UNIT-1
DATA SCIENCE TECHNOLOGY STACK


mlr
OpenAI
Keras
spaCy
OpenCV
TensorFlow
statsmodels
learn
PyTorch
plotly
SciPy
KNIME
sas
NumPy
matplotlib
Seaborn
Scala
alteryx
Java
Studio
jupyter
pandas
kafka
Spark
GitHub
DASK
git
python
MySQL
SQLAlchemy
Flask
Python
node
ORACLE
DATABASE
cassandra
PREFECT
PostgreSQL
GO
JS
Express
APOLLO
elastic
mongoDB
TS
presto
SVELTE

WHAT IS DATA SCIENCE TECH STACK?

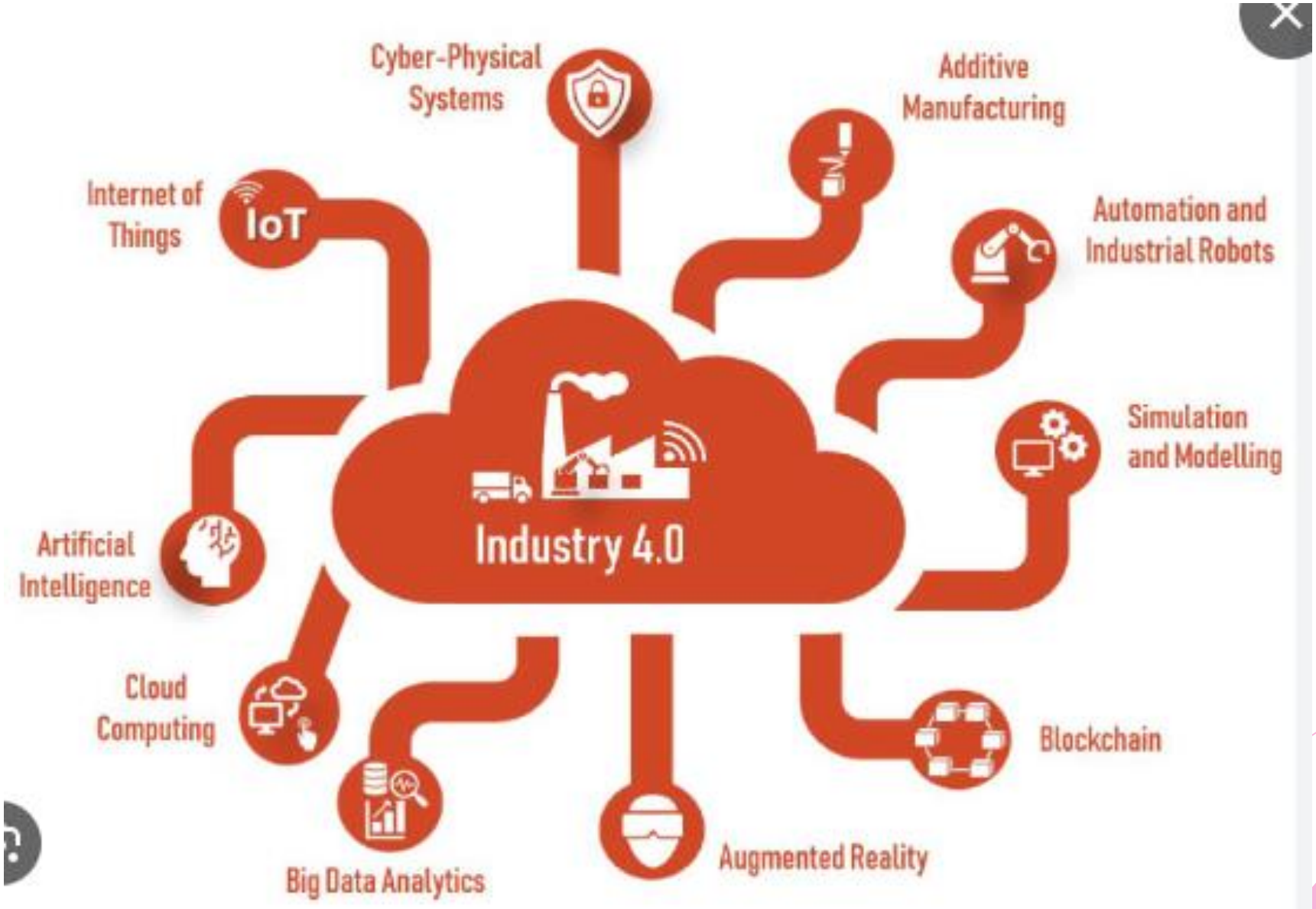
- The data science tech stack is not only about the framework used to create models or the runtime for inference jobs. It extends to your complete data engineering pipeline, business intelligence tools, and the way in which models are deployed.

RAPID INFORMATION FACTORY ECOSYSTEM

- Rapid Information Factory (RIF) System is a technique and tool which is used for processing the data in the development.

- The Rapid Information Factory is a massive parallel data processing platform capable of processing theoretical unlimited size data sets.

SMART FACTORY BASED ON CYBER SYSTEMS AND IOT



LAYERS OF RAPID INFORMATION FACTORY ECOSYSTEM

- Functional Layer
- Operational Management Layer
- Audit, Balance and Control layer
- Utility Layer
- Business Layer

FUNCTIONAL LAYER

- The functional layer is the core processing capability of the factory.
- Here we use R-A-P-T-O-R framework.
- RAPTOR stands for Rapid Algorithmic Prototyping Tool For Ordered Reasoning.
- Common components supporting other layers.
- o Maintenance Utilities.
- o Data Utilities.
- o Processing Utilities.

BUSINESS LAYER

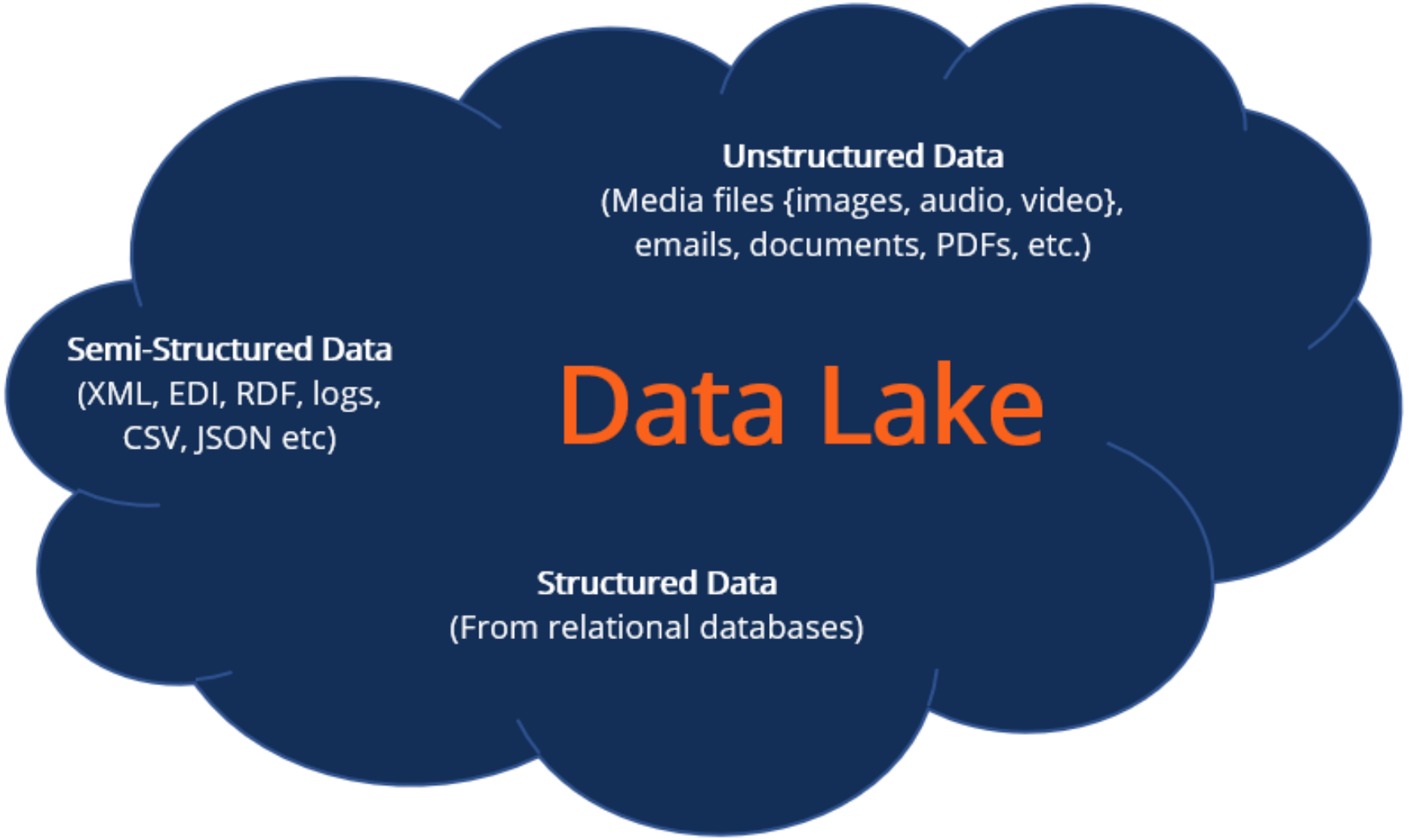
- Contains the business requirements (Functional and Non-functional).

DATA SCIENCE STORAGE TOOLS

- Data Science ecosystem has a bunch of series of tools which are used to build your solution. By using this tools and techniques you will get rapid information in advanced for its better capability and new development will occur each day.

