aioug
ALL INDIA ORACLE USERS GROUP
THE POWER TO CONNECT & CONQUER

sangam 17
....meeting of minds

Harness the Power of Data in a Big Data Lake

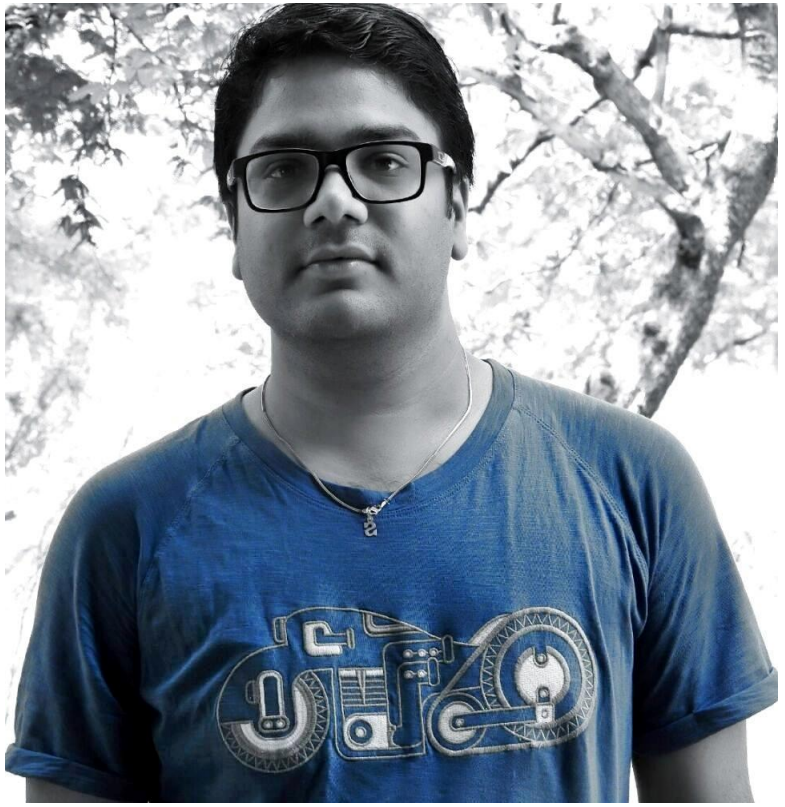
Saurabh K. Gupta
Manager, Data & Analytics, GE
www.amazon.com/author/saurabhgupta
@saurabhkg

Disclaimer:

“This report has been prepared by the Authors who are part of GE. The opinions expresses herein by the Authors and the information contained herein are in good faith and Authors and GE disclaim any liability for the content in the report. The report is the property of GE and GE is the holder of the copyright or any intellectual property over the report. No part of this document may be reproduced in any manner without the written permission of GE. This Report also contains certain information available in public domain, created and maintained by private and public organizations. GE does not control or guarantee the accuracy, relevance, timelines or completeness of such information. This Report constitutes a view as on the date of publication and is subject to change. GE does not warrant or solicit any kind of act or omission based on this Report.”

About me

- Manager, Data & Analytics at GE Digital
- 10+ years of experience in data architecture, engineering, analytics.
- Authored couple of books
  - Oracle Advanced PL/SQL Developer Professional Guide (2012)
  - Advanced Oracle PL/SQL Developer's Guide (2016)
  - *Current session - excerpt from an "untitled book" (TBD)*
- Speaker at AIOUG, IOUG, NASSCOM
- Twitter @saurabhkg, Blog@ sbhoracle.wordpress.com



Why I’m here?

- Data lake is relatively a new term when compared to all fancy ones since the industry realized the potential of data. Industry is planning their way out to adopt big data lake as the key data store but what challenges them is the traditional approach. Traditional approaches pertaining to data pipelines, data processing, data security still hold good but architects do need to leap an extra mile while designing big data lake.

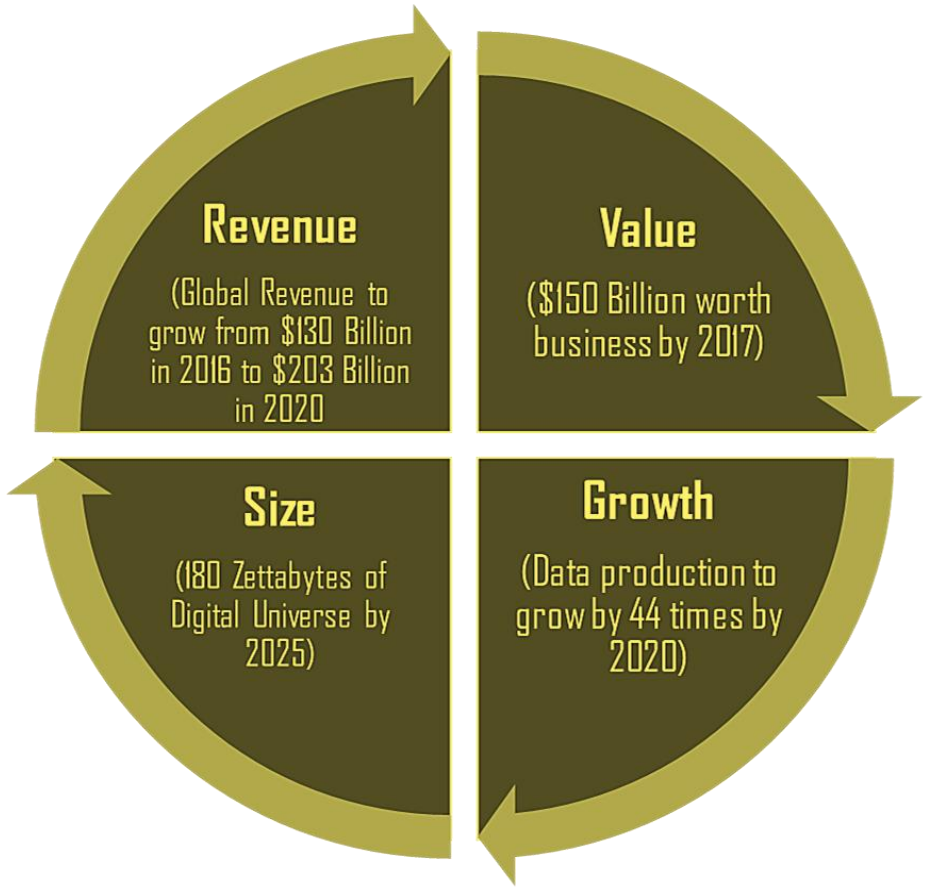
- This session will focus on this shift in approaches. We will explore what are the road blockers while setting up a data lake and how to size the key milestones. Health and efficiency of a data lake largely depends on two factors - data ingestion and data processing. Attend this session to learn key practices of data ingestion under different circumstances. Data processing for variety of scenarios will be covered as well.

