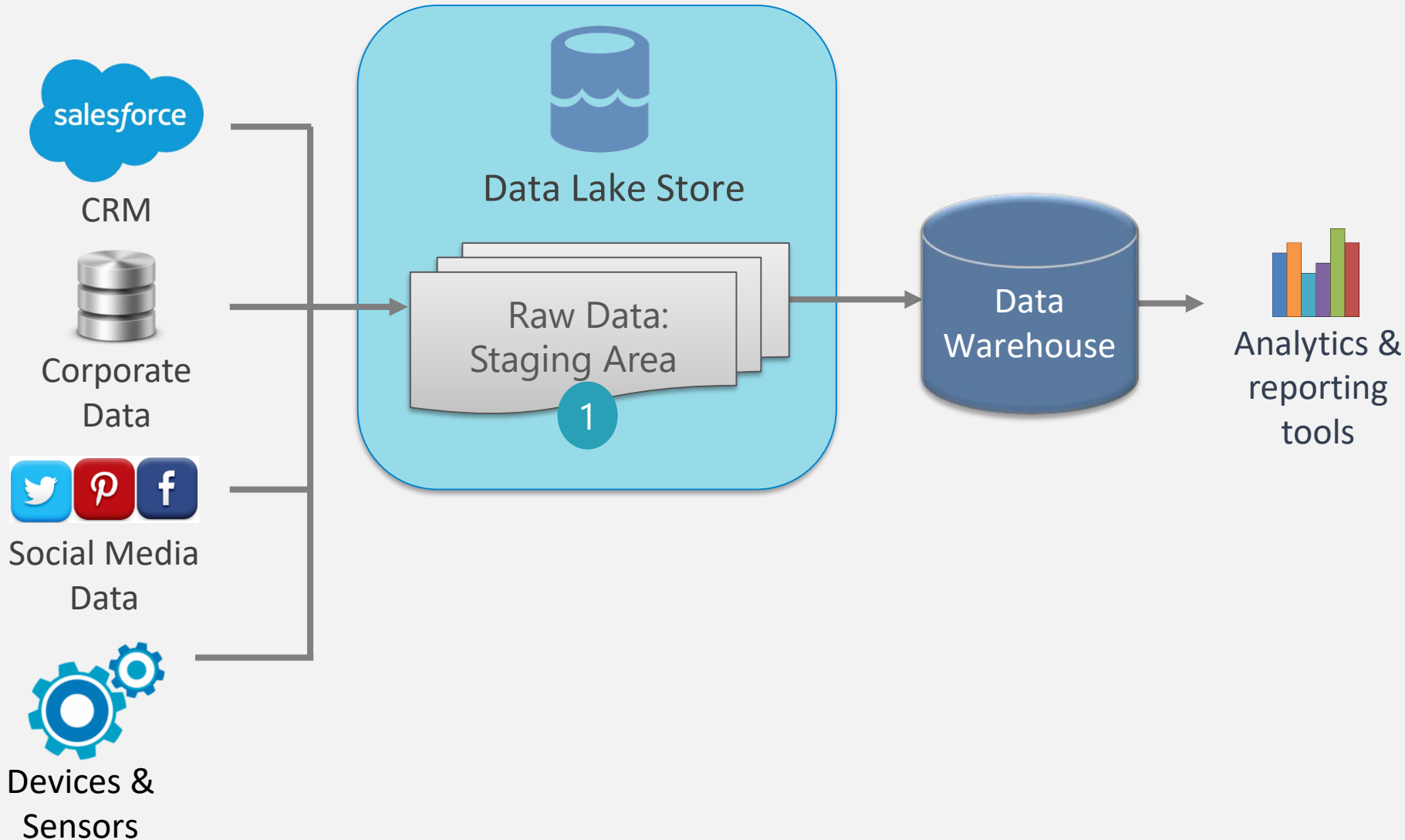


Objectives of a Data Lake

- ✓ Defer work to 'schematize' after value & requirements are known
- ✓ Achieve **agility** faster than a traditional data warehouse can
- ✓ Speed up **decision-making** ability
- ✓ Storage for **additional types of data** which were historically difficult to obtain



Data Lake as a Staging Area for DW

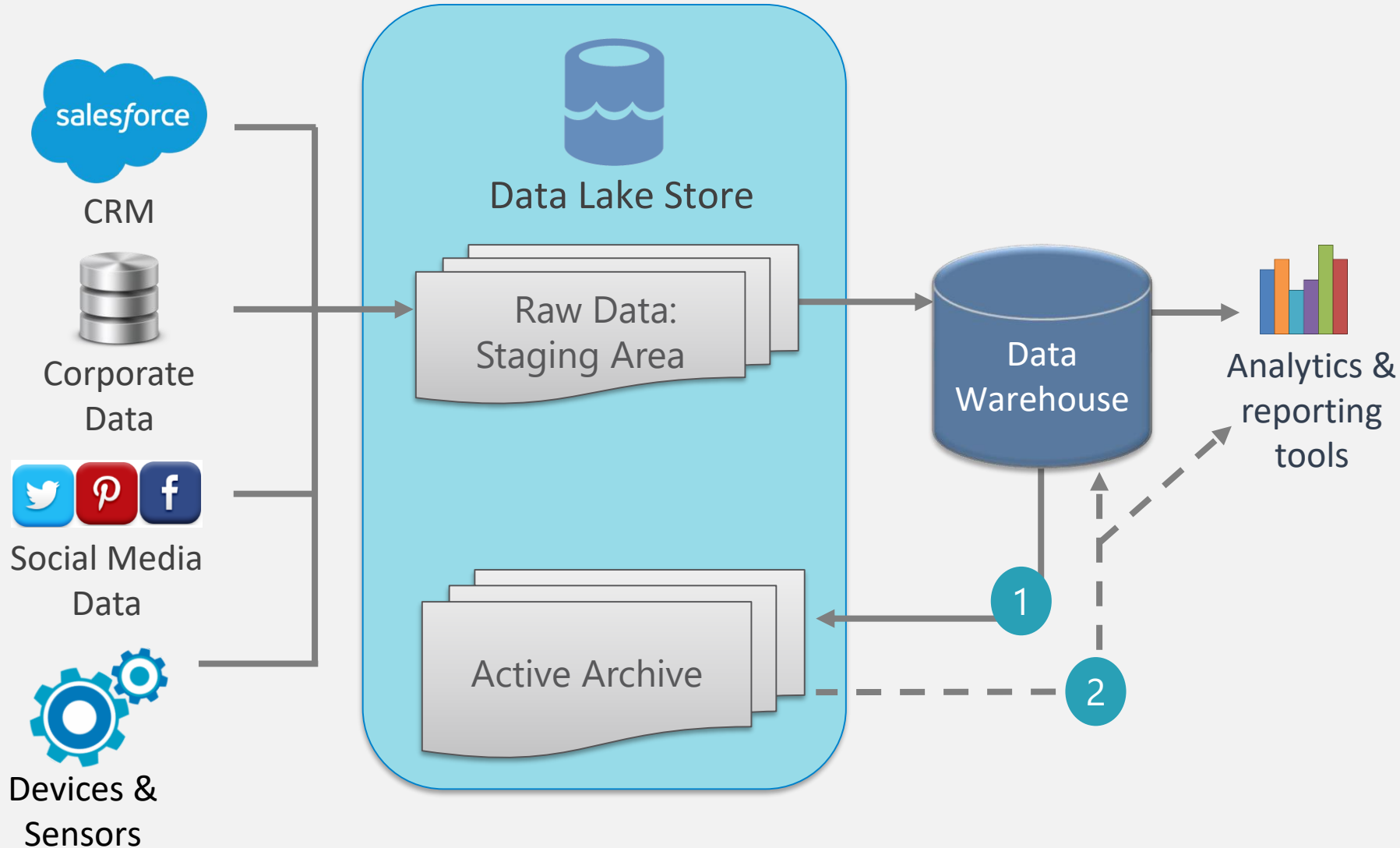


Strategy:

- Reduce storage needs in data warehouse
- Practical use for data stored in the data lake

- 1 Utilize the data lake as a landing area for DW staging area, instead of the relational database