DATA LAKE

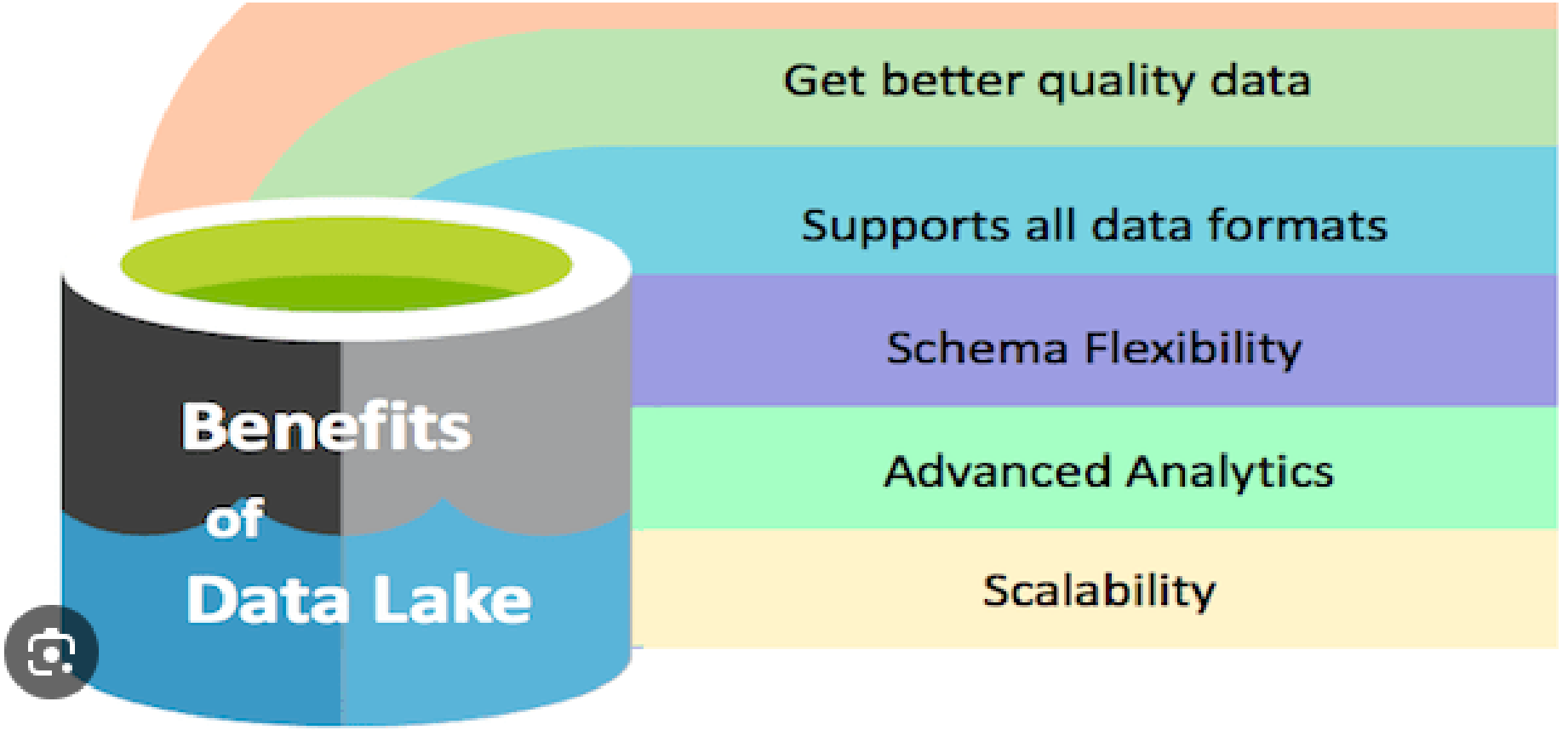
- A Data Lake is storage repository of large amount of raw data that means structure, semi-structure, unstructured data.


Unstructured Data
(Media files {images, audio, video},
emails, documents, PDFs, etc.)
Semi-Structured Data
(XML, EDI, RDF, logs,
CSV, JSON etc)
Data Lake
Structured Data
(From relational databases)

Contt..

- This is the place where you can store three types of data structure, semi-structure, unstructured data with no fix amount of limit and storage to store the data.

- Data Lake allow us to transform the raw data that means structure, semi-structure, unstructured data into the structure data format so that SQL query could be performed for the analysis.


Benefits
of
Data Lake
Get better quality data
Supports all data formats
Schema Flexibility
Advanced Analytics
Scalability

DATA VAULT

- A data vault is a system made up of a model, methodology and architecture that is specifically designed to solve a complete business problem as requirements change.

DATA VAULT

- Data vault is a customized, dynamic solution that gives business users access to all data (current and historical).

DATA VAULT USE CASE

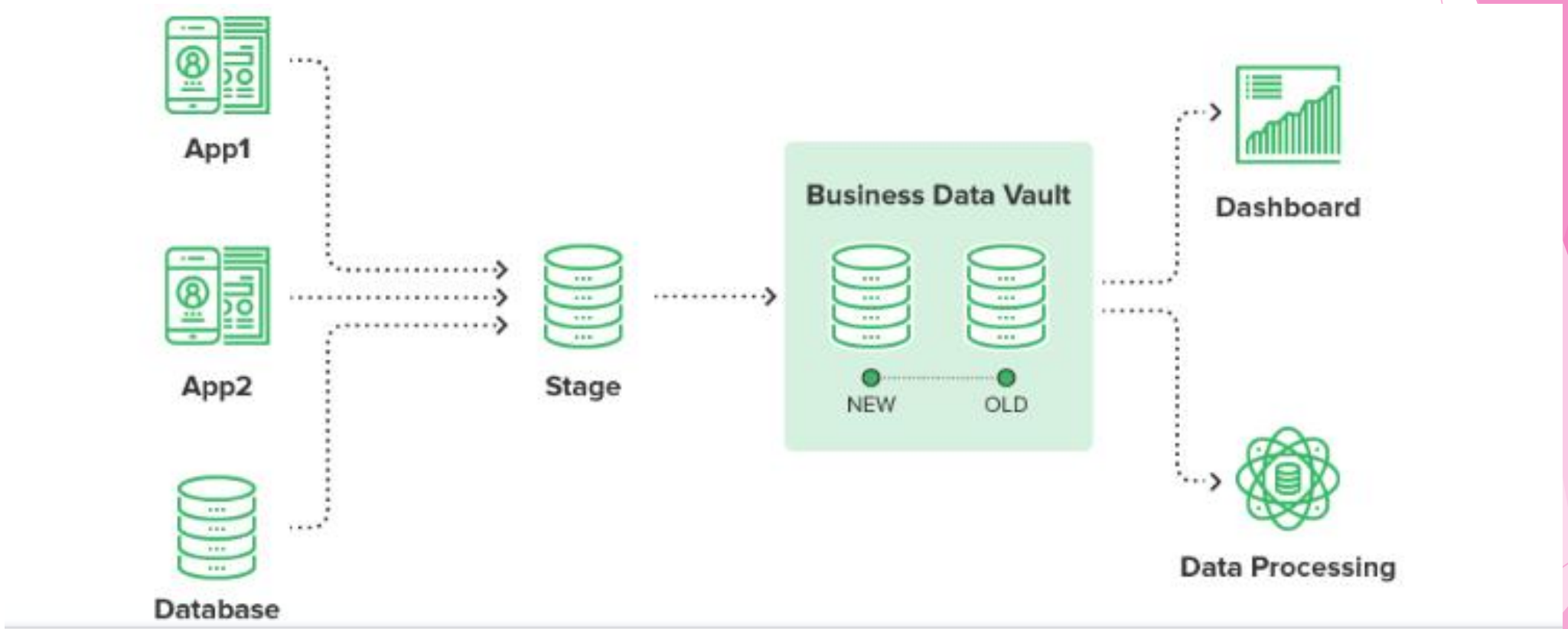
- The biggest data vault use case is when a business, such a bank, needs to audit their data.

DATA VAULT USE CASE

- If you decide you need to update your security model to include additional fields and new applications in your enterprise.

DATA VAULT USE CASE

- Using a data vault, you are able to checkpoint the time you made the security model changes and update your infrastructure with the changes, including all associated applications.