Agenda

- Data Lake – the evolution
- Data Lake architecture
- Data Lake ingestion strategy
  - Structured
  - Unstructured
- Change data capture
  - Principles
- Data Processing techniques

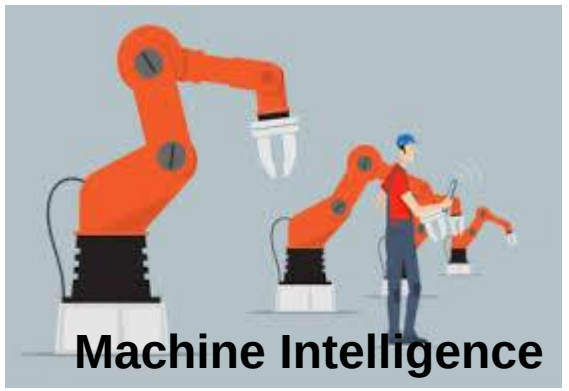
Data Explosion

Future data trends



Data as a Service

DaaS



Evolution of Data Lake

- James Dixon’s “time machine” vision of data
- Leads Data-as-an-Asset strategy

“If you think of a datamart as a store of bottled water – cleansed and packaged and structured for easy consumption – the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples.”

-James Dixon

What is Data Lake? And Why?

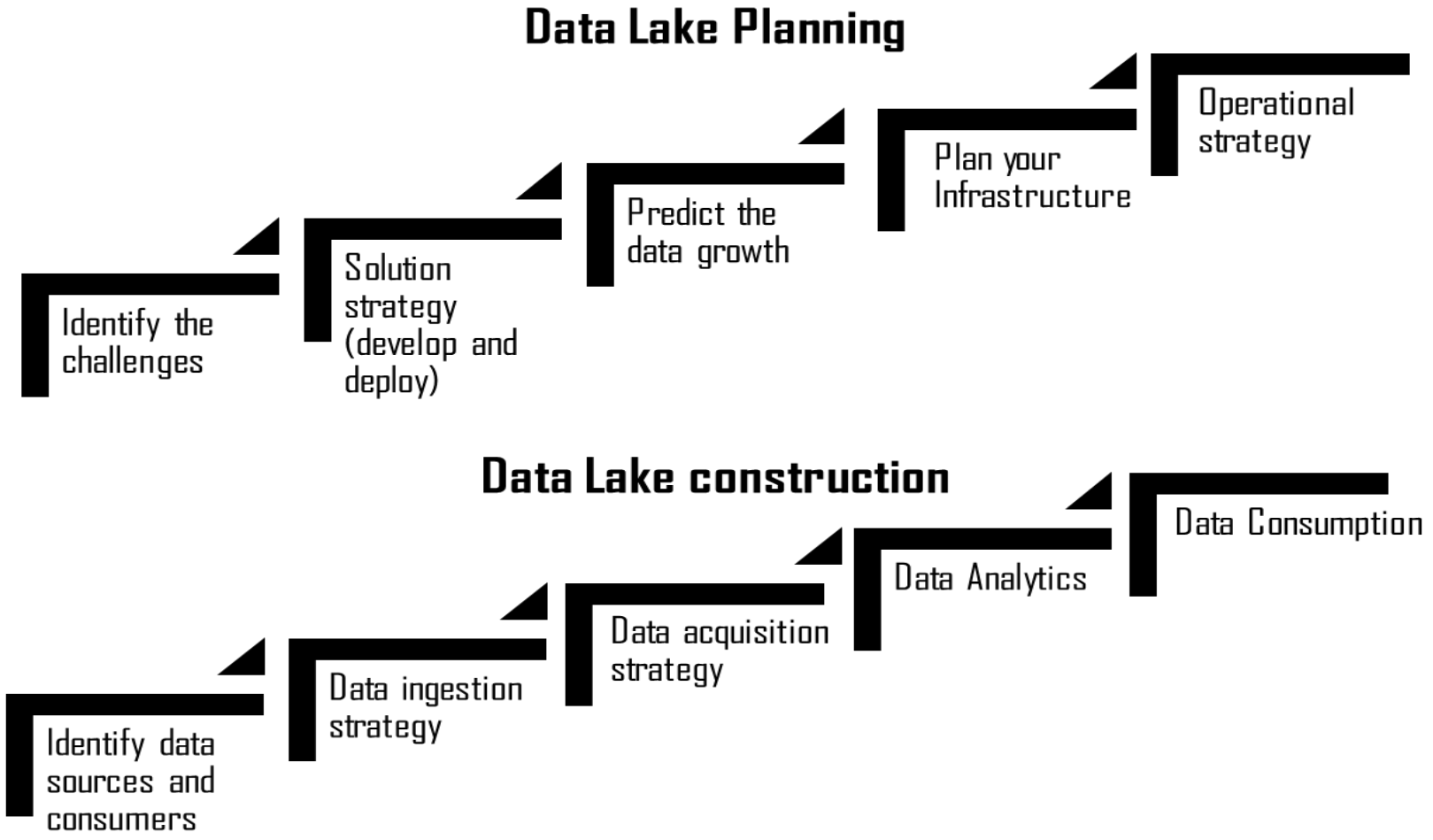
What?

- Data lake represents a state of enterprise at a given time
- Store all data in much detailed fashion at one place
- Empower business analytics applications, predictive and deep learning models with one “time machine” data store
- Unifies data discovery, data science and enterprise BI in an organization

Why?