Data Lake for Active Archiving

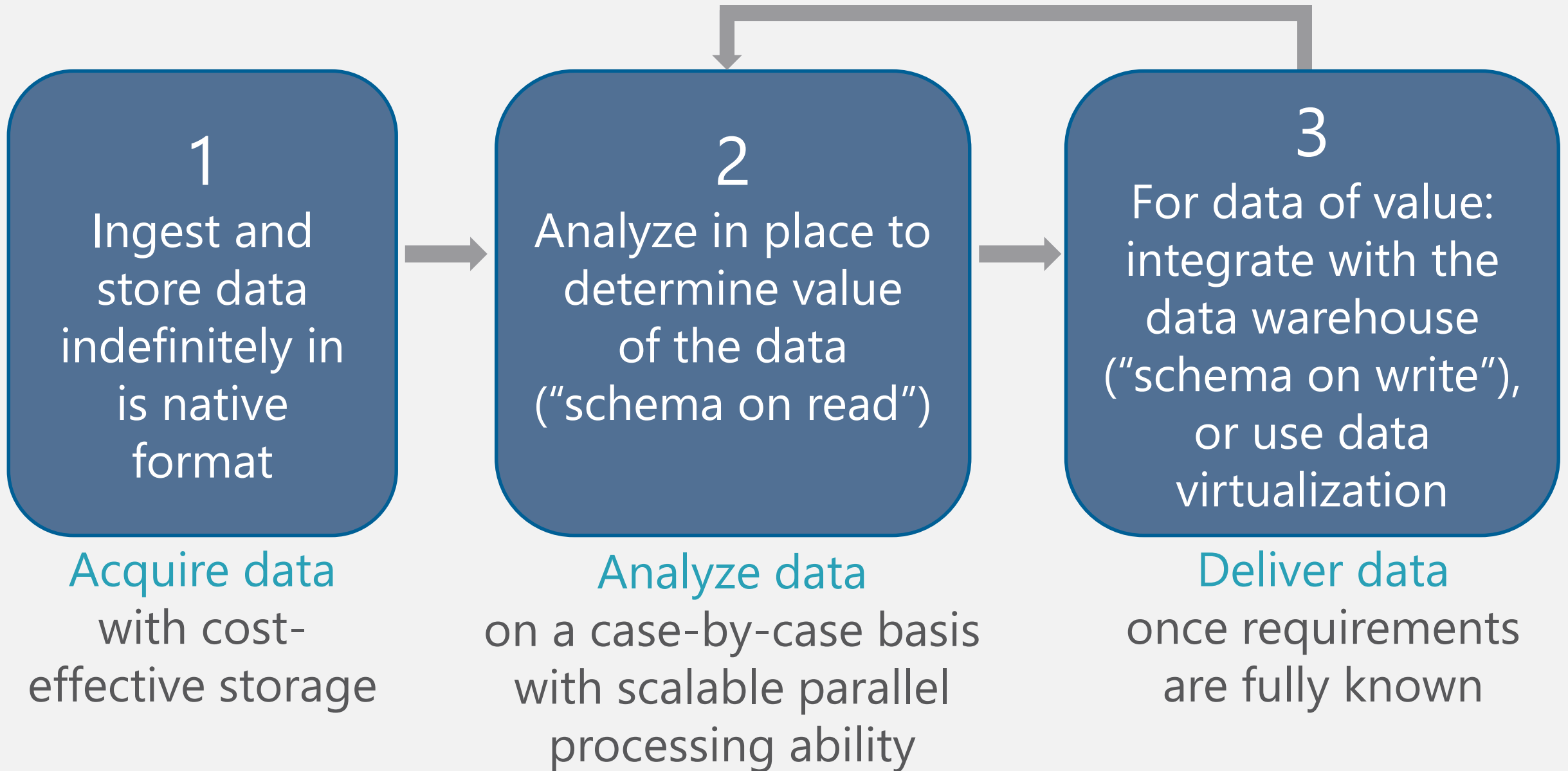


Strategy:

Data archival, with query ability available when needed

- 1 Archival process based on data retention policy
- 2 Federated query to access current & historical data

Iterative Data Lake Pattern



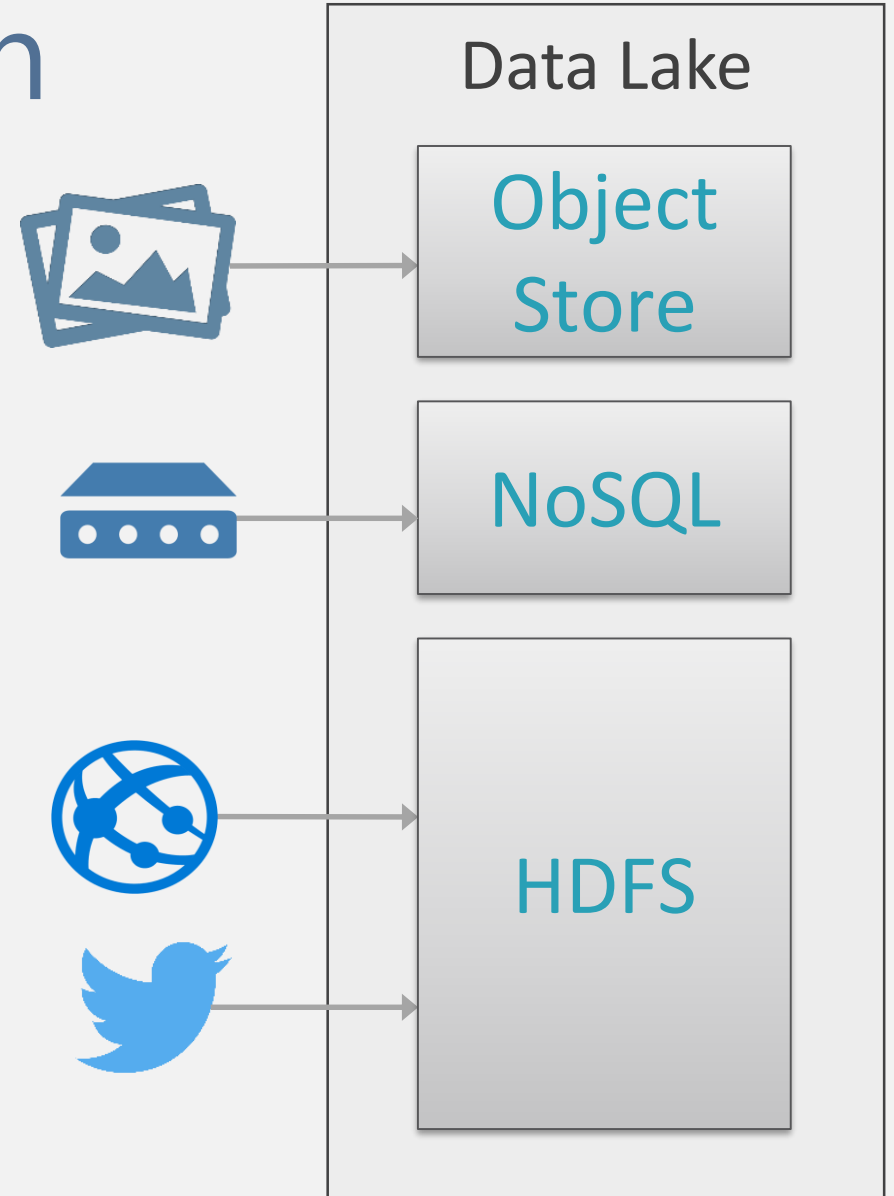
Data Lake Implementation

A data lake is a conceptual idea. It can be implemented with **one or more** technologies.

HDFS (Hadoop Distributed File Storage) is a very common option for data lake storage. However, Hadoop is not a requirement for a data lake. A data lake may also span > 1 Hadoop cluster.

NoSQL databases are also very common.

Object stores (like Amazon S3 or Azure Blob Storage) can also be used.



Coexistence of Data Lake & Data Warehouse



Data Lake Values:

- ✓ Agility
- ✓ Flexibility
- ✓ Rapid Delivery
- ✓ Exploration

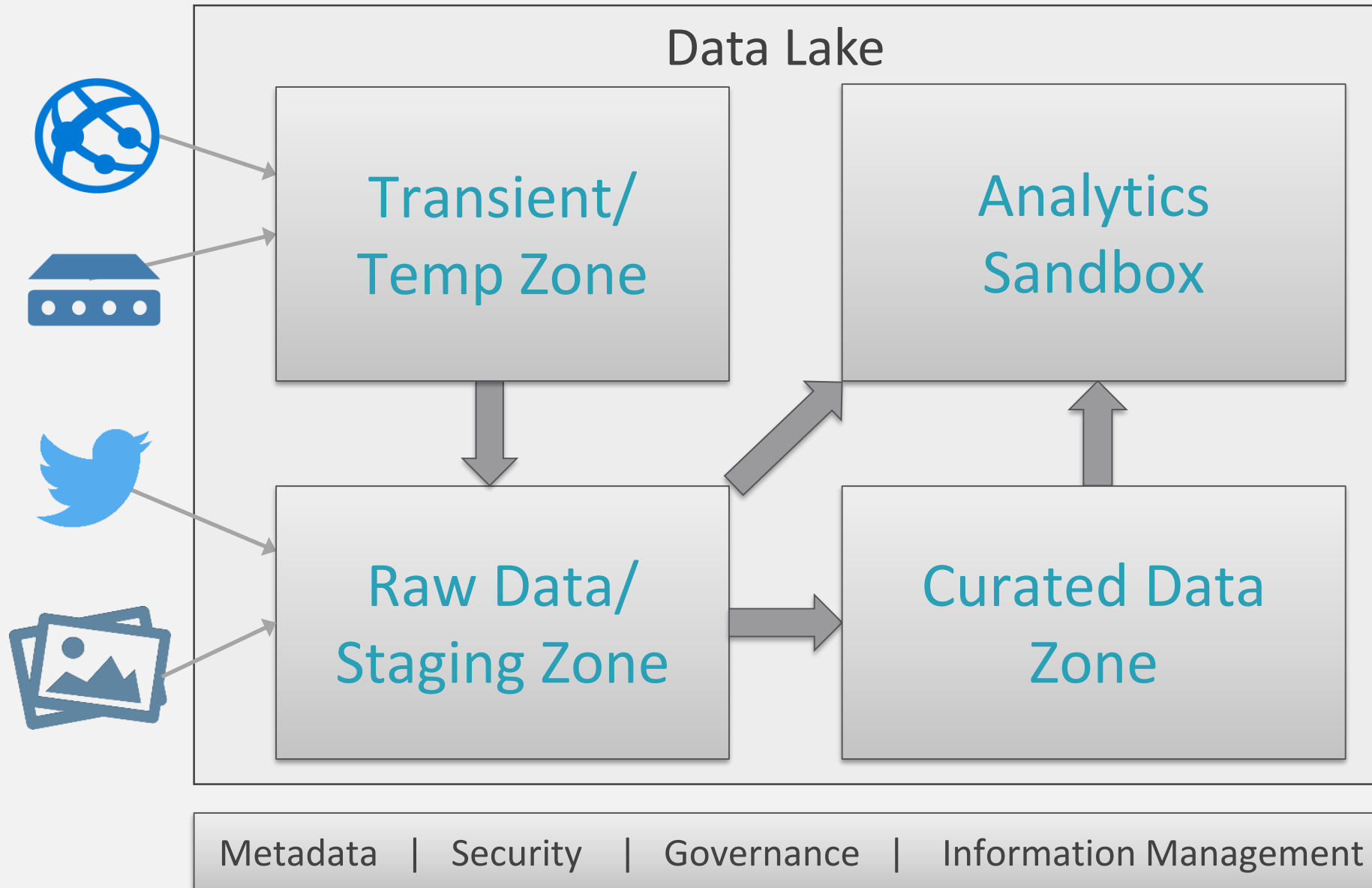


DW Values:

- ✓ Governance
- ✓ Reliability
- ✓ Standardization
- ✓ Security

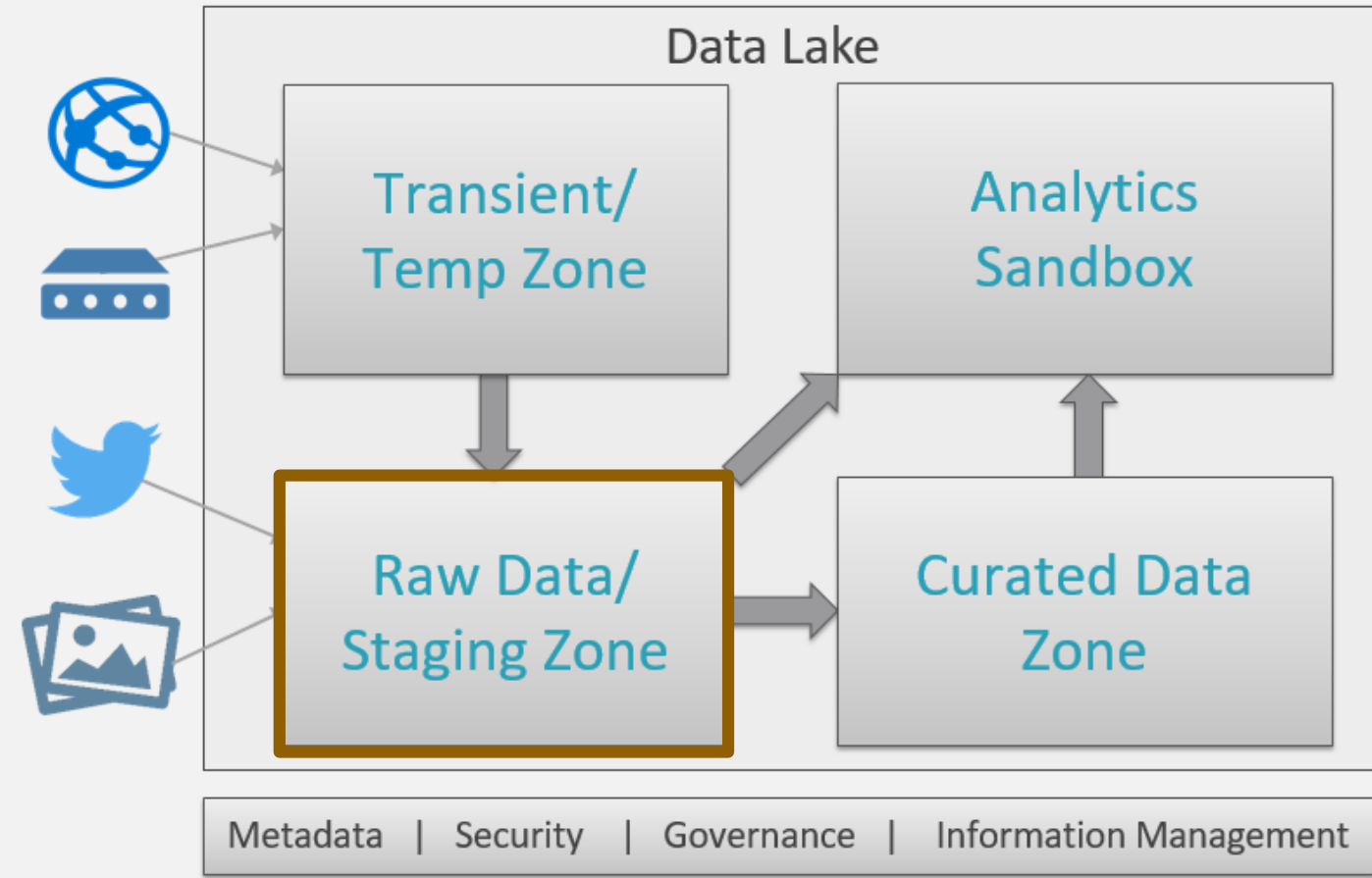
↓ Less effort	Data acquisition	↑ More effort
↑ More effort	Data retrieval	↓ Less effort

Zones in a Data Lake



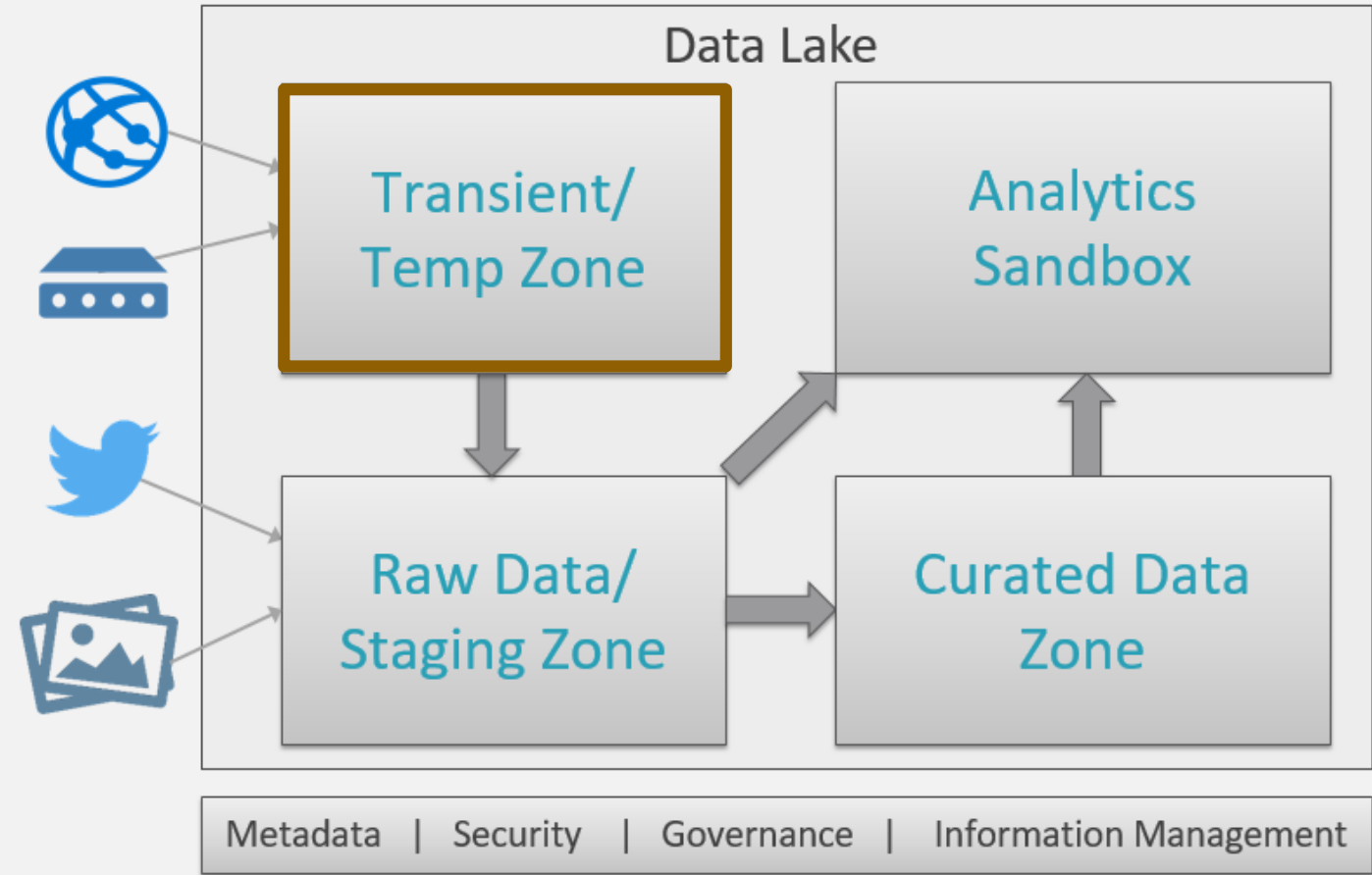
Raw Data Zone

- ✓ Raw data zone is **immutable** to change
- ✓ **History** is retained to accommodate **future unknown needs**
- ✓ **Staging** may be a distinct area on its own
- ✓ Supports **any type of data**
 - Streaming
 - Batch
 - One-time
 - Full load
 - Incremental load, etc...