App1
App2
Database
Stage
Business Data Vault
NEW
OLD
Dashboard
Data Processing

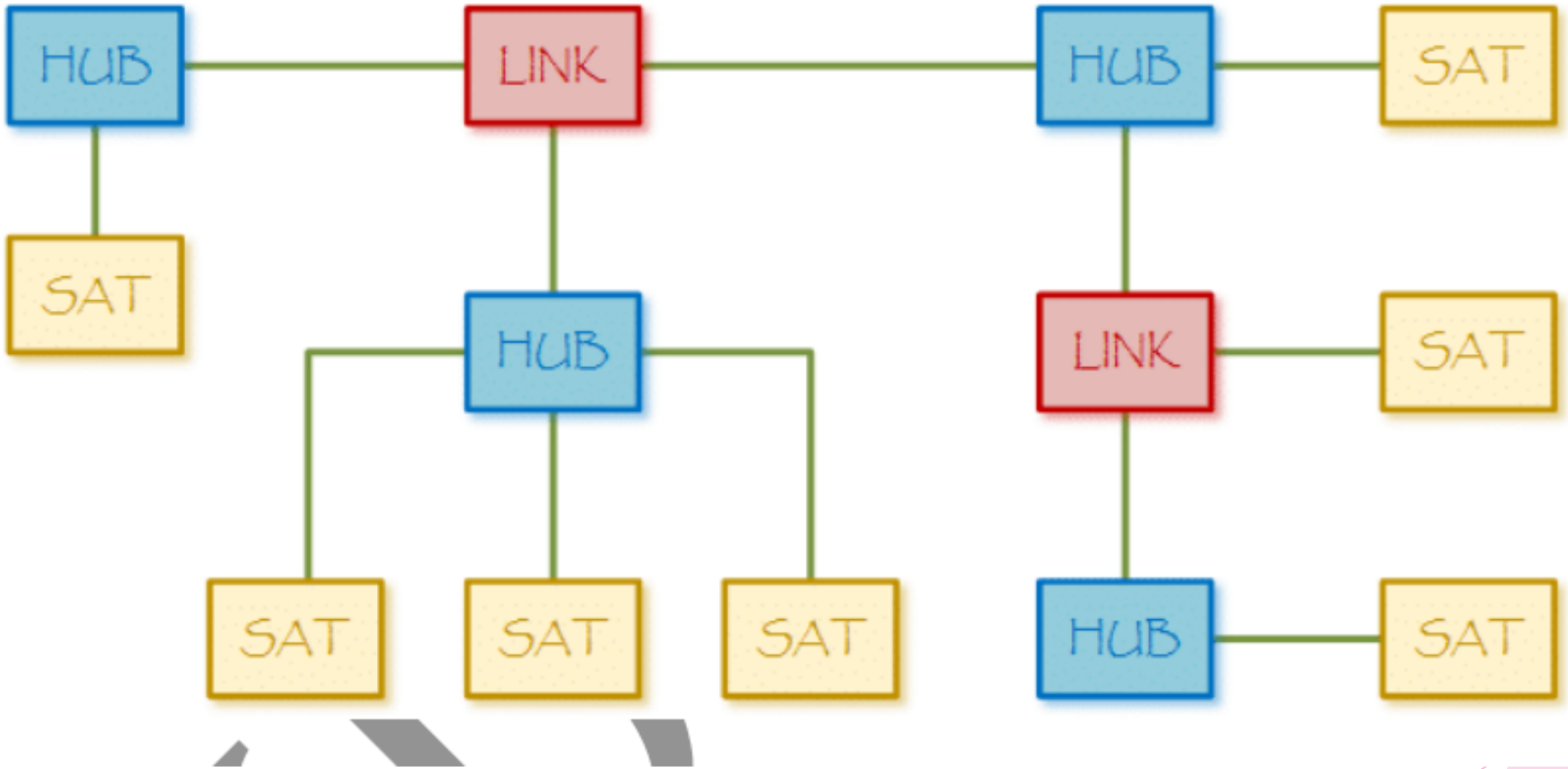
SAMPLE TECHNOLOGIES USED

- REDSHIFT
- RDBMS
- SNOWFLAKE

HUB,LINK AND SATELLITE

- **Hubs** represent core business concepts such as they represent Customer Id/Product Number/Vehicle identification number (VIN). Users will use a business key to get information about a Hub.

- **Links** represent relationships between hubs, and **satellites** store information about hubs and relationships between them.


HUB
LINK
HUB
SAT
SAT
HUB
LINK
SAT
SAT
SAT
SAT
HUB
SAT

Contt..

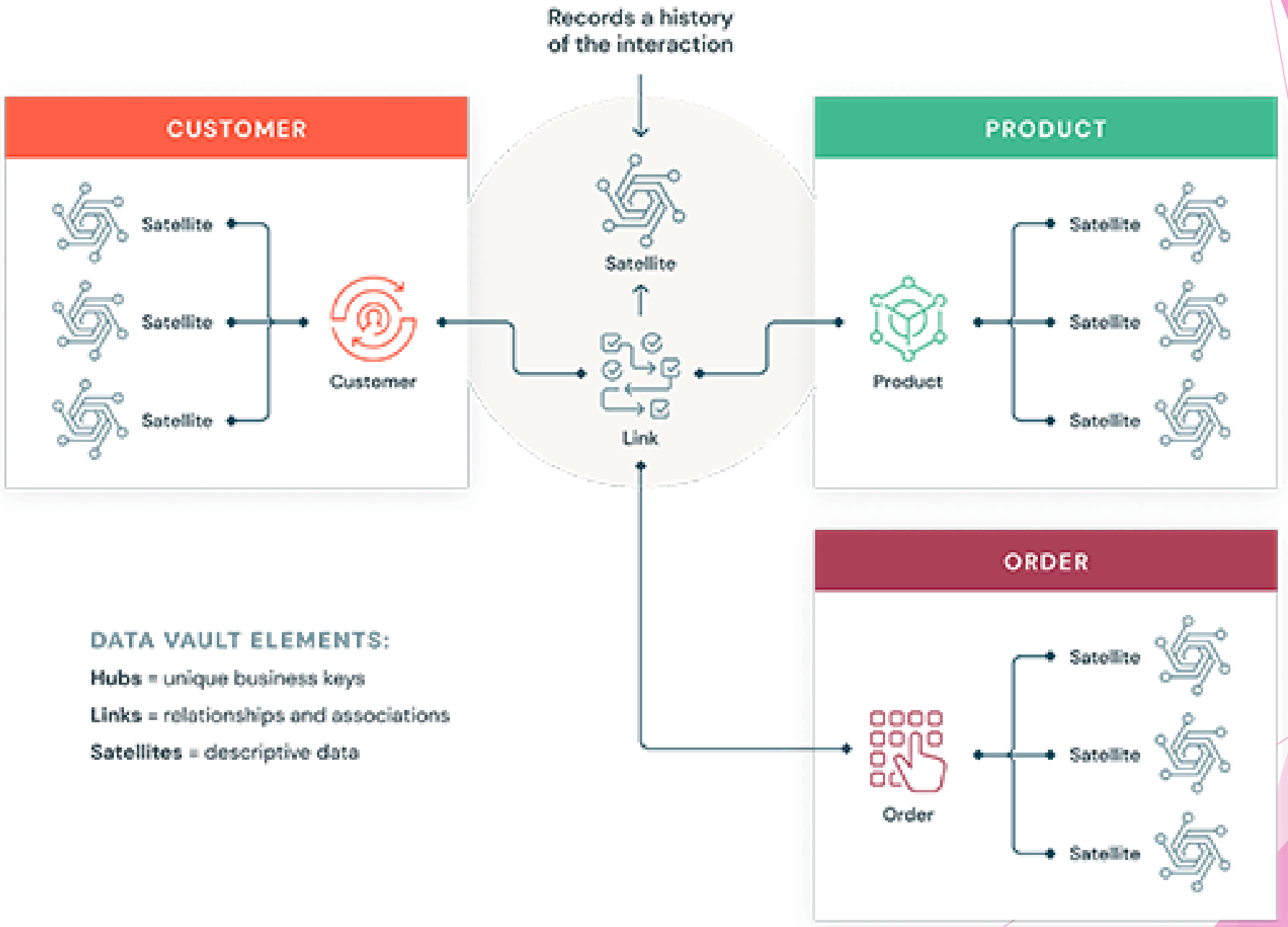
- The combinations of all these three i.e. hub, link, and satellites are formed together to help the data analytics and data scientists and data engineer to store the business structure, types of information or data into it.

HUB,LINK AND SATELLITE

- Satellites fill the gap in answering the missing descriptive information on core business concepts.

- A satellite cannot have a direct connection to another satellite.
- A hub or link may have one or more satellites.

Data vault modeling



DATA VAULT ELEMENTS:

Hubs = unique business keys

Links = relationships and associations

Satellites = descriptive data

DATAWAREHOUSE BUS MATRIX

- One key item Kimball provides in the Requirements Phase is a Bus Matrix.
- It provides an incremental and integrated approach to data warehouse design.
- It is a guide to the logical design phase, and a mechanism to communicate the data in the overall architecture back to the business.

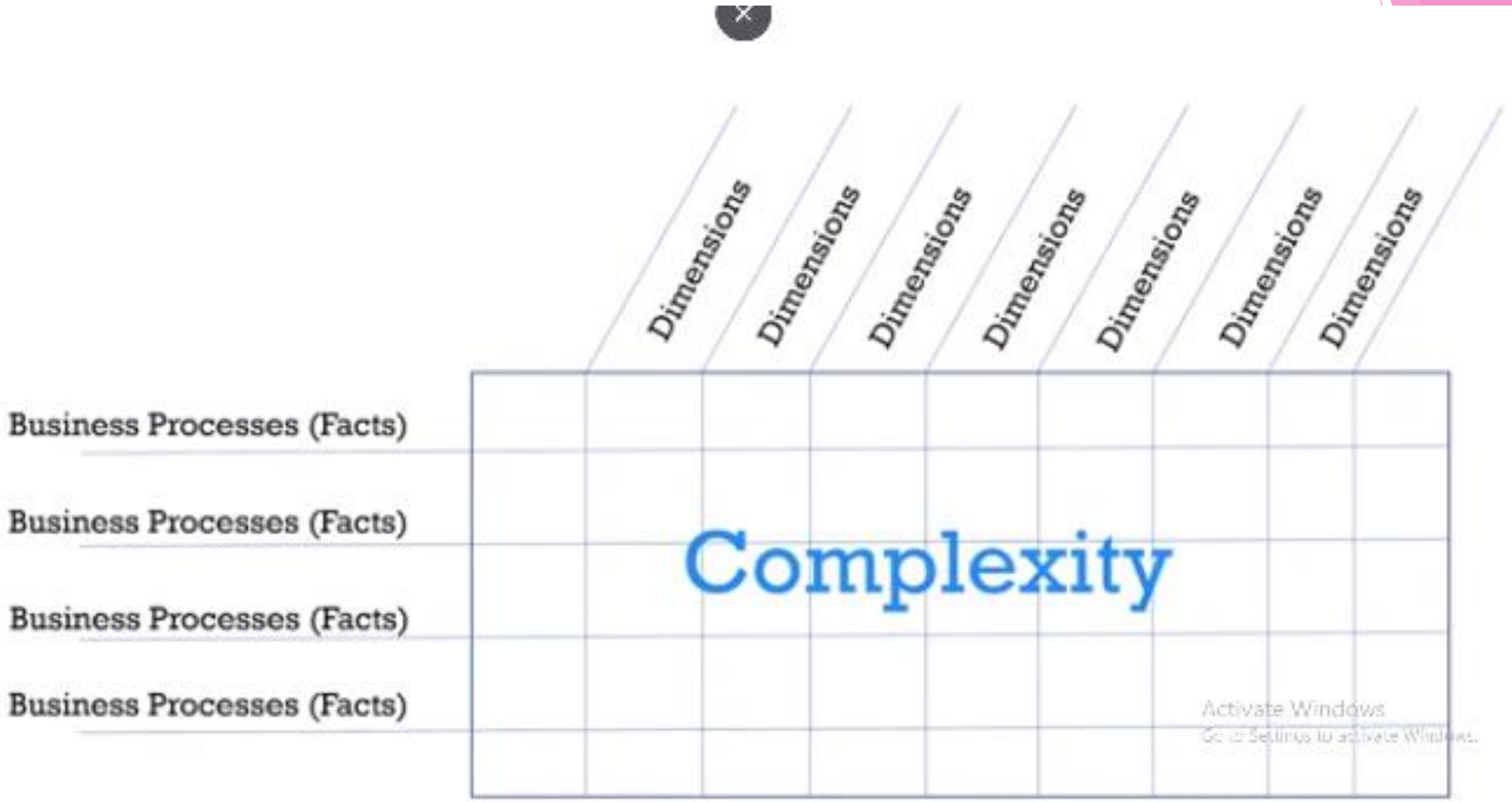
DATAWAREHOUSE BUS MATRIX

- Before moving onto the actual designing of the data warehouse, there is need to create a bus matrix through the collaborative efforts of business analysts team as well as the design team.