- Scalable framework to store massive volumes of data and churn out analytical insights
- Data ingestion and processing framework becomes the cornerstone rather than just the storage
- warrants data discovery platforms to soak the data trends at a horizontal scale and produce visual insights

Data lake – constructing strategy



Data lake vs Data warehouse

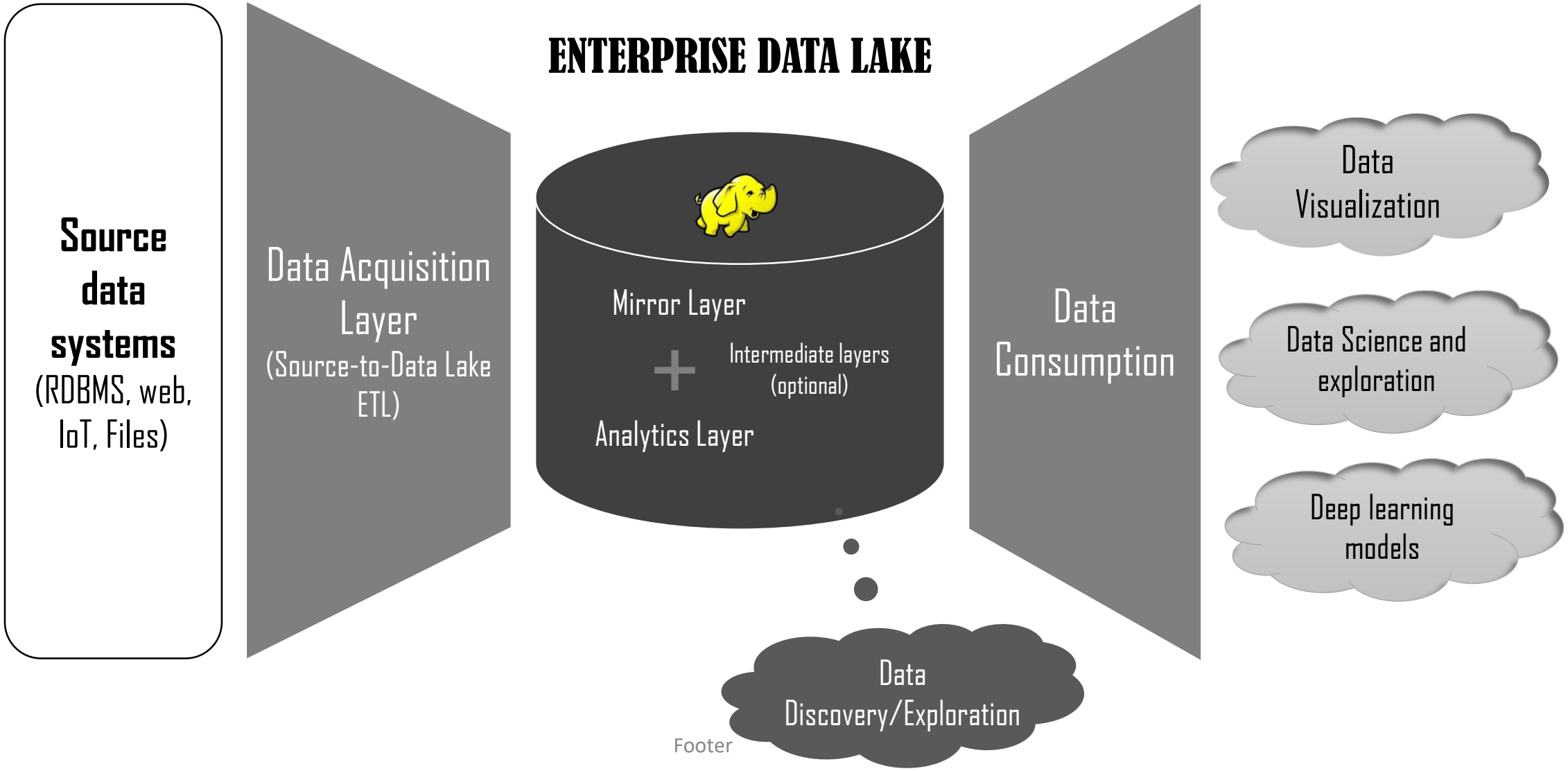
Data Lake

- Data lake gets the data in its raw format, unprocessed and untouched to build mirror layer. Follows **schema on-read** strategy
- War room for data scientists, analysts and data engineering specialists from multiple domains to run analytical models.
- Data lake is primarily hosted on Hadoop, which is open source and comes with free community support