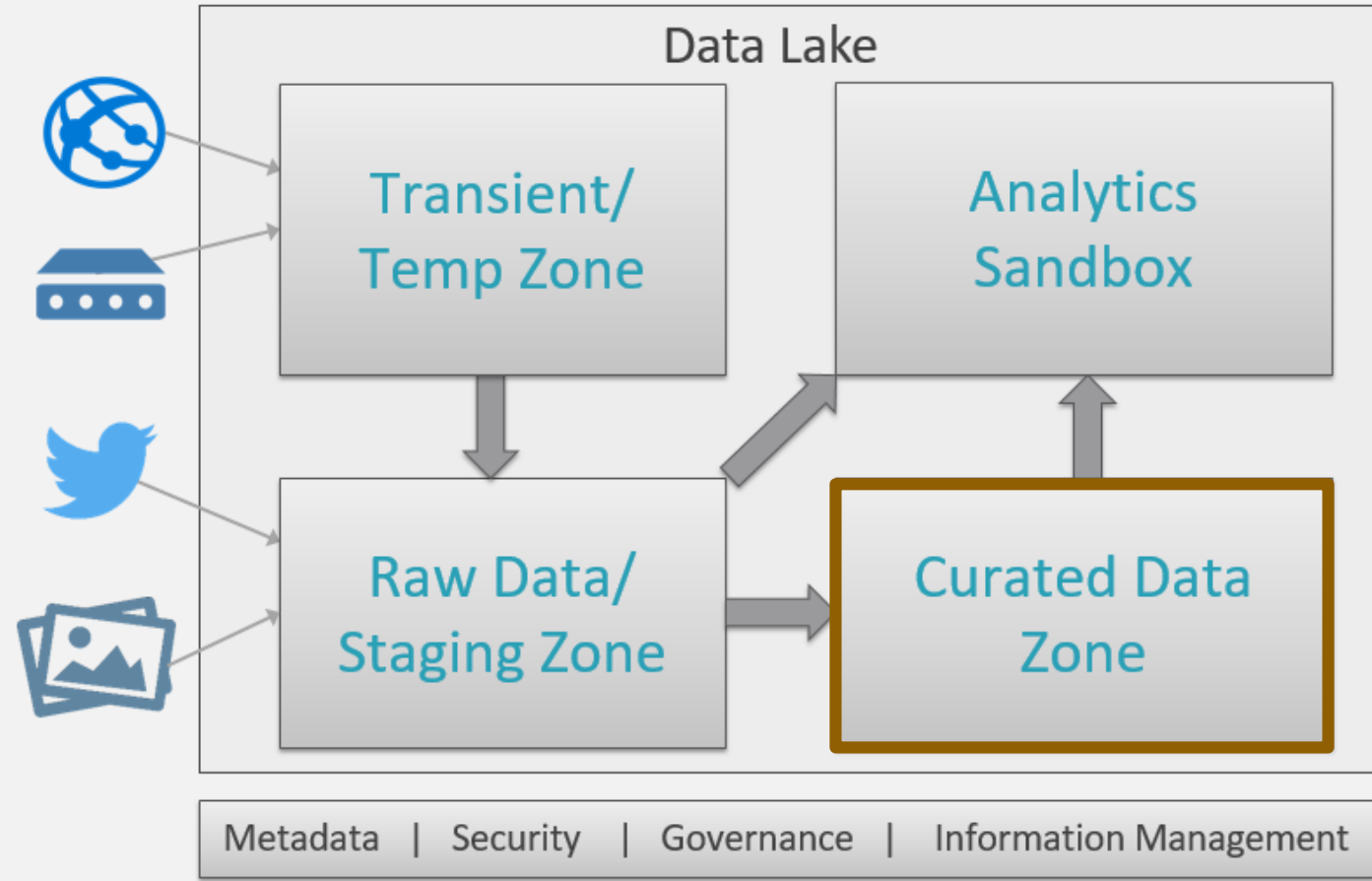
Transient Zone

- ✓ Useful when **data quality** or **validity** checks are necessary before data can be landed in the Raw Zone
- ✓ All landing zones considered “kitchen area” with **highly limited access**
 - Transient Zone
 - Raw Data Zone
 - Staging Area



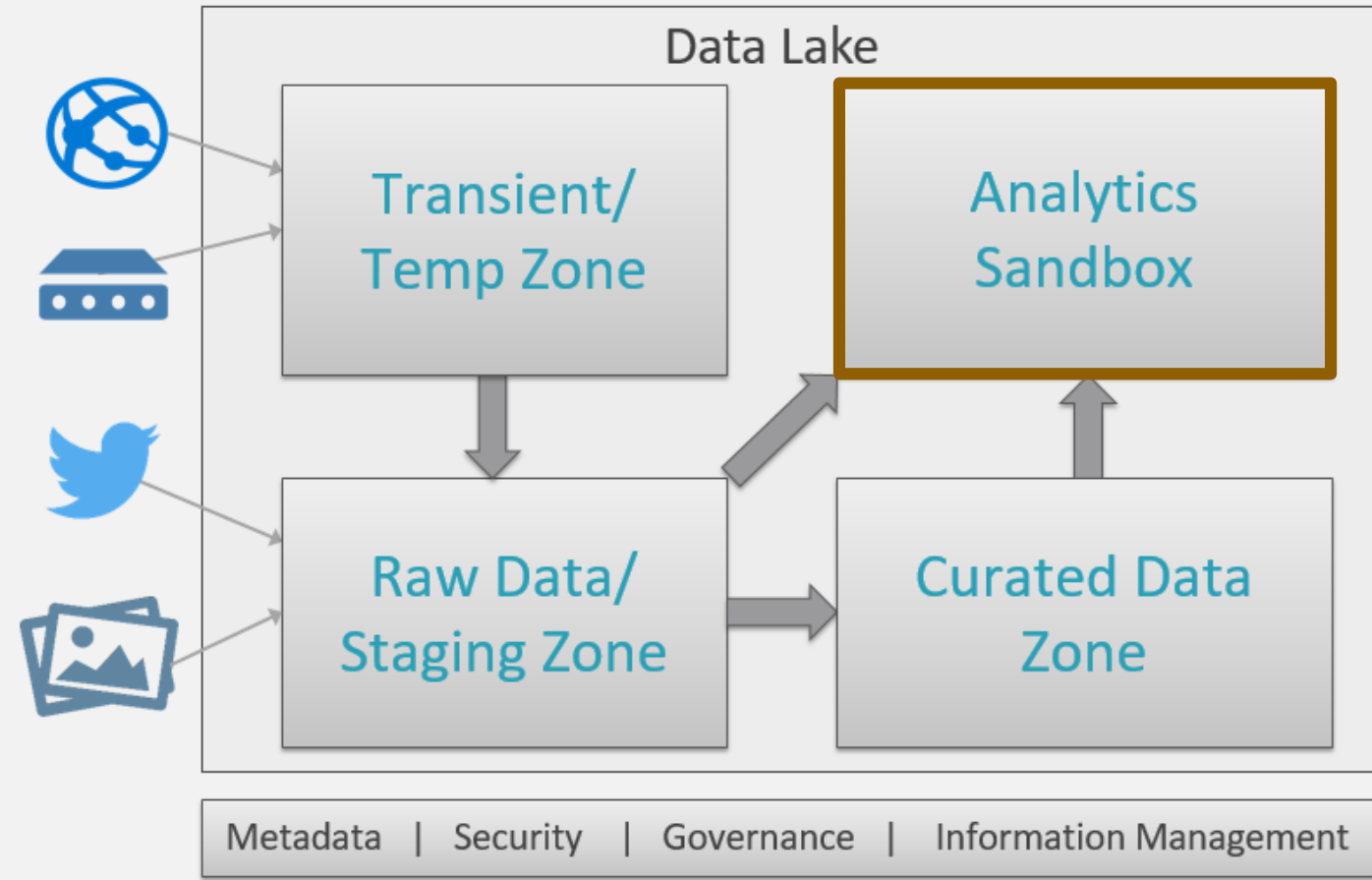
Curated Data Zone

- ✓ Cleansed, organized data for **data delivery**:
 - Data consumption
 - Federated queries
 - Provides data to other systems
- ✓ Most **self-service** data access occurs from the Curated Data Zone
- ✓ Standard **governance** & **security** in the Curated Data Zone

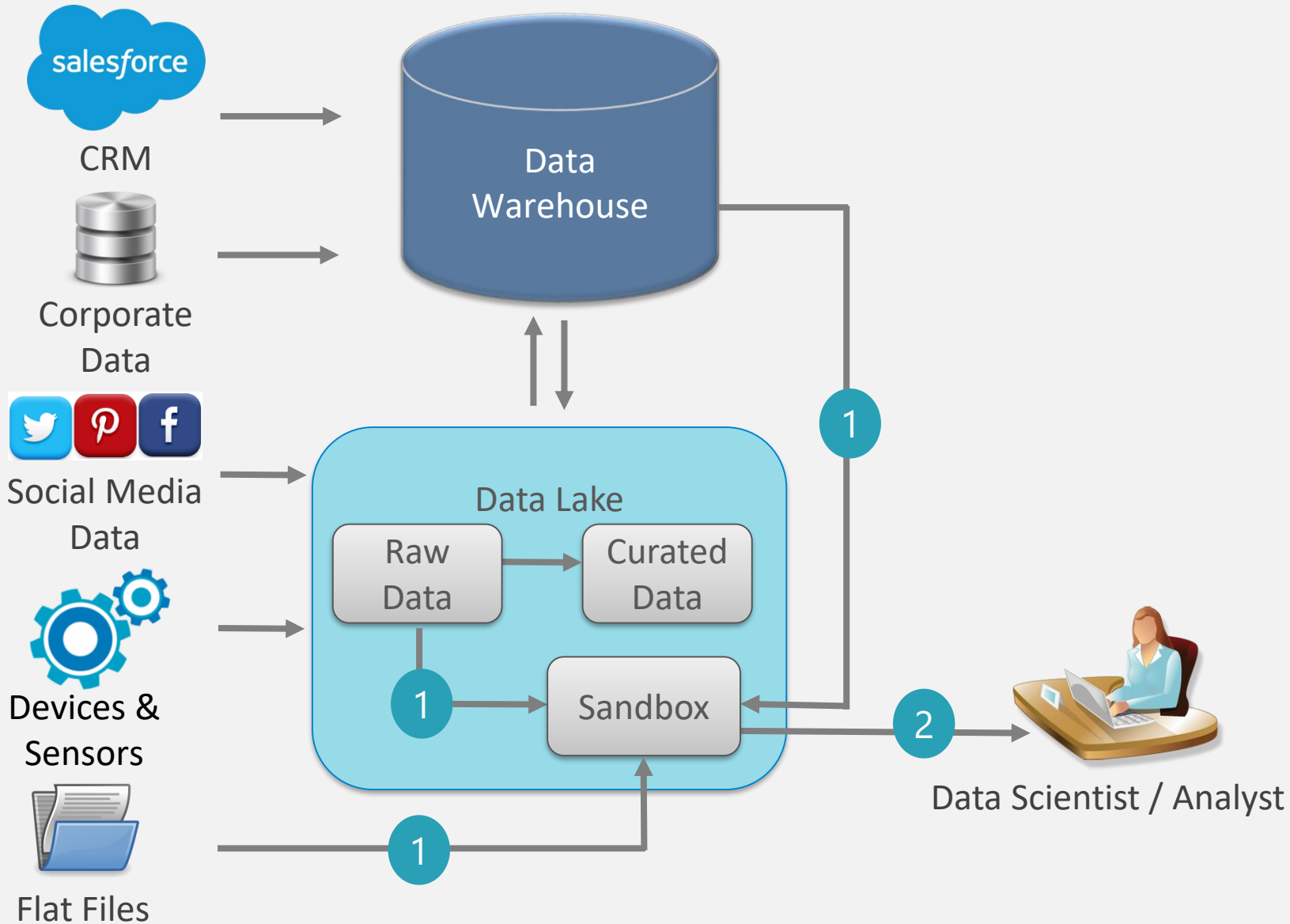


Analytics Sandbox

- ✓ Data science and **exploratory** activities
- ✓ **Minimal governance** of the Analytics Sandbox
- ✓ Valuable efforts are **"promoted"** from Analytics Sandbox to the Curated Data Zone or to the data warehouse