- In this way , the bus matrix can feed the business community like what exactly the implementation would look like and what kind of questions will be answered.

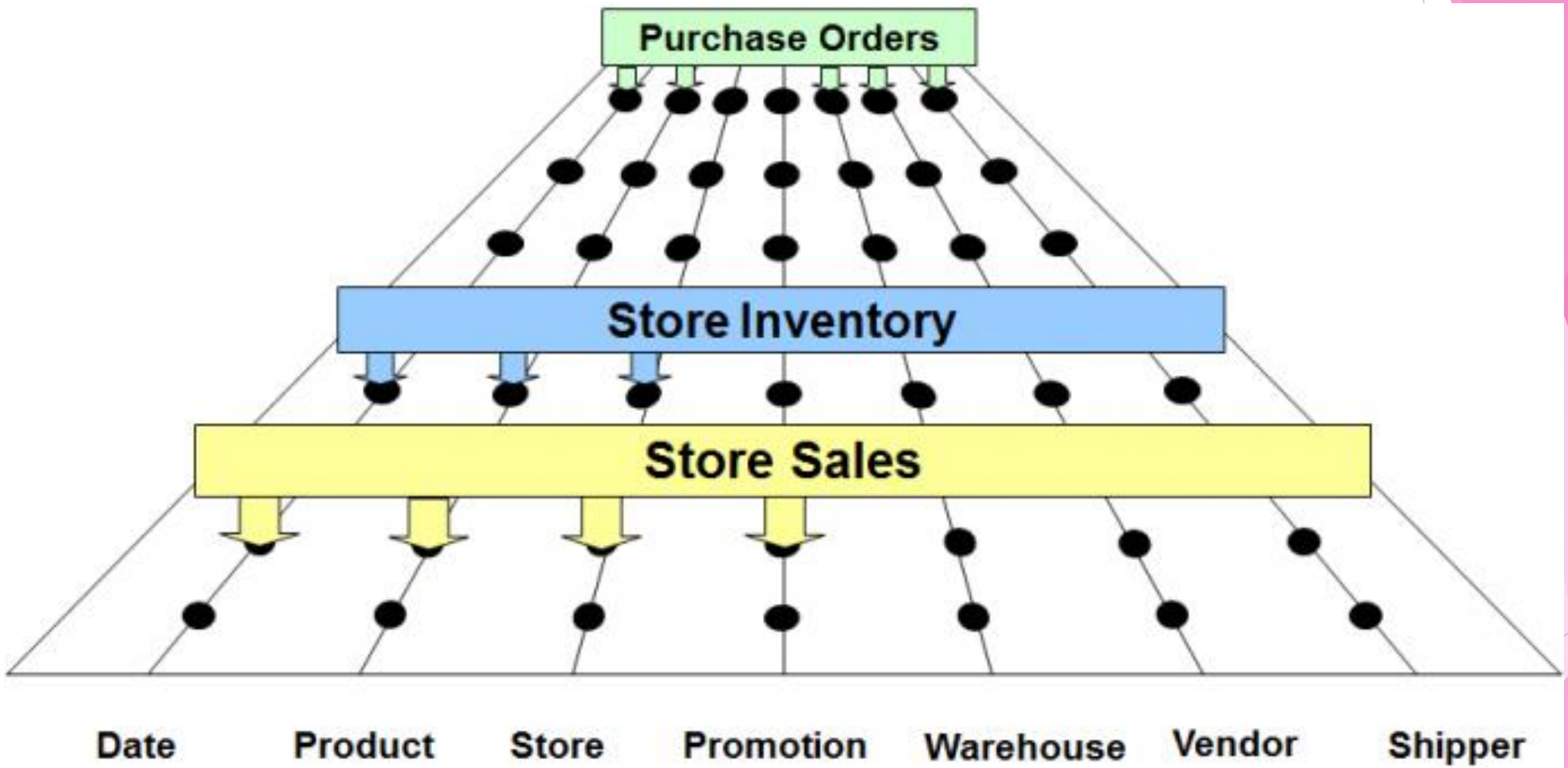
DATAWAREHOUSE BUS MATRIX

- It should be the first thing a Data Warehouse Architect does to define a Data Warehouse project.


Dimensions
Dimensions
Dimensions
Dimensions
Dimensions
Dimensions
Dimensions
Dimensions
Business Processes (Facts)
Business Processes (Facts)
Business Processes (Facts)
Business Processes (Facts)
Complexity

SECTIONS OF THE BUS MATRIX

- There are three sections.
- The rows of the bus matrix showcase the business processes on the left hand side of the bus matrix.
- On the top , the column shows the dimensions .
- In the center, is your complexity matrix.


Purchase Orders
Store Inventory
Store Sales
Date
Product
Store
Promotion
Warehouse
Vendor
Shipper

Conformed Dimensions

- Conformed dimensions deliver consistent descriptive attributes across dimensional models. They support the ability to drill across and integrate data from multiple business processes.

Facts	Calender	Staff	Task	Service	Client	Scheduled Calendar (Role Play)	Started Calendar (Role Play)	Completed Calendar (Role Play)	Billed Calendar (Role Play)	Dimension Usage Count	Complexity
Time Record	**1**	**1**	**1**							3	C
Service	**1**			**1**	**1**	**1**	**1**	**1**	**1**	7	C
Employee Monthly FTE	**1**									1	C
Fact Usage Count	3	1	1	1	1	1	1	1	1		
Complexity	M	M	M	M	M	M	S	S	S		

A “1” in the cell at the intersection of a Fact row and Dimension column indicates that the Fact is associated to that dimension (and therefore the Fact can be analysed by attributes of the Dimension).

By completing a matrix, its possible to see the overall **high level design** of your Data Warehouse on a single sheet.

DATA SCIENCE PROCESSING TOOLS

- Data science tools are used for diving into raw and complicated data (unstructured or structured data) and processing, extracting, and analyzing it to dig out valuable insights by applying different data processing techniques such as statistics, computer science, predictive modeling and analysis, and deep learning.

Contt…

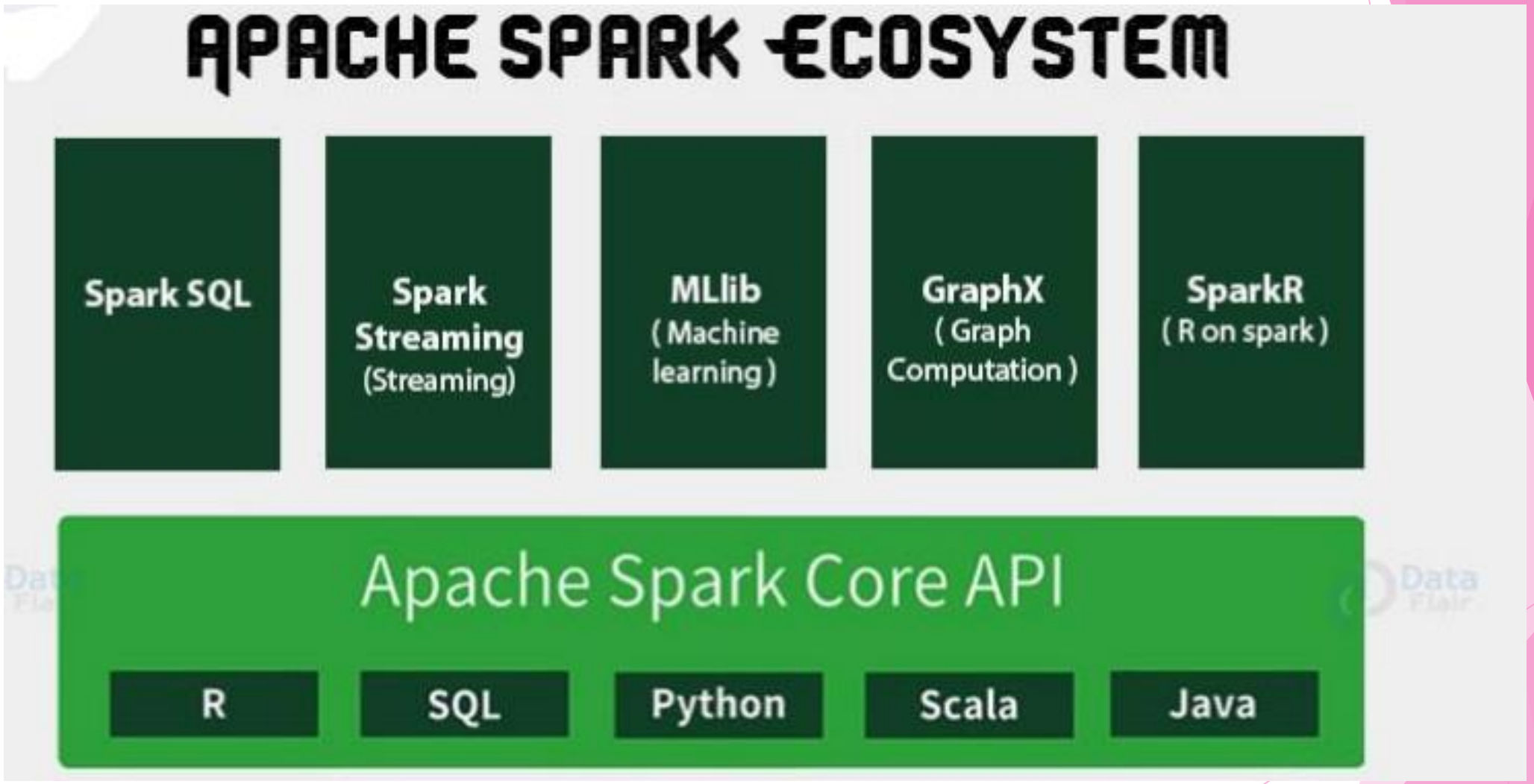
- These tools contain some predefined algorithms, functions, and user-friendly graphical user interfaces (GUIs).