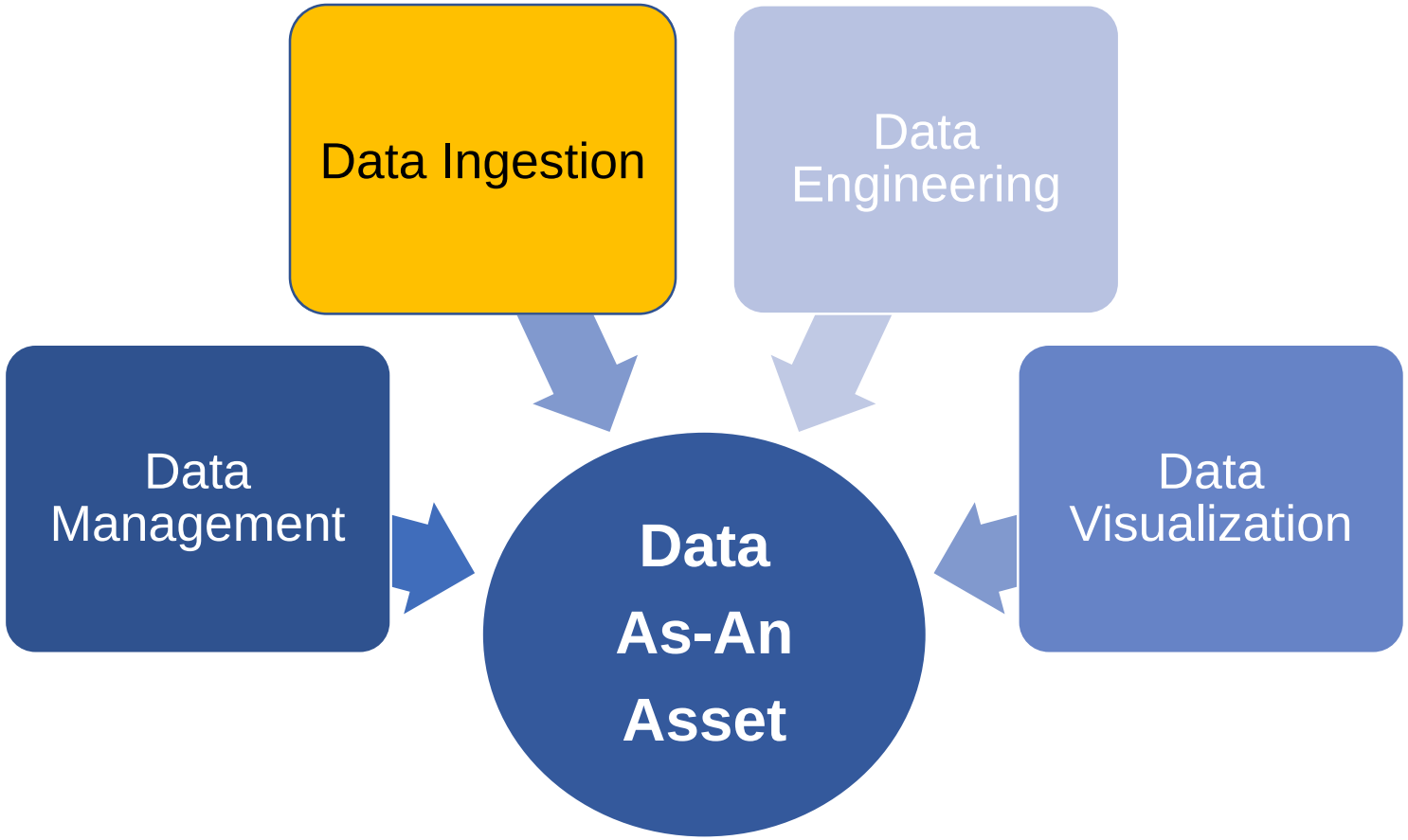
Data Warehouse

- Data warehouse pulls data from the source and takes through a processing layer i.e. **schema on-write**
- Data warehouse targets management staff and business leaders who expect structured analyst reports end of the day
- Data warehouse, runs on relatively expensive storage to withstand high-scale data processing.