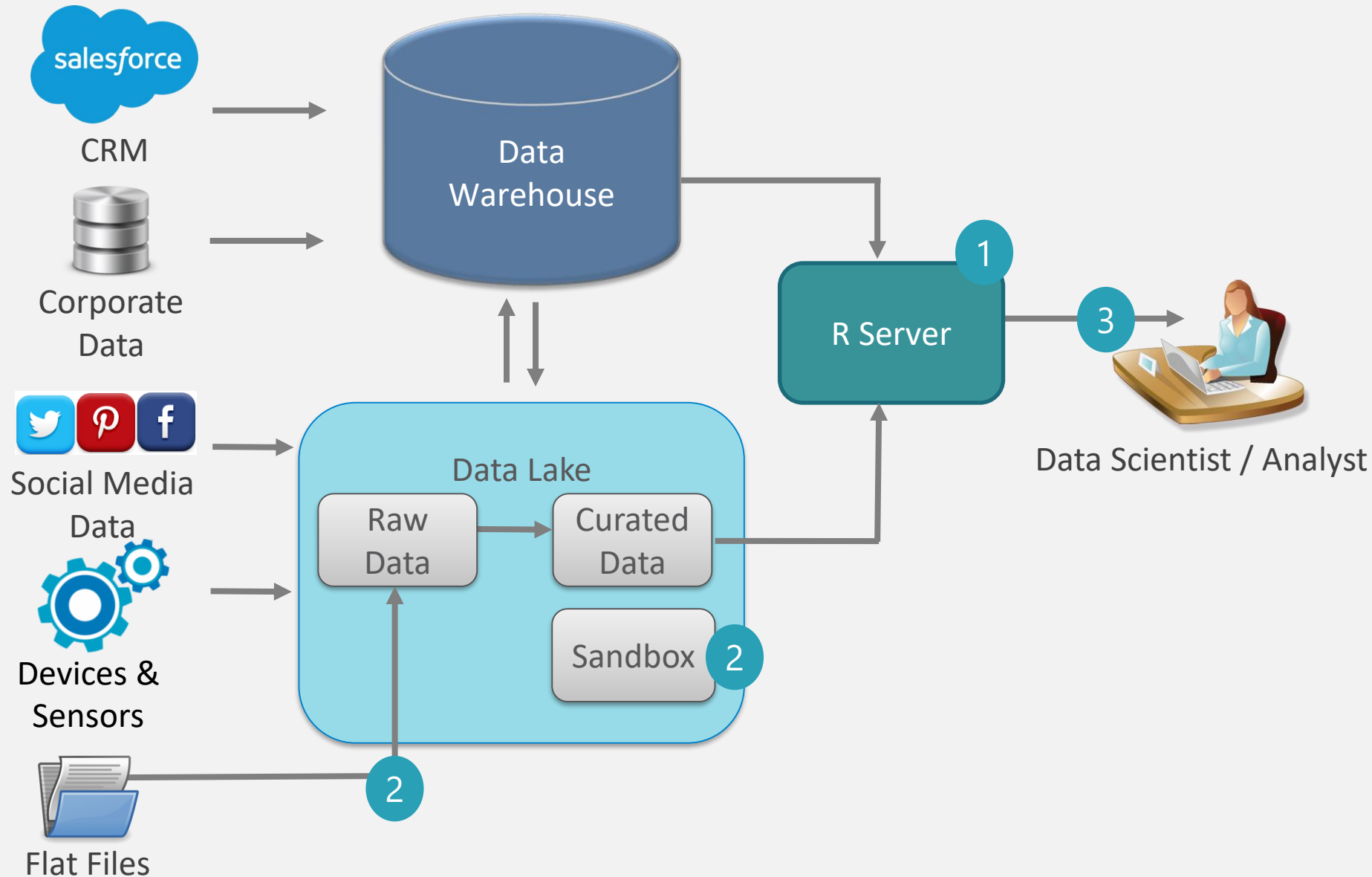
Sandbox Solutions: Develop



Objective:

- 1 Utilize sandbox area in the data lake for data preparation
- 2 Execution of R scripts from local workstation for **exploratory** data science & advanced analytics scenarios

Sandbox Solutions: Operationalize



Objective:

- 1 Trained model is promoted to run in production server environment
- 2 Sandbox use is discontinued once solution is promoted
- 3 Execution of R scripts from server for **operationalized** data science & advanced analytics scenarios

Organizing the Data Lake

Plan the structure based on **optimal data retrieval**.
The organization pattern should be self-documenting.

Organization is frequently based upon:

Subject
area

Time
partitioning

Security
boundaries

Downstream
app/purpose

Metadata capabilities
of your technology will
have a *big* impact on
how you choose to
handle organization.

-- The objective is to avoid a chaotic data swamp --

Organizing the Data Lake

Raw Data Zone

Subject Area

Data Source

Object

Date Loaded

File(s)

Sales

Salesforce

CustomerContacts

2016

12

2016_12_01

CustCct.txt

Example 1

Pros: Subject area at top level, organization-wide,
Partitioned by time

Cons: No obvious security or organizational
boundaries

Curated Data Zone

Purpose

Type

Snapshot Date

File(s)

Sales Trending Analysis

Summarized

2016_12_01

SalesTrend.txt



Organizing the Data Lake

Raw Data Zone

Organization Unit

Subject Area

Data Source

Object

Date Loaded

File(s)

East Division

Sales

Salesforce

CustomerContacts

2016

12

2016_12_01

CustCct.txt

Example 2

Pros: Security at the organizational level,
Partitioned by time

Cons: Potentially siloed data, duplicated data

Curated Data Zone

Organizational Unit

Purpose

Type

Snapshot Date

File(s)

East Division

Sales Trending Analysis

Summarized

2016_12_01

SalesTrend.txt



Organizing the Data Lake

Other options which affect organization and/or metadata:

Data Retention Policy

Temporary data
Permanent data
Applicable period (ex: project lifetime)
etc...

Business Impact / Criticality

High (HBI)
Medium (MBI)
Low (LBI)
etc...

Owner / Steward / SME

Probability of Data Access

Recent/current data
Historical data
etc...

Confidential Classification

Public information
Internal use only
Supplier/partner confidential
Personally identifiable information (PII)
Sensitive – financial
Sensitive – intellectual property
etc...

Challenges of a Data Lake

Technology

- ✓ Complex, multi-layered architecture
- ✓ Unknown storage & scalability
- ✓ Data retrieval
- ✓ Working with un-curated data
- ✓ Performance
- ✓ Change management

Process

- ✓ Right balance of deferred work vs. up-front work
- ✓ Ignoring established best practices for data management
- ✓ Data quality
- ✓ Governance
- ✓ Security

People

- ✓ Expectations
- ✓ Data stewardship
- ✓ Redundant effort
- ✓ Skills required to make analytical use of the data

Ways to Get Started with a Data Lake

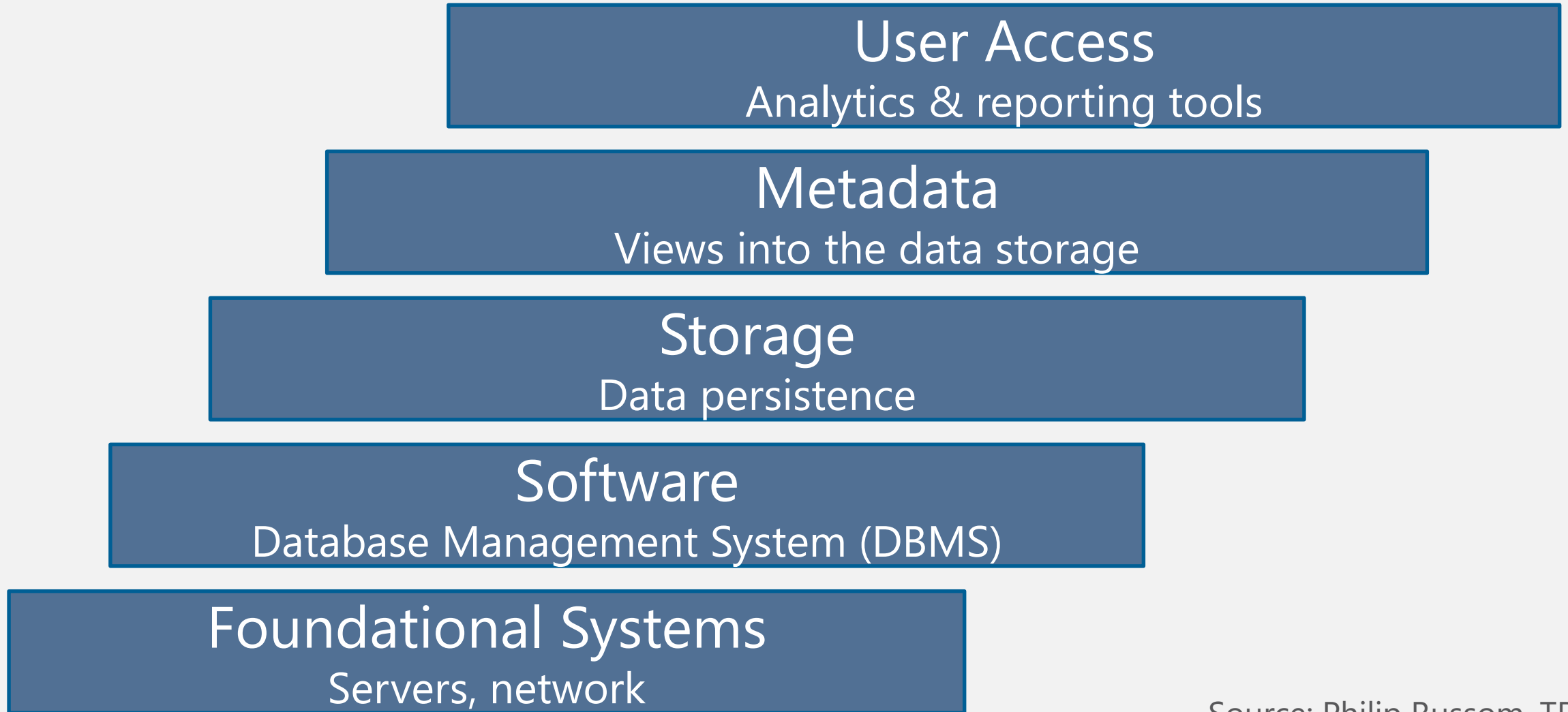
1. Data lake as **staging** area for DW
2. Offload **archived** data from DW back to data lake
3. Ingest a **new type of data** to allow time for longer-term planning