DIFFERENT DATA SCIENCE PROCESSING TOOLS

- Spark
- Mesos
- Akka
- Cassandra
- Kafka
- Elastic Search
- R
- Scala
- Python
- MQTT(Message Queuing Telemetry Transport.)

Spark

- Apache Spark is an open source clustering computing framework.
- Apache Spark provide an interface for the programmer and developer to directly interact with the system and make data parallel and compatible with data scientist and data engineer.
- Apache Spark has the capabilities and potential, process all types and variety of data with repositories including Hadoop distributed file system, NoSQL database as well as apache spark.


APACHE SPARK ECOSYSTEM
Spark SQL
Spark Streaming (Streaming)
MLlib (Machine learning)
GraphX (Graph Computation)
SparkR (R on spark)
Apache Spark Core API
R
SQL
Python
Scala
Java

•**Spark Core:** basic functionality of Spark (task scheduling, memory management, fault recovery, storage systems interaction).

•**Spark SQL:** package for working with structured data queried via SQL as well as HiveQL

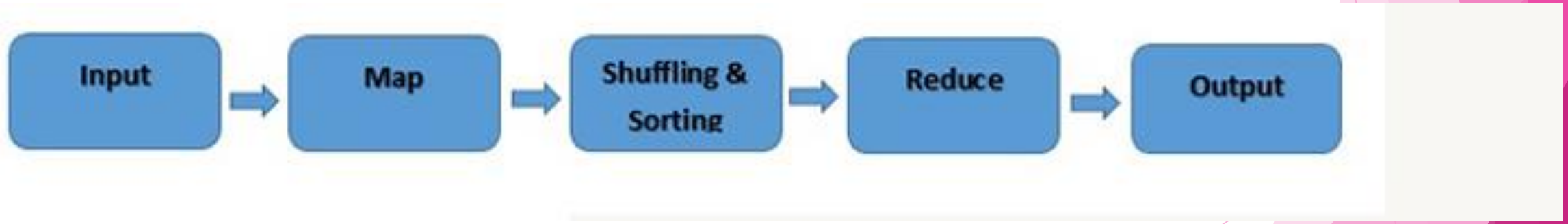
•**Spark Streaming:** a component that enables processing of live streams of data (e.g., log files, status updates messages)

•**MLLib**: MLLib is a machine learning library like Mahout. It is built on top of Spark and has the provision to support many machine learning algorithms.

•**GraphX**: For graphs and graphical computations, Spark has its own Graph Computation Engine, called GraphX.

Spark

- It has been built on top of the Hadoop distributed file system which make the data more efficient, more reliable and make it more extendable on the Hadoop map reduce



MESOS

- **Apache Mesos** *is the first Open Source cluster manager that handles the workload in* distributed *environment through dynamic resource sharing and isolation*.

- Mesos was developed at the University of California, Berkley.

- It is good for deployment and management of applications in large-scale cluster environments.

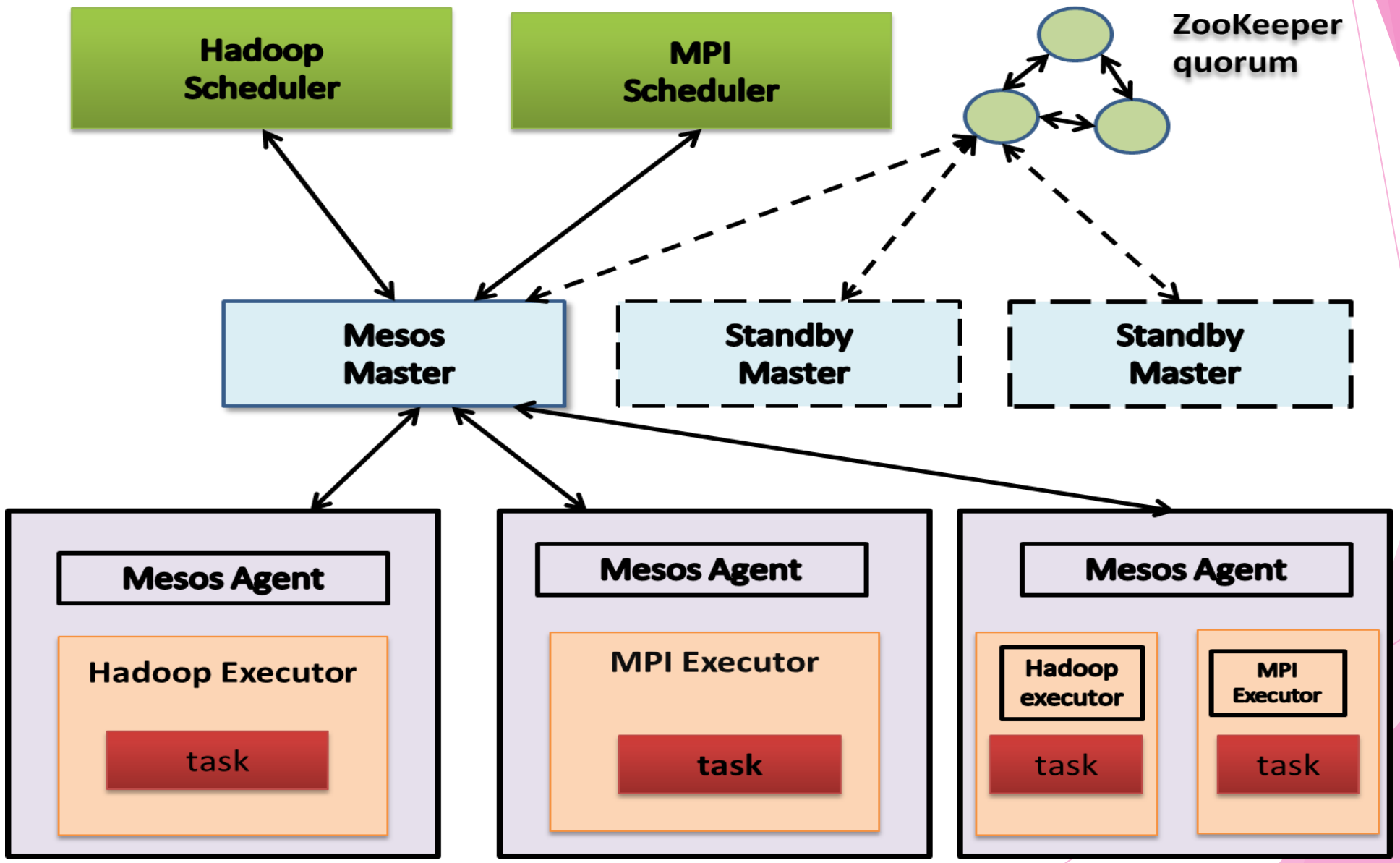
- It is a resource management platform for Hadoop and Big Data cluster. Companies such as Twitter, Xogito, and Airbnb utilize Apache Mesos.

NEED OF MESOS

- Many resource manager exists today like Hadoop on Demand, batch scheduler(e.g. Torque), VM Scheduler but the problem with Hadoop like workload is **data locality,** it was compromised because of static partitioning of nodes and since the job holds nodes for full duration of time, **utilization** of the system was affected.

- Mesos shares a resource in a fine-grained manner meaning it allows the framework to achieve data locality by taking turns and reading data stored on each machine.

ARCHITECTURE OF APACHE MESOS



ARCHITECTURE OF APACHE MESOS

- Mesos Master
- The master is a central source of all running task, it stores in memory all the data related to the task.
- For the completed task, there is only fixed amount of memory available, thus allowing the master to serve the user interface and data about the task with the minimal latency.

- Mesos Agent
- The Mesos agent publishes the information related to the host they are running in, including data about running task and executors, available resources of the host and other metadata. It guarantees the delivery of status update of the tasks to the schedulers.

MESOS FRAMEWORK

- Mesos Framework has two parts: The Scheduler and The Executor.

- It is the responsibility of scheduler to launch task when the resource requirement and constraints match with received offer the Mesos master. It is also responsible for handling task failures and errors.

- The **executor** executes the task launched by the scheduler and notifies back the status of each task.

AKKA

- Akka is a open-source library or a toolkit. It is used to create concurrent, distributed and fault-tolerant application.