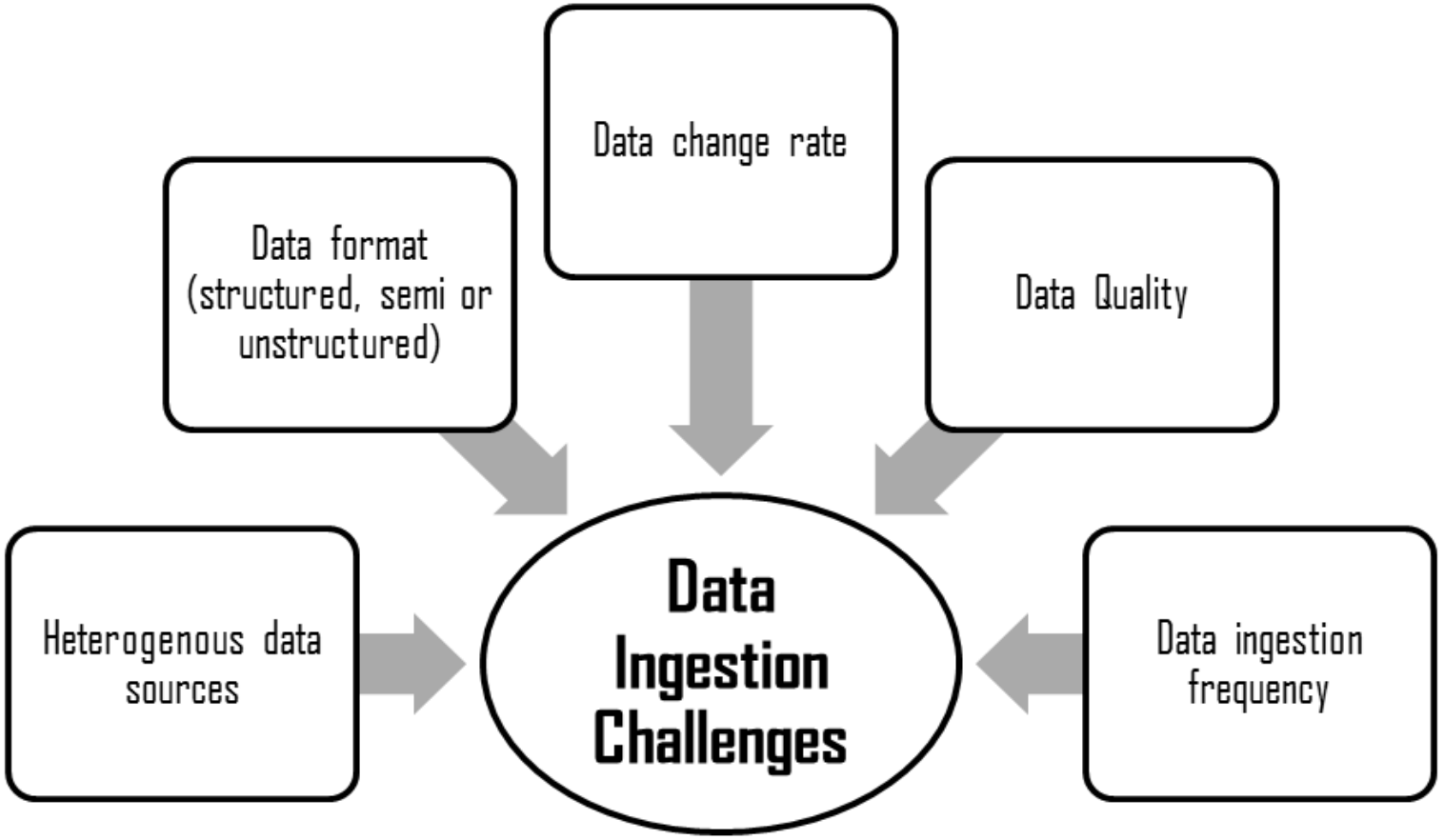
Data Lake architecture



Data Lake operational pillars

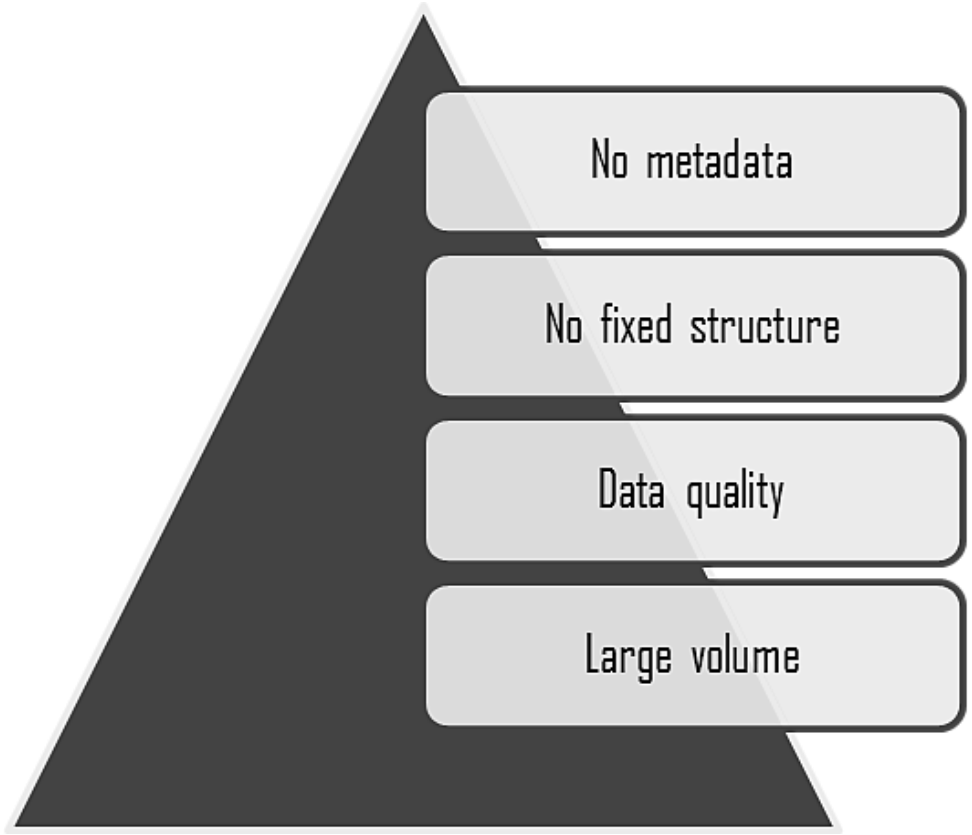


Data Ingestion challenges



Understand the data

- **Structured data** is an organized piece of information
  - Aligns strongly with the relational standards
  - Defined metadata
  - Easy ingestion, retrieval, and processing
- **Unstructured data** lacks structure and metadata
  - Not so easy to ingest
  - Complex retrieval
  - Complex processing



Understand the data sources

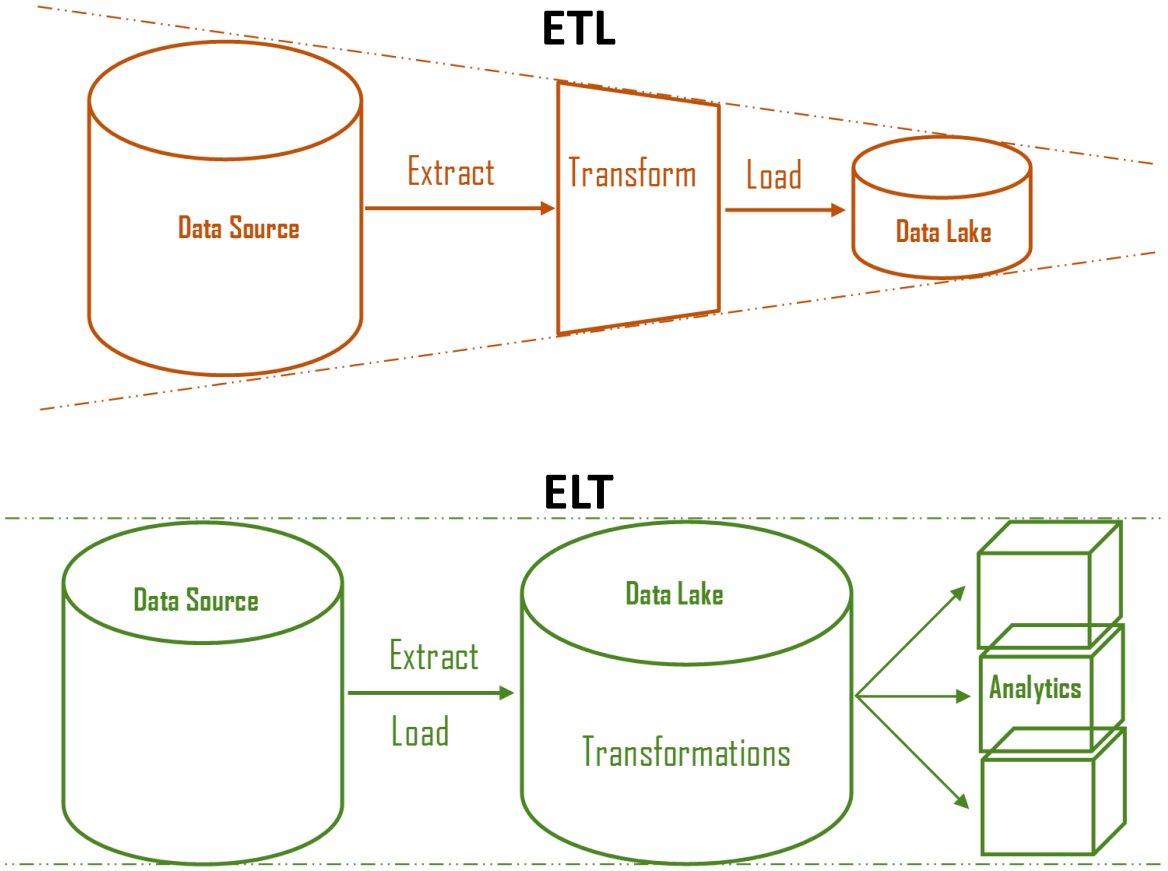
- OLTP and Data warehouses – structured data from typical relational data stores.
- Data management systems – documents and text files
- Legacy systems – essential for historical and regulatory analytics
- Sensors and IoT devices – Devices installed on healthcare, home, and mobile appliances and large machines can upload logs to data lake at periodic intervals or in a secure network region
- Web content – data from the web world (retail sites, blogs, social media)
- Geographical data – data flowing from location data, maps, and geo-positioning systems.

Data Ingestion Framework
Design Considerations

- Data format – What format is the data to be ingested?
- Data change rate – Critical for CDC design and streaming data. Performance is a derivative of throughput and latency.
- Data location and security –
  - Whether data is located on-premise or public cloud infrastructure. While fetching data from cloud instances, network bandwidth plays an important role.
  - If the data source is enclosed within a security layer, ingestion framework should be enabled establishment of a secure tunnel to collect data for ingestion
- Transfer data size (file compression and file splitting) – what would be the average and maximum size of block or object in a single ingestion operation?
- Target file format – Data from a source system needs to be ingested in a hadoop compatible file format.