Getting Real Value From the Data Lake

1. Selective integration with the **data warehouse** – physical or virtual
2. Data science **experimentation with APIs**
3. **Analytical toolsets** on top of the data lake
 - Hive
 - Pig
 - Spark
 - Impala
 - Presto
 - Drill
 - Solr
 - Kafka
 - etc...
4. **Query interfaces** on top of the data lake using familiar technologies
 - SQL-on-Hadoop
 - OLAP-on-Hadoop
 - Data virtualization
 - Metadata
 - Data cataloging

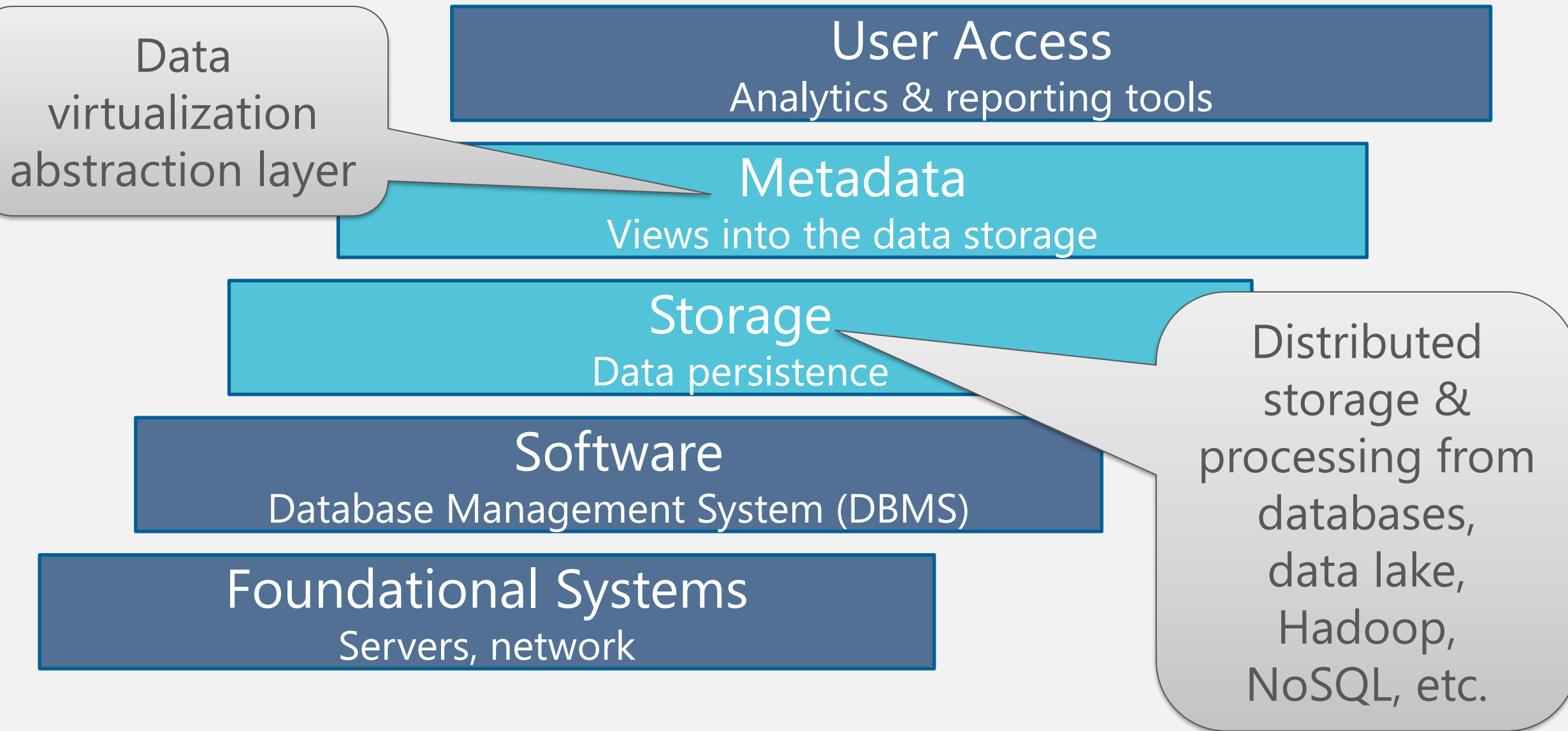
The Logical Data Warehouse & Data Virtualization

Conceptual Data Warehouse Layers



Source: Philip Russom, TDWI

Logical Data Warehouse



Logical Data Warehouse

An LDW is a data warehouse which uses “repositories, virtualization, and distributed processes in combination.”

7 major components of an LDW:

Data
Virtualization

Repository
Management

Auditing &
Performance
Services

Distributed
Processing

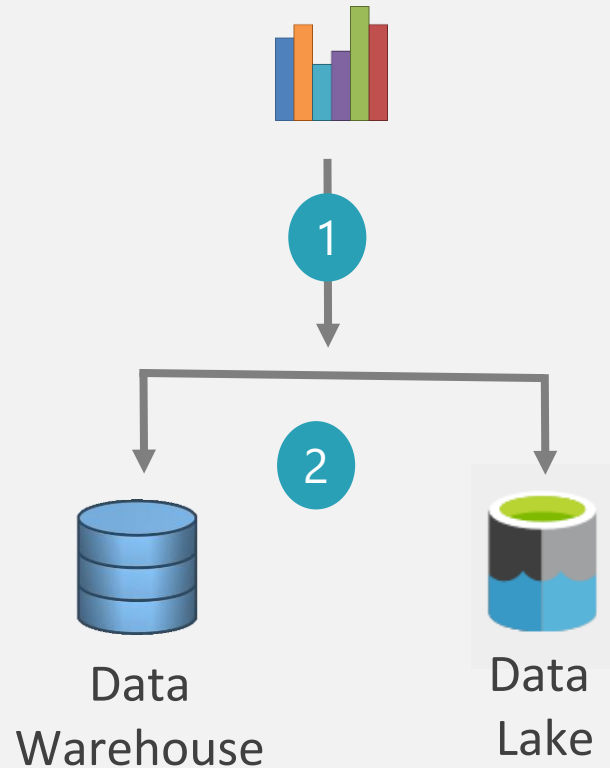
Metadata
Management

Taxonomy/Ontology
Resolution

Service Level
Agreement
Management

We will focus on
these two aspects

Data Virtualization

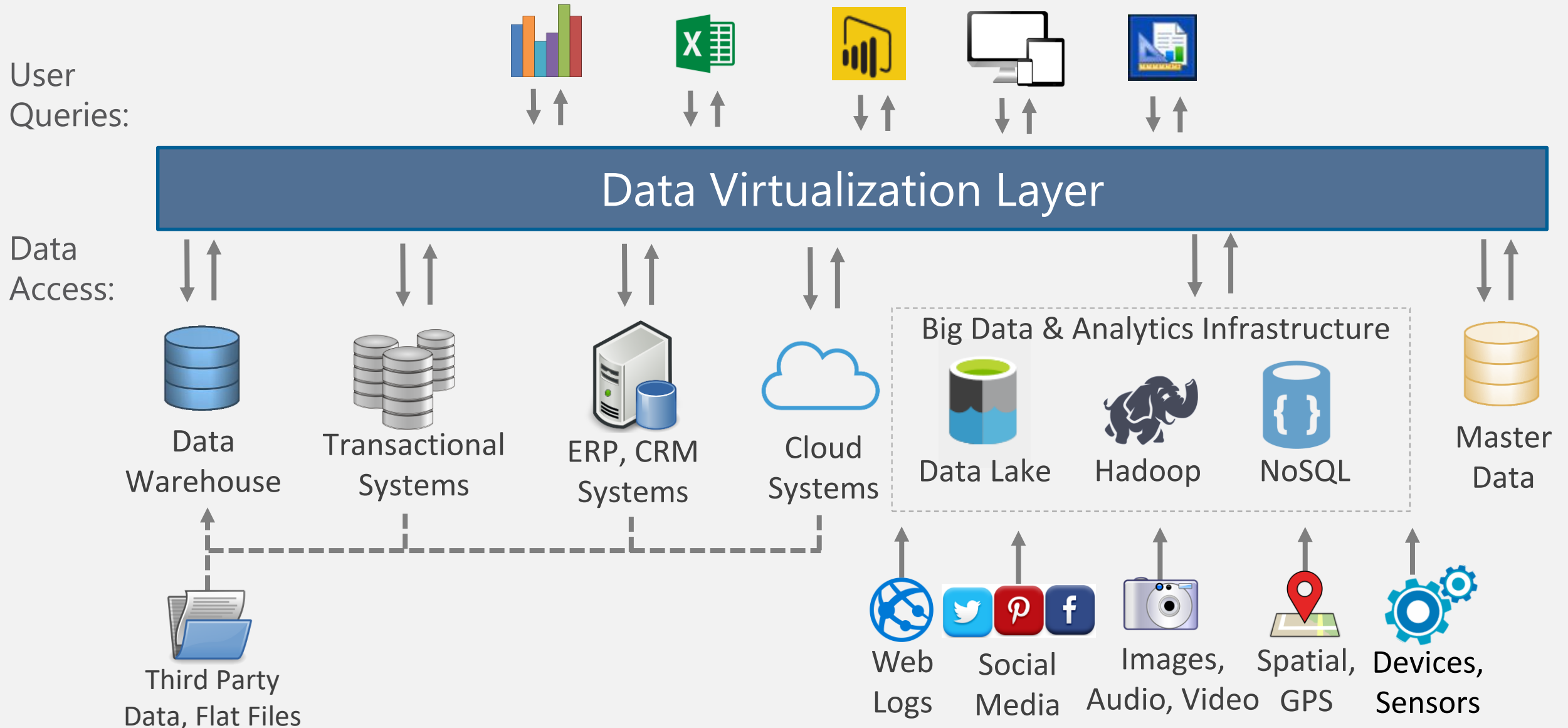


Objective:

Ability to access various data platforms without doing full data integration

- 1 User issues query from analytical tool of choice
- 2 Data returned from this federated query across > 1 data source

Data Virtualization



Objectives of Data Virtualization

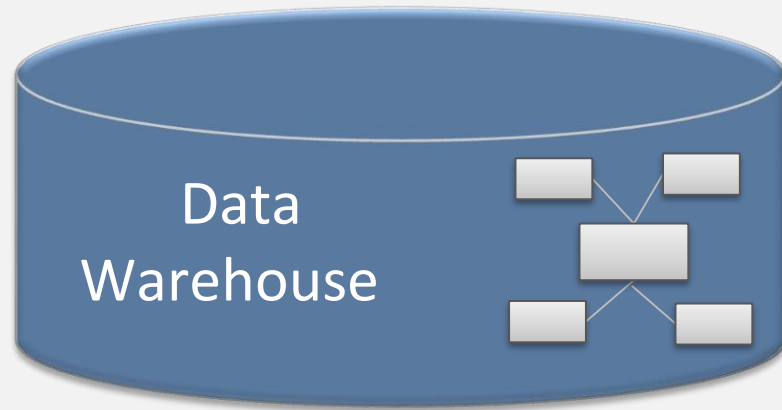
- ✓ Add flexibility & speed to a traditional data warehouse
- ✓ Make current data available quickly 'where it lives' useful when:
 - Data is too large to practically move
 - Data movement window is small
 - Data cannot legally move out of a geographic region
- ✓ Enable user access to various data platforms
- ✓ Reduce data latency; enable near real-time analytics
- ✓ Reduce data redundancy & processing time
- ✓ Facilitate a polyglot persistence strategy (use the best storage for the data)

Challenges of Data Virtualization

- ✓ **Performance**; impact of adding **reporting load on source** systems
- ✓ Limited ability to handle **data quality & referential integrity** issues
- ✓ **Complexity** of virtualization layer
(ex: different data formats, query languages, data granularities)
- ✓ **Change management** & managing lifecycle+impact of changes
- ✓ Lack of **historical** data; inability to do **point-in-time** historical analysis
- ✓ Consistent **security, compliance & auditing**
- ✓ **Real-time** reporting can be confusing with its frequent data changes
- ✓ **Downtime** of underlying data sources
- ✓ **Auditing & reconciling** abilities across systems

Wrap-Up & Questions

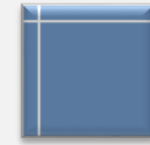
Growing an Existing DW Environment



Data Lake



Hadoop



In-Memory
Model



NoSQL

Growing a DW:

- ✓ Data modeling strategies
- ✓ Partitioning
- ✓ Clustered columnstore index
- ✓ In-memory structures
- ✓ MPP (massively parallel processing)

Extending a DW:

- ✓ Complementary data storage & analytical solutions
- ✓ Cloud & hybrid solutions
- ✓ Data virtualization (virtual DW)

-- Grow around your existing data warehouse --

Final Thoughts

Traditional data warehousing still is important, but needs to **co-exist** with other platforms. **Build around your existing DW infrastructure.**

Plan your data lake with **data retrieval** in mind.

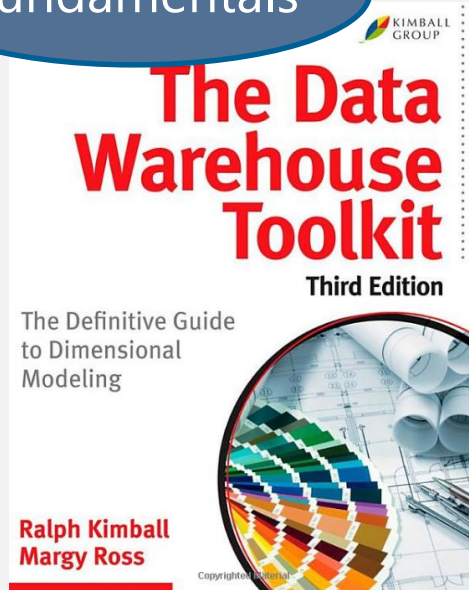
Expect to balance ETL with some (perhaps limited) **data virtualization** techniques in a **multi-platform** environment.

Be fully aware of data lake & data virtualization **challenges** in order to craft your own achievable & realistic **best practices**.

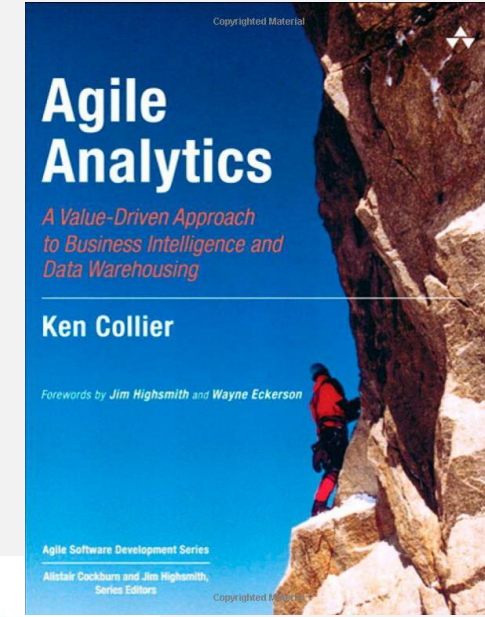
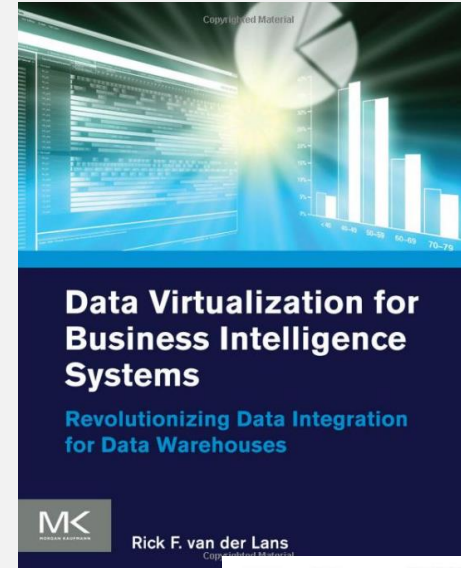
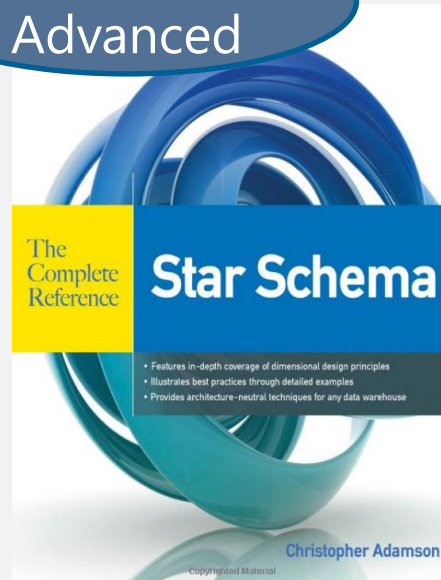
Plan to work in an **agile** fashion. Conduct frequent **proof of concept** projects to prove assumptions.

Recommended Resources

Fundamentals

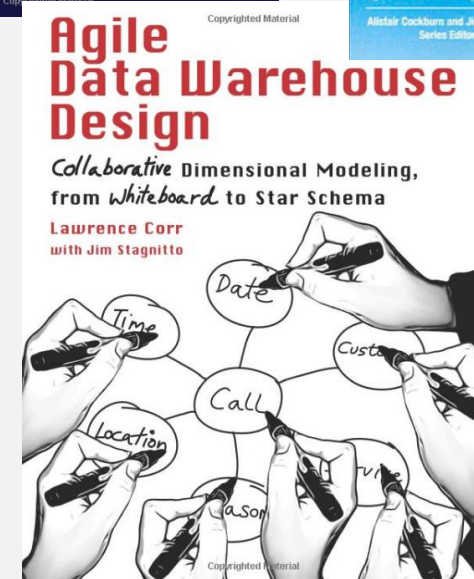


More Advanced



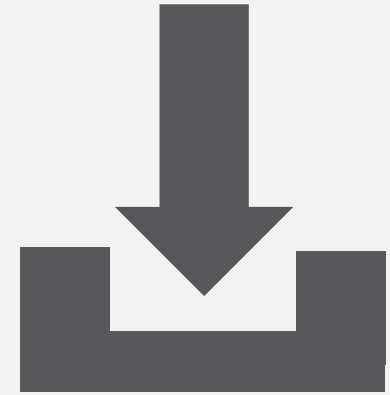
Recommended Whitepaper:

How to Build An Enterprise Data Lake:
Important Considerations Before Jumping
In by Mark Madsen, Third Nature Inc.