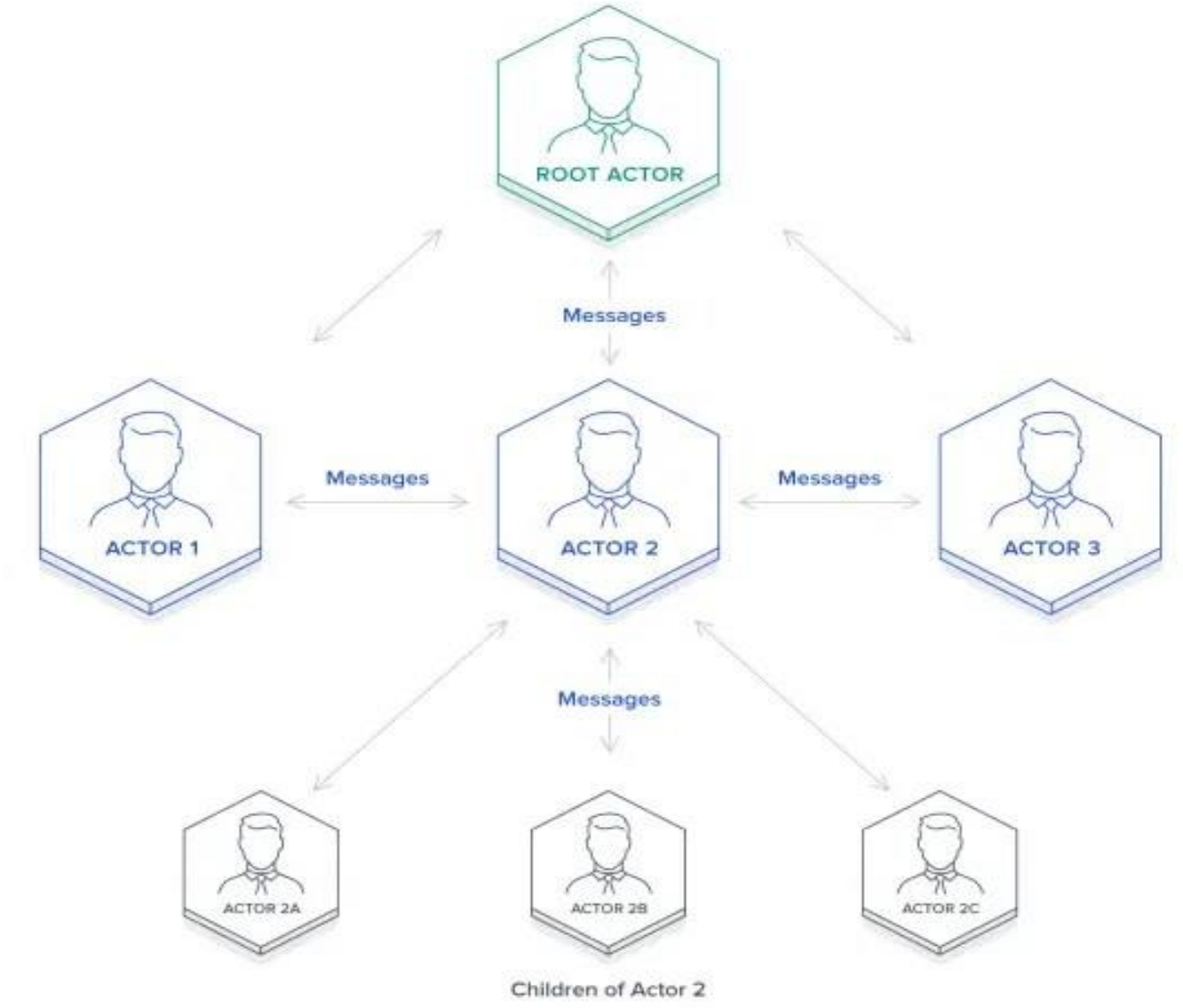
- Akka is written in Scala. It implements Actor Based Model. The Actor Model provides a higher level of abstraction for writing concurrent and distributed applications.

Akka Actor

- An actor is essentially nothing more than an object that receives messages and takes actions to handle them and It is decoupled from the source of the message so it’s only responsible for properly recognizing the type of message it has received and taking action accordingly.

Upon receiving a message, an actor may take one or more of the following actions:

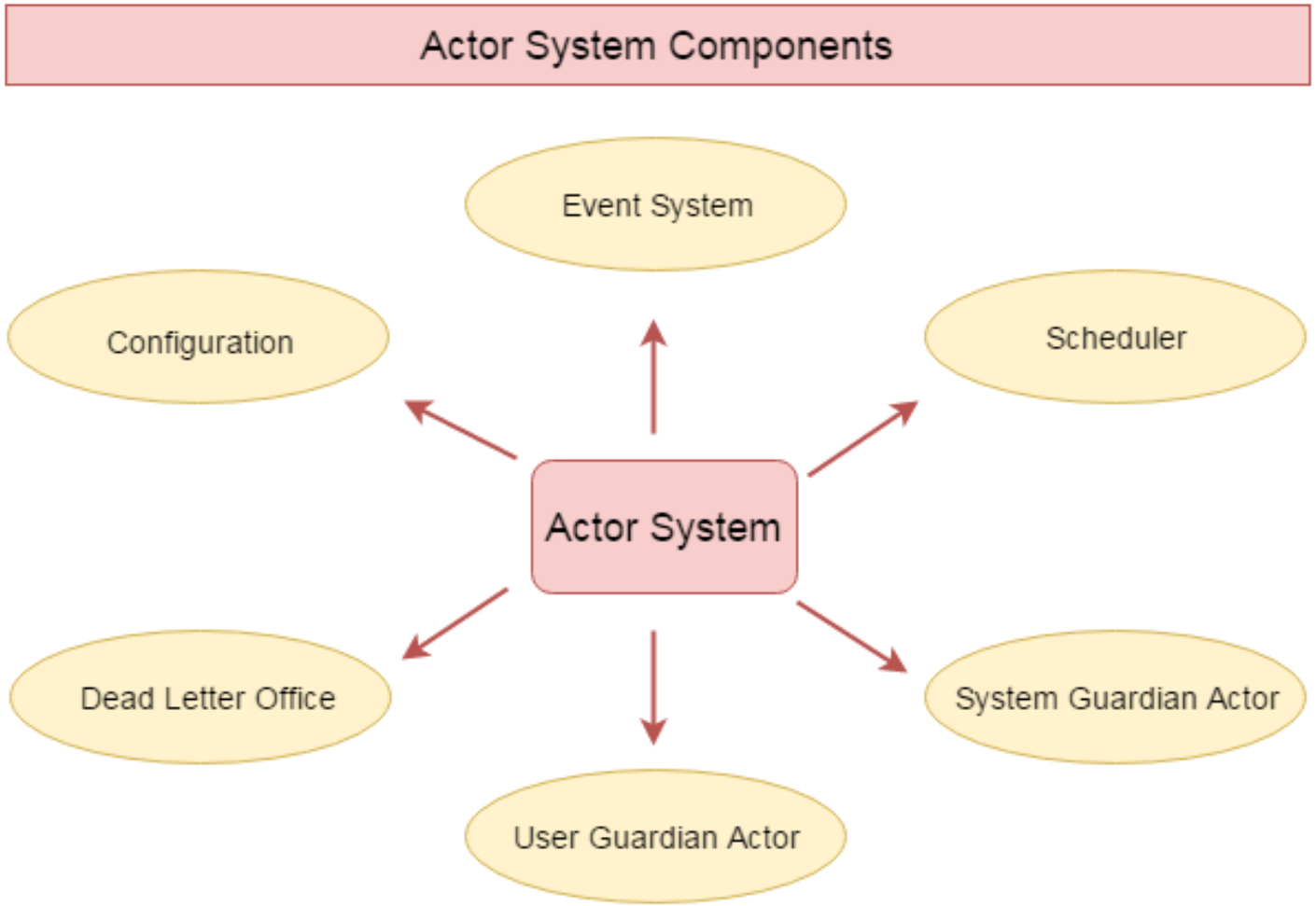
- Execute some operations itself (such as performing calculations, calling an external web service, and so on).
- Forward the message, or a derived message, to another actor.
- Instantiate a new actor and forward the message to it


ROOT ACTOR
Messages
ACTOR 1
Messages
ACTOR 2
Messages
ACTOR 3
Messages
ACTOR 2A
ACTOR 2B
ACTOR 2C
Children of Actor 2

WHY NAMED AKKA?

- It is the name of a beautiful Swedish mountain up in the northern part of Sweden called Laponia. The mountain is also sometimes called 'The Queen of Laponia'.

Akka ActorSystem Components



DEADLETTER OFFICE

- Messages sent via unreliable network will be lost without forwarding to dead letter office.
- The main use of this facility is for debugging purpose, especially if a message sent by an actor does not arrive consistently.
- You can implement this by importing **akka.actor.DeadLetter** package.

User Guardian Actor

- It is parent actor of actors created by user by using ActorSystem. This special guardian is used to achieve an orderly shut-down sequence where logging remains active while all normal actors terminated.
- It monitors all user created actors.

System Guardian Actor

- The system guardian monitors the user guardian and initiate its own shut-down upon reception of the Terminated message.

Scheduler

- It is used to handle scheduled tasks. It provides the facility to schedule messages.
- You can schedule sending of messages and execution of tasks. It creates new instance for each ActorSystem for scheduling tasks to happen at specific time.
- You can implement Scheduler by importing **akka.actor.Scheduler** package.

Event System

- It is used to carry log messages and dead letters. You can also used it to publish messages across entire ActorSystem.

- You can get eventStream reference by calling actorSystemRef.eventStream() method.

Configuration

- ActorSystem provides a configuration component which is used to configure application.
- You can access it from your actor system.

CASSANDRA

- Apache Cassandra is a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.

- It is a type of NoSQL database. Cassandra was developed at Facebook for inbox search.

- It was open-sourced by Facebook in July 2008.

FEATURES OF CASSANDRA

- **Elastic scalability** – Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement.

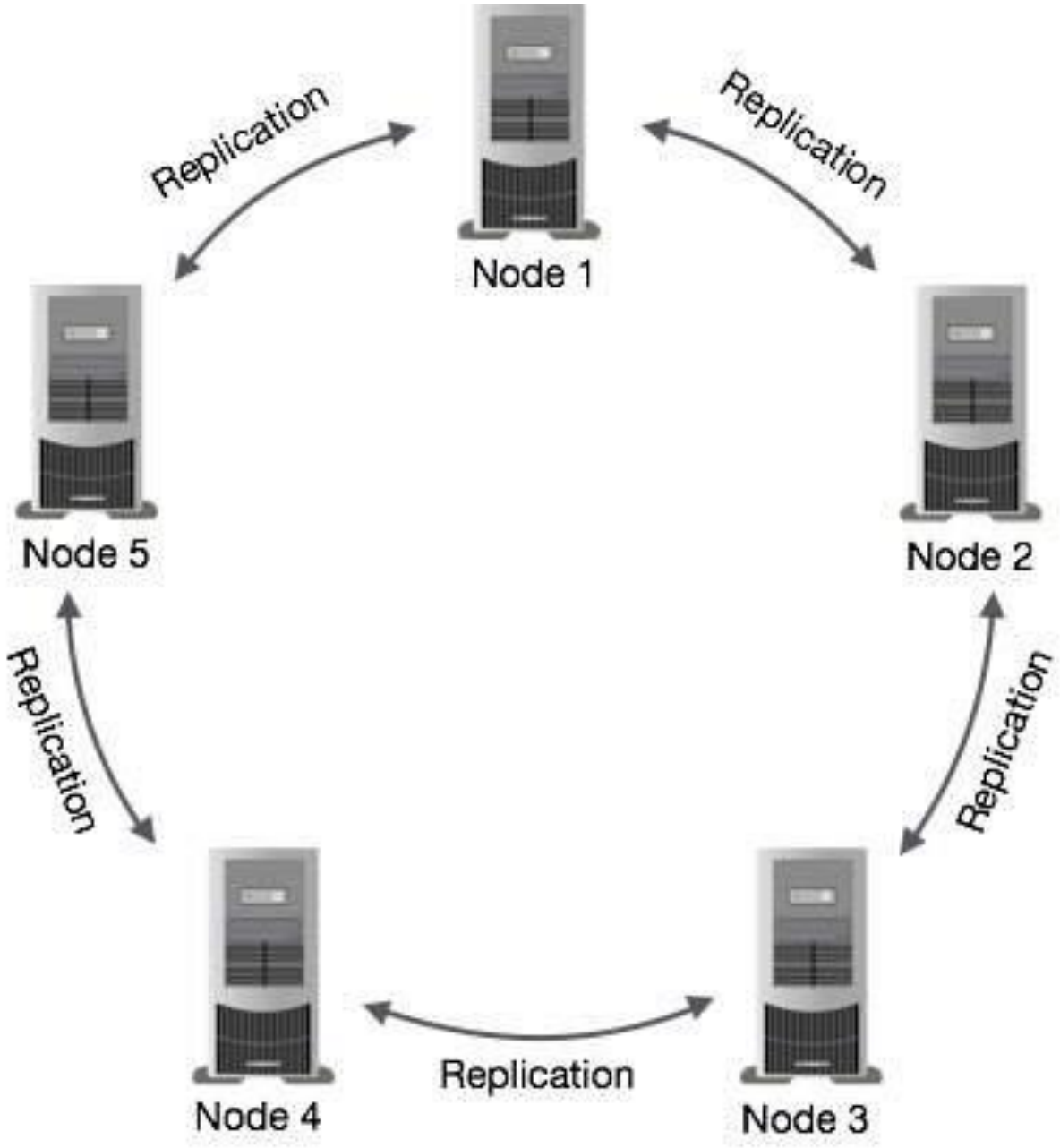
- **Fast linear-scale performance** – Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster. Therefore it maintains a quick response time.