ETL vs ELT for Data Lake

- Heavy transformation may restrict data surface area for data exploration
  - Brings down the data agility
- Transformation on huge volumes of data may foster a latency between data source and data lake
- Curated layer to empower analytical models



Batched data ingestion principles

Structured data

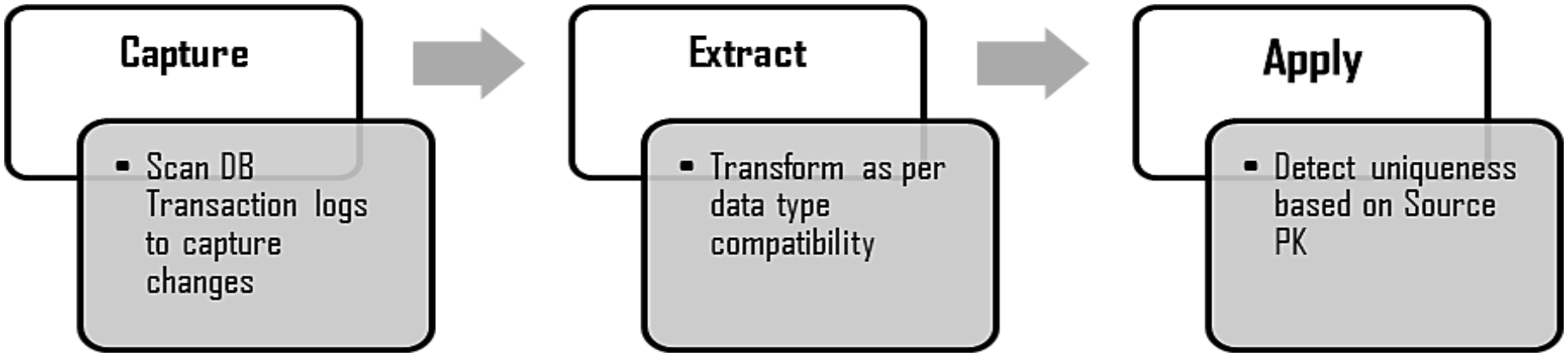
- Data collector fires a SELECT query (also known as *filter query*) on the source to pull incremental records or full extract
- Query performance and source workload determine how efficient data collector is
- Robust and flexible

Ingestion techniques

- Change Track flag – flag rows with the operation code
- Incremental extraction – pull all the changes after certain timestamp
- Full Extraction – refresh target on every ingestion run

Change Data Capture

- Log mining process
- **Capture** changed data from the source system's transaction logs and **integrate** with the target
- Eliminating the need to run SQL queries on source system. Incurs no load overhead on a transactional source system.
- Achieves near real-time replication between source and target



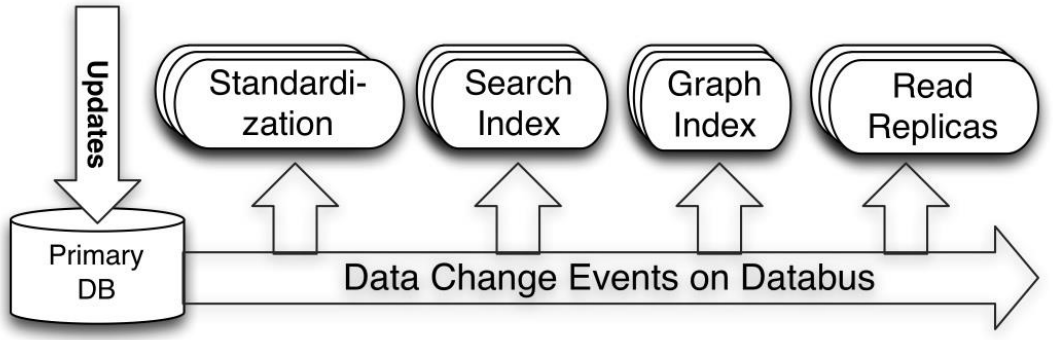
Change Data Capture
Design Considerations

- Source database be enabled for logging
- Commercial tools - Oracle GoldenGate, HVR, Talend CDC, custom replicators
- Keys are extremely important for replication
  - Helps capture job in establishing uniqueness of a record in the changed data set
  - Source PK ensures the changes are applied to the correct record on target
  - PK not available; establish uniqueness based on composite columns
  - Establish uniqueness based on a *unique* constraint - terrible design!!
- Trigger based CDC
  - Event on a table triggers the change to be captured in an change-log table
  - Change-log table merged with the target
  - Works when source transaction logs are not available

LinkedIn Databus

CDC capture pipeline

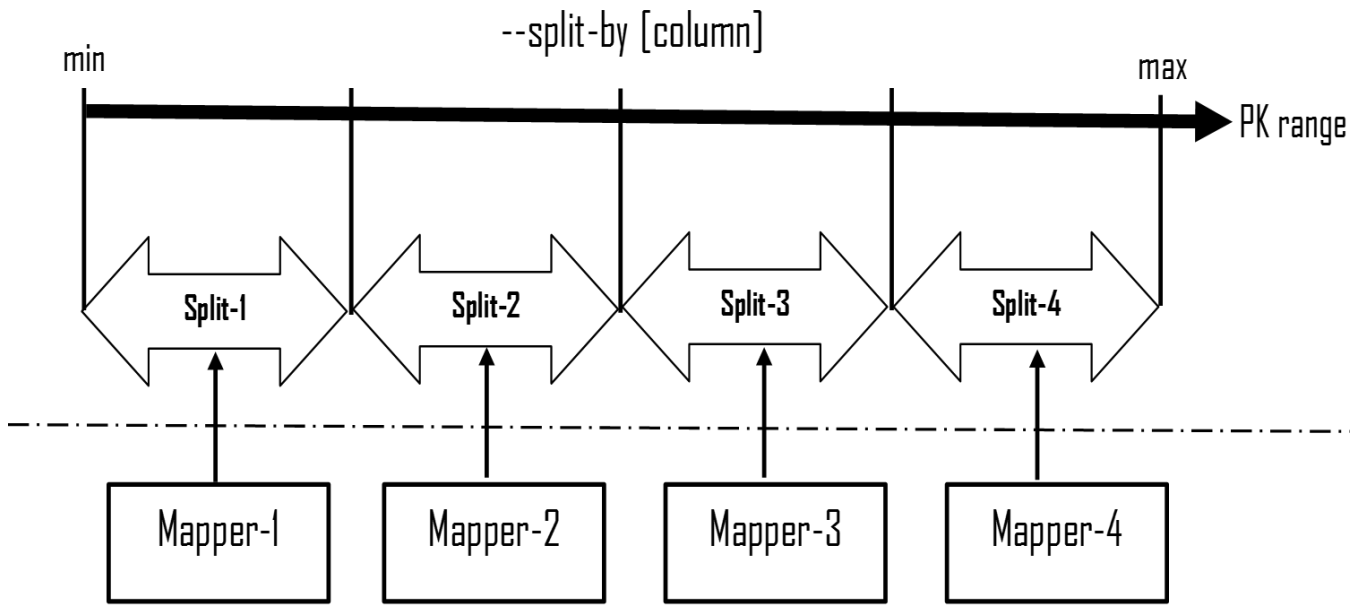
- Relay is responsible for pulling the most recent committed transactions from the source
  - Relays are implemented through tungsten replicator
- Relay stores the changes in logs or cache in compressed format
- Consumer pulls the changes from relay
- Bootstrap component – a snapshot of data source on a temporary instance. It is consistent with the changes captured by Relay
  - If any consumer falls behind and can't find the changes in relay, bootstrap component transforms and packages the changes to the consumer
  - A new consumer, with the help of client library, can apply all the changes from bootstrap component until a time. Client library will point the consumer to Relay to continue pulling most recent changes