Thank You!

To download a copy of this presentation:
SQLChick.com "Presentations & Downloads" page



Melissa Coates
Solution Architect,
BlueGranite

Blog: sqlchick.com
Twitter: @sqlchick

*Creative Commons License:
Attribution-NonCommercial-NoDerivative Works 3.0*



Appendix A: Terminology

Terminology

Logical Data Warehouse

Facilitates access to various source systems via data virtualization, distributed processing, and other system components

Data Virtualization

Access to one or more distributed data sources without requiring the data to be physically materialized in another data structure

Data Federation

Accesses & consolidates data from multiple distributed data stores

Terminology

Polyglot Persistence

Using the most effective data storage technology to handle different data storage needs

Schema on Write

Data structure is applied at design time, requiring additional up-front effort to formulate a data model

Schema on Read

Data structure is applied at query time rather than when the data is initially stored; deferred up-front effort facilitates agility

Terminology

Defining the Components of a Modern Data Warehouse

<http://www.sqlchick.com/entries/2017/1/9/defining-the-components-of-a-modern-data-warehouse-a-glossary>

Appendix B:

What Makes A Data Warehouse "Modern"

What Makes a Data Warehouse “Modern”



Variety of subject areas & **data sources** for analysis with capability to handle large **volumes** of data



Expansion beyond a single relational DW/data mart structure to include **Hadoop**, **Data Lake**, or **NoSQL**



Logical design across **multi-platform architecture** balancing scalability & performance



Data virtualization in addition to data integration

What Makes a Data Warehouse “Modern”



Support for **all types & levels of users**



Flexible **deployment** (including mobile) which is **decoupled** from tool used for development



Governance model to support trust and security, and **master data management**



Support for **promoting self-service solutions** to the corporate environment

What Makes a Data Warehouse “Modern”



Ability to facilitate **near real-time** analysis on **high velocity** data (Lambda architecture)



Support for **advanced analytics**



Agile delivery approach with fast delivery cycle



Hybrid integration with **cloud** services



APIs for downstream access to data

What Makes a Data Warehouse “Modern”



Some DW **automation** to improve speed, consistency, & flexibly adapt to change



Data cataloging to facilitate data search & document business terminology



An **analytics sandbox** or workbench area to facilitate agility within a **bimodal BI** environment



Support for **self-service BI** to augment corporate BI;
Data discovery, data exploration, self-service data prep

Appendix C:

Challenges With Modern Data Warehousing

Challenges with Modern Data Warehousing

Reducing time to
value

Minimizing chaos

Evolving &
maturing
technology

Balancing 'schema
on write' with
'schema on read'

How strict to be
with dimensional
design?

Agility

Challenges with Modern Data Warehousing

Hybrid
scenarios

Multi-
platform
infrastructure

Ever-
increasing
data volumes

File type &
format
diversity

Real-time
reporting
needs

Effort & cost
of data
integration

Broad skillsets
needed

Complexity

Challenges with Modern Data Warehousing

Self-service solutions
which challenge
centralized DW

Managing 'production'
delivery from IT and
user-created solutions

Handling ownership
changes (promotion) of
valuable solutions

Balance with Self-Service Initiatives

Challenges with Modern Data Warehousing

Data quality

Master data

Security

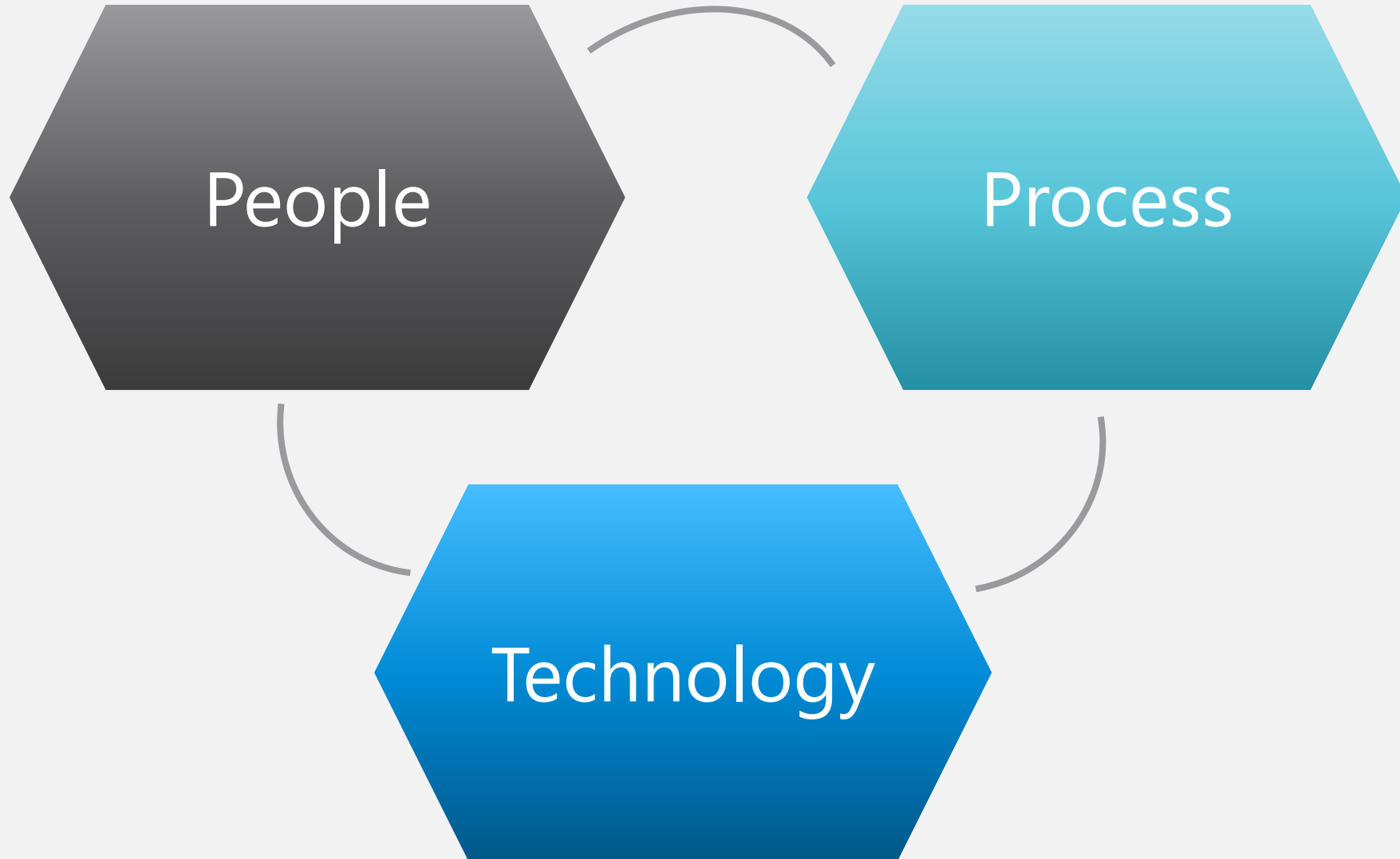
Governance

The Never-Ending Challenges

Appendix D:

Challenges of a Data Lake

Challenges of a Data Lake



Challenges of a Data Lake: Technology

Complex, multi-layered architecture

- ✓ Polyglot persistence strategy
- ✓ Architectures & toolsets are emerging and maturing

Unknown storage & scalability

- ✓ Can we realistically store “everything?”
- ✓ Cloud deployments are attractive when scale is undetermined

Working with un-curated data

- ✓ Inconsistent dates and data types
- ✓ Data type mismatches
- ✓ Missing or incomplete data
- ✓ Flex-field data which can vary per record
- ✓ Different granularities
- ✓ Incremental data loads
- etc...

Challenges of a Data Lake: Technology

Performance

- ✓ Trade-offs between latency, scalability, & query performance
- ✓ Monitoring & auditing

Data retrieval

- ✓ Easy access for data consumers
- ✓ Organization of data to facilitate data retrieval
- ✓ Business metadata is **critical** for making sense of the data

Change management

- ✓ File structure changes (inconsistent 'schema-on-read')
- ✓ Integration with master data
- ✓ Inherent risks associated with a highly flexible repository
- ✓ Maintaining & updating over time (meeting future needs)

Challenges of a Data Lake: Process

Finding right balance
of agility
(deferred work vs.
up-front work)

- ✓ Risk & complexity are still there – just shifted
- ✓ Finding optimal level of chaos which invites experimentation
- ✓ When schema-on-read is appropriate (Temporarily? Permanently?)
- ✓ How the data lake will coexist with the data warehouse
- ✓ How to operationalize self-service/sandbox work from analysts

Data quality

- ✓ How to reconcile or confirm accuracy of results
- ✓ Data validation between systems

Challenges of a Data Lake: Process

Governance & security

- ✓ Challenging to implement data governance
- ✓ Difficult to enforce standards
- ✓ Securing and obfuscating confidential data
- ✓ Meeting compliance and regulatory requirements

Ignoring established best practices for data management

- ✓ New best practices are still evolving
- ✓ Repeating the same data warehousing failures (ex: silos)
- ✓ Erosion of credibility due to disorganization & mismanagement
- ✓ The “build it and they will come” mentality

Challenges of a Data Lake: People

Redundant effort

- ✓ Little focus on reusability, consistency, standardization
- ✓ Time-consuming data prep & cleansing efforts which don't add analytical value
- ✓ Effort to operationalize self-service data prep processes
- ✓ Minimal sharing of prepared datasets, calculations, previous findings, & lessons learned among analysts

Expectations

- ✓ High expectations for analysts to conduct their own data preparation, manipulation, integration, cleansing, analysis
- ✓ Skills required to interact with data which is schema-less

Data stewardship

- ✓ Unclear data ownership & stewardship for each subject area or source