- **Flexible data storage** – Cassandra accommodates all possible data formats including: structured, semi-

CASSANDRA ARCHITECTURE

- The design goal of Cassandra is to handle big data workloads across multiple nodes without any single point of failure.

- Cassandra has peer-to-peer distributed system across its nodes, and data is distributed among all the nodes in a cluster.


Node 1
Node 2
Node 3
Node 4
Node 5
Replication
Replication
Replication
Replication
Replication

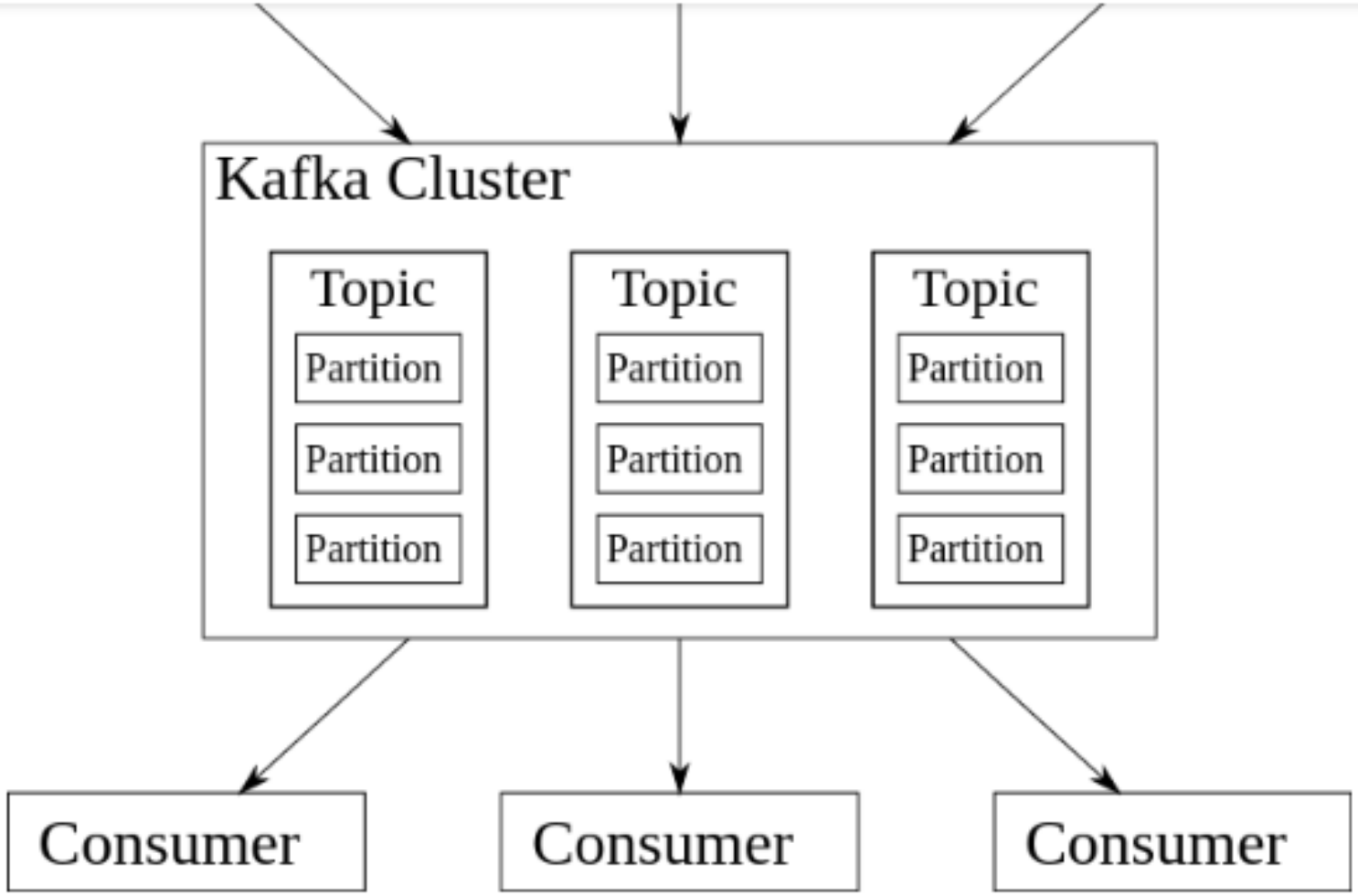
KAFKA

- Kafka is an open source software which provides a framework for storing, reading and analysing streaming data.
- Apache Kafka is used by big internet companies like LinkedIn, Twitter, Airbnb, and many others.
- Kafka is designed to be run in a “distributed” environment, which means that rather than sitting on one user’s computer, it runs across several (or many) servers, leveraging the additional processing power and storage capacity that this brings.

KAFKA

- Kafka was originally created at LinkedIn, where it played a part in analysing the connections between their millions of professional users in order to build networks between people.

Apache Kafka Architecture and Use Cases Explained



What is Kafka used for?

- In order to stay competitive, businesses today rely increasingly on real-time data analysis allowing them to gain faster insights and quicker response times.

How does Kafka work?

- Apache takes information – which can be read from a huge number of data sources – and organises it into “topics”. As a very simple example, one of these data sources could be a transactional log where a grocery store records every sale.

- Kafka would process this stream of information and make “topics” – which could be “number of apples sold”, or “number of sales between 1pm and 2pm” which could be analysed by anyone needing insights into the data.

- This may sound similar to how a conventional database lets you store or sort information, but in the case of Kafka it would be suitable for a national chain of grocery stores processing thousands of apple sales every minute.

ELASTIC SEARCH

- Elastic search is a distributed, open source search and analytics engine designed for horizontal scalability, reliability, and stress-free management.

- It combines the speed of search with the power of analytics, via a sophisticated, developer-friendly query language covering structured, unstructured, and time-series data.

- Some of the projects were **Logstash** (data processing pipeline, commonly used for parsing text-based files).

R

- R-language was introduced in Feb 2000 by **R-Foundation**. It is written in Fortran.
- R is defined as the programming language, mainly used in statistical computing and graphics.
- It is a free software environment used by leading data miners, practitioners and statisticians.
- Language is primarily beneficial in the development of statistical-based software and data analytics.

R

- Companies like Barclays, American Express, and Bank of America use R-Language for their data analytics needs.

Scala for **Data Science**

SCALA

- Scala is a high-level programming language that mixes object-oriented and functional programming.

- Scala's name implies that it is a scalable programming language. It was created in 2003 by Martin Odersky and his research team. These days we widely use Scala in Data Science and **Machine Learning** fields.

- Scala is a small, fast, and efficient multi-paradigm programming language built on a compiler. The **JVM(Java Virtual Machine)** is Scala's main advantage . Scala code is first compiled by a Scala compiler, which generates bytecode, which is then transported to the JVM for output generation.

Why we learn Scala for Data Science:

PYTHON

- Python is a high-level, general-purpose programming language created by Guido van Rossum and released in 1991.

- It is important to note that it is an interpreted language: Python has a design philosophy that emphasizes code readability.

- Python uses a dynamic type system and automatic memory management and supports multiple programming paradigms (object-oriented, imperative, functional programming, and procedural).

MQTT

- MQTT stands for **Message Queuing Telemetry Transport**. MQTT is a machine to machine internet of things connectivity protocol.

- It is a machine to machine protocol, i.e., it provides communication between the devices.

- It is a publish and subscribe system where we can publish and receive the messages as a client. It makes it easy for communication between multiple devices.

COMPONENTS OF MQTT

- Message
- The message is the data that is carried out by the protocol across the network for the application.

- **Client**
- If any program or device uses an MQTT, then that device is referred to as a client.

- **Publish:** When the client sends the data to the server, then we call this operation as a publish.

- **Subscribe:** When the client receives the data from the server, then we call this operation a subscription.

COMPONENTS OF MQTT

- A server accepts the network connection from the client, accepts the messages from the client, processes the subscribe and unsubscribe requests, forwards the application messages to the client, and closes the network connection from the client.

- The label provided to the message is checked against the subscription known by the server is known as TOPIC.

DATA SCIENCE FRAMEWORK

- Data science is a series of discoveries of converting raw unstructured data from your data lake into actionable business data. This process is a cycle of discovering and evolving your understanding of the data you are working with to supply you with metadata that you need. We need to build a basic framework that is used for data processing.