Apache Sqoop

- Native member of Hadoop tech stack for data ingestion
- Batched ingestion, no CDC
- Java based utility (web interface in Sqoop2) that spawns Map jobs from MapReduce engine to store data in HDFS
- Provides full extract as well as incremental import mode support
- Runs on HDFS cluster and can populate tables in Hive, HBase
- Can establish a data integration layer between NoSQL and HDFS
- Can be integrated with Oozie to schedule import/export tasks
- Supports connectors to multiple relational databases like Oracle, SQL Server, MySQL

Sqoop architecture



- Mapper jobs of MapReduce processing layer in Hadoop
- By default, a sqoop job has four mappers
- Rule of Split
  - Values of `--split-by` column must be equally distributed to each mapper
  - `--split-by` column must be a primary key
  - `--split-by` column should be a primary key

Sqoop

Design considerations - I

- Mappers
  - `--num-mappers` [n] argument
  - run in parallel within Hadoop. No formula but needs to be judiciously set
- Cannot split
  - `--autoreset-to-one-mapper` to perform unsplit extraction
- Source has no PK
  - Split based on natural or surrogate key
- Source has character keys
  - Divide and conquer! Manual partitions and run one mapper per partition
  - If key value is an integer, no worries

Sqoop

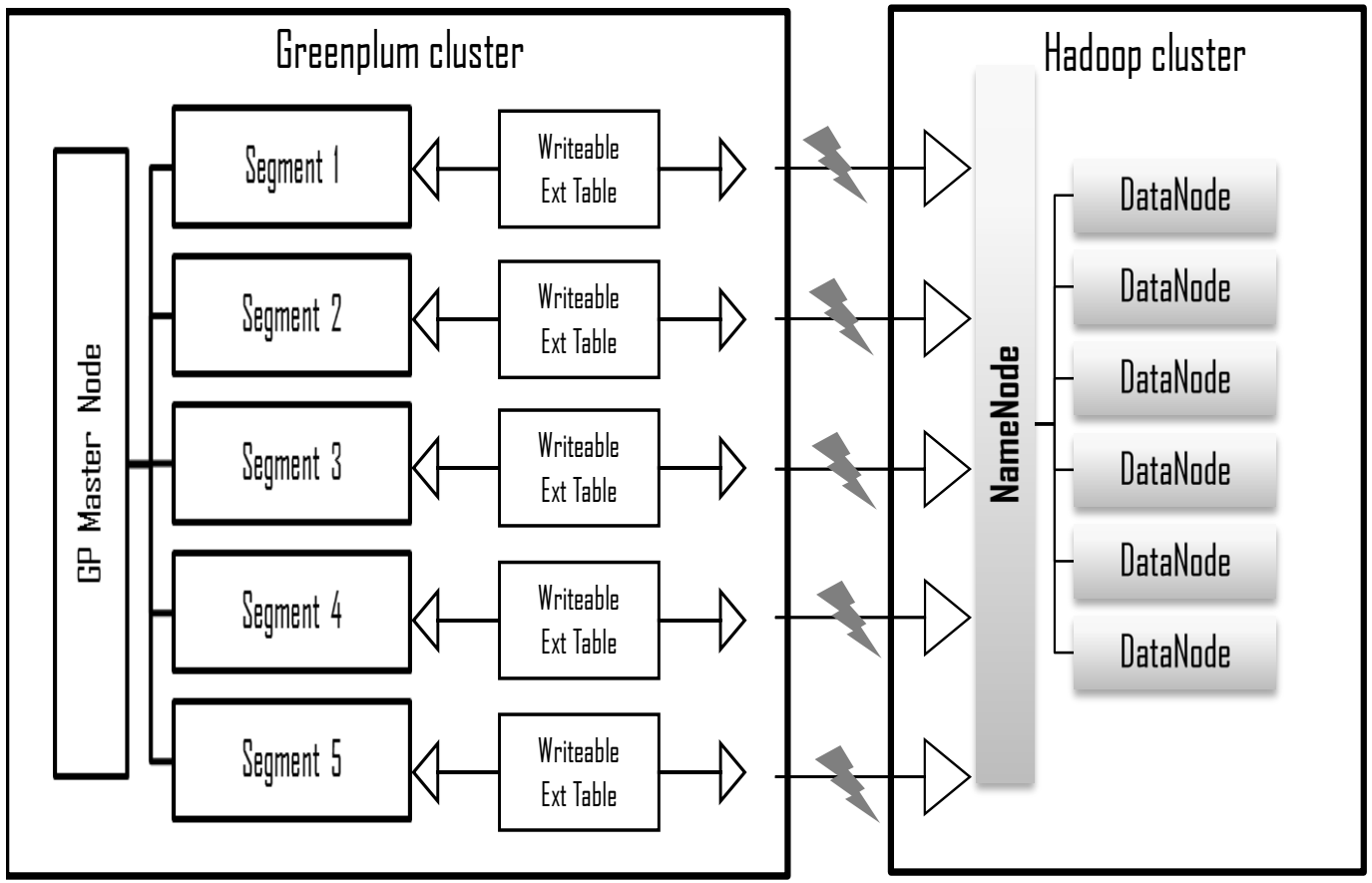
Design considerations - II

- If only subset of columns is required from the source table, specify column list in --columns argument.
  - For example, --columns “orderId, product, sales”
- If limited rows are required to be “sqooped”, specify --where clause with the predicate clause.
  - For example, --where “sales > 1000”
- If result of a structured query needs to be imported, use --query clause.
  - For example, --query ‘select orderId, product, sales from orders where sales>1000’
- Use `--hive-partition-key` and `--hive-partition-value` attributes to create partitions on a column key from the import
- Delimiters can be handled through either of the below ways –
  - Specify `--hive-drop-import-delims` to remove delimiters during import process
  - Specify `--hive-delims-replacement` to replace delimiters with an alternate character

Oracle copyToBDA

- Licensed under Oracle BigData SQL
  - Stack of Oracle BDA, Exadata, Infiniband
- Helps in loading Oracle database tables to Hadoop by –
  - Dumping the table data in Data Pump format
  - Copying them into HDFS
- Full extract and load
- Source data changes
  - Rerun the utility to refresh Hive tables

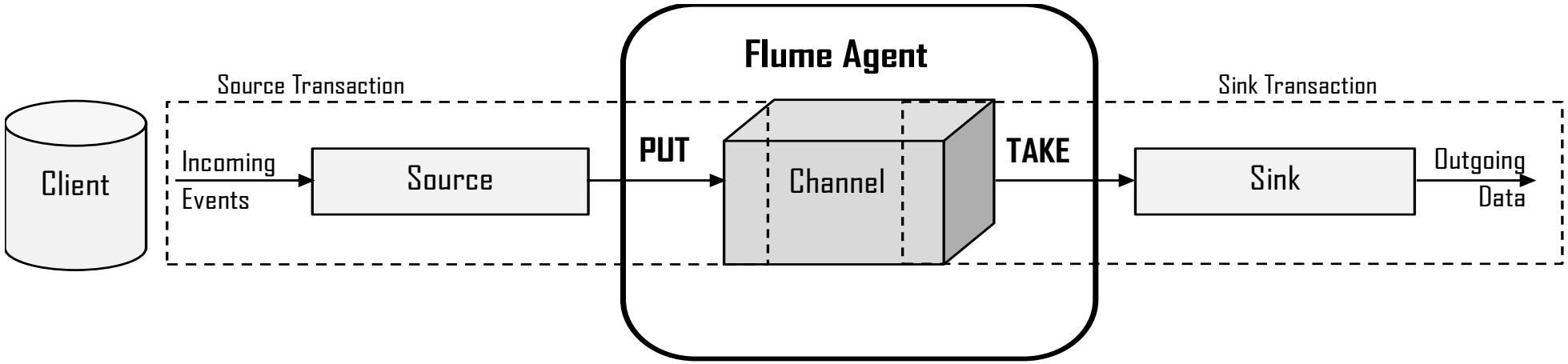
Greenplum's GPHDFS



- Setup on all segment nodes of a Greenplum cluster
- All segments concurrently push the local copies of data splits to Hadoop cluster
- Cluster segments yield the power of parallelism

Ingest unstructured data using Flume

- Distributed system to capture and load large volumes of log data from different source systems to data lake
- Collection and aggregation of streaming data as events

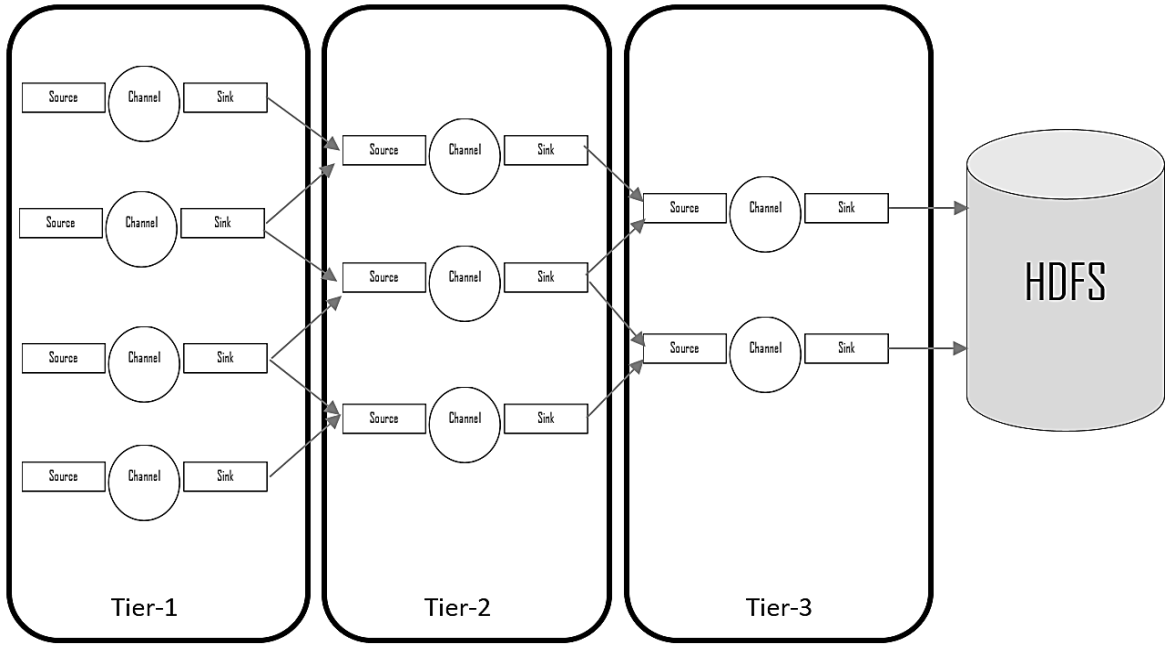


Apache Flume
Design considerations - I

- Channel type
  - MEMORY - events are read from source to memory
    - Good performance, but volatile. Not cost effective
  - FILE – events are ready from source into file system
    - Controllable performance. Persistent and Transactional guarantee
  - JDBC - events are read and stored in Derby database
    - Slow performance
  - Kafka – store events in Kafka topic
- Event batch size - maximum number of events that can be batched by source or sink in a single transaction
  - Fatter the better. But not for FILE channel
  - Stable number ensures data consistency

Apache Flume
Design considerations - II

- Channel capacity and transaction capacity
  - For MEMORY channel, channel capacity is limited by RAM size.
  - For FILE, channel capacity is limited by disk size
  - Should not exceed batch size configured for the sinks
- Channel selector
  - An event can either be replicated or multiplexed
  - Preferable vs conditional
- Handle high throughput systems
  - Tiered architecture to handle event flow
  - Aggregate and push approach



Questions?

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
go raibh maith agat
mochchakkeram
merci
obrigado
bedankt
dziękuję
sukriya
kop khun krap
terima kasih
спасибо
Баярлалаа
рахмат
dank je
tapadh leat
arigatô
dakujem
grazie
takk
sagolun
hvala
mauruuru
taiku
감사합니다
ありがとう
ευχαριστώ
мерси
хвала
дякую
спасибі
mersi
kia ora
barka
welalin
tack
spas
faafetai lava
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
kiitos
dankie
nandri
nanni
dhanyavad
bayarlalaa
gràçie
köszönöm
enkosi
akun
dankon
aciu
asante
manana
obrigada
tenki
murakoze
chokrane
djiere dieuf
tau
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpáy
sobodi
děkuji
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
tanemirt
rahmet
dhanyavadagalu
shukriya
mercé
trugarez
diolch
xièxie
তোমাকে ধন্যবাদ