- A data science framework is a collection of libraries that provides data mining functionality, i.e., methods for exploring the data, cleaning it up, and transforming it into some more useful format that can be used for data processing or machine learning tasks.

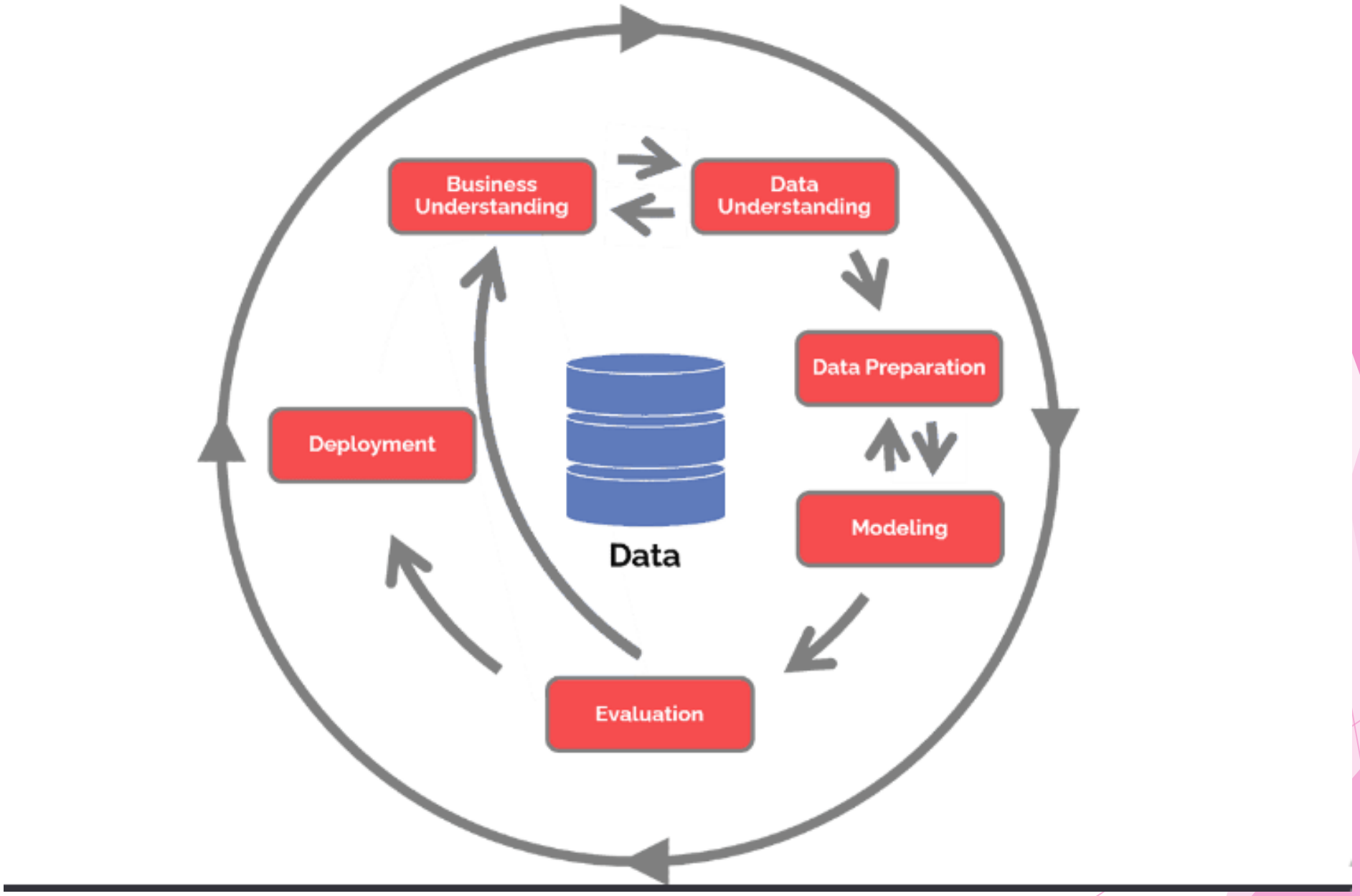
CROSS INDUSTRY STANDARD PROCESS FOR DATA MINING(CRISP-DM)

- CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is an industry-proven way to guide your data mining efforts.

- As a **methodology**, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks.

- As a **process model**, CRISP-DM provides an overview of the data mining life cycle.

Published in 1999 to standardize data mining processes across industries, it has since become the most common methodology for data mining, analytics, and data science projects.


Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Data

PHASES OF CRISP-DM

- Business understanding – What does the business need?
- Data understanding – What data do we have / need? Is it clean?
- Data preparation – How do we organize the data for modeling?
- Modeling – What modeling techniques should we apply?
- Evaluation – Which model best meets the business objectives?
- Deployment – How do stakeholders access the results?

Homogeneous Ontology for Recursive Uniform Schema

- The Homogeneous Ontology for Recursive Uniform Schema (HORUS) is used as an internal data format structure that enables the framework to reduce the permutations of transformations required by the framework.

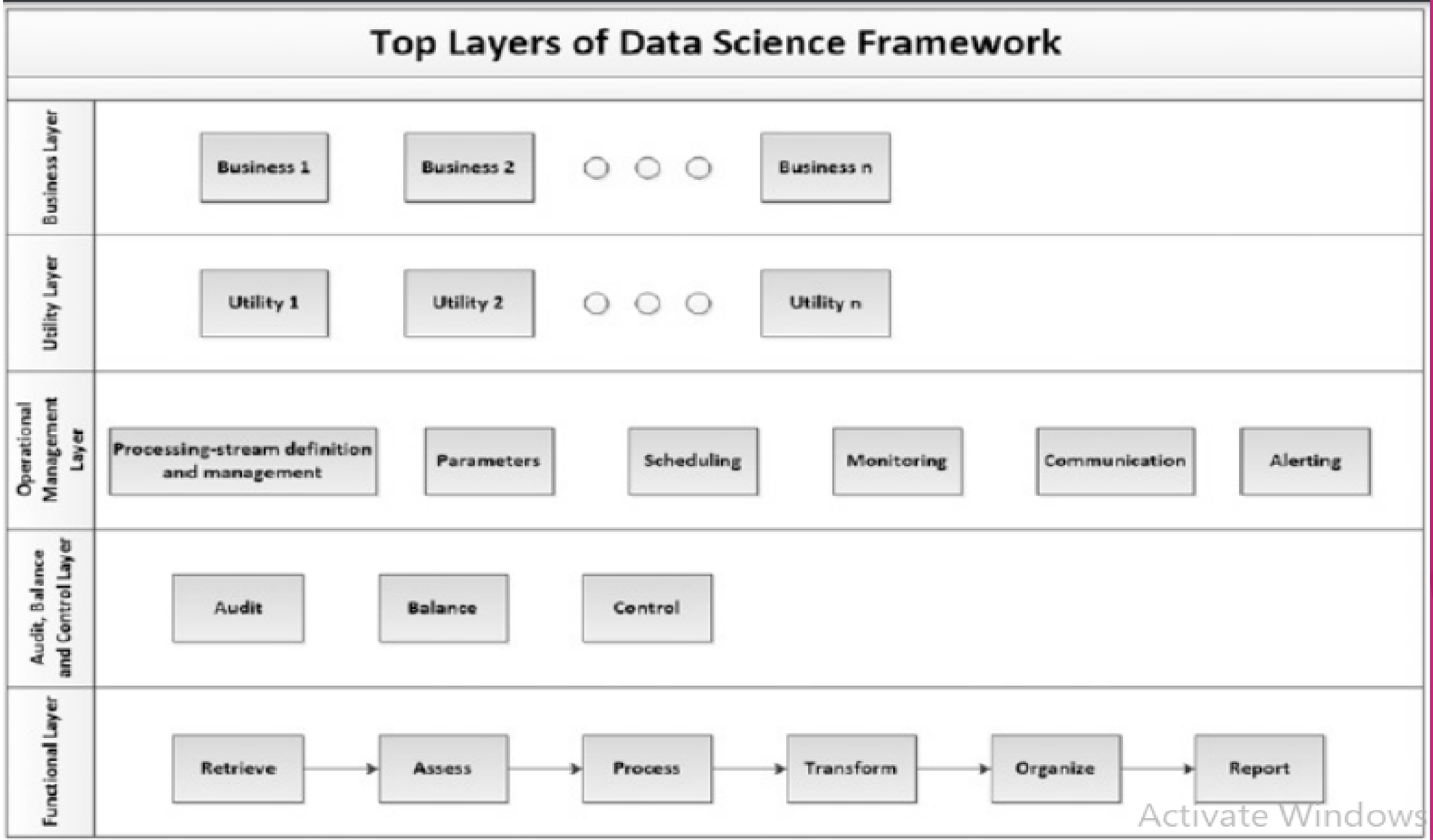
- External data formats are converted to HORUS format, and then a HORUS format is transformed into any other external format.

Homogeneous Ontology for Recursive Uniform Schema

- The basic concept is to take native raw data and then transform it first to a single format.
- That means that there is only one format for text files, one format for JSON or XML, one format for images and video.
- Therefore, to achieve any-to-any transformation coverage, the framework's only requirements are a data-format-to-HORUS and HURUS-to- dataformat converter.

TOP LAYERS OF A LAYERED FRAMEWORK

- Business Layer
- Utility Layer
- Operational Management Layer
- Audit, balance and control layer
- Functional Layer



BUSINESS LAYER

- The business layer indicated is the principal source of the requirements and business information needed by data scientists and engineers for processing. It includes:

- Up-to-date organizational structure chart
- Business description of the business processes you are investigating
- List of your subject matter experts
- Project plans
- Budgets
- Functional requirements
- Nonfunctional requirements

UITLITY LAYER

- The utility layeris a common area in which you store all your utilities. You will be faced regularly with immediate requirements demanding that you fix something in the data or run your “magical” algorithm.

OPERATIONAL MANAGEMENT LAYER

- Operations management is an area of the ecosystem concerned with designing and controlling the process chains of a production environment and redesigning business procedures.
- This layer stores what you intend to process. It is where you plan your data science processing pipelines.

OPERATIONAL MANAGEMENT LAYER

- The operations management layer is where you record
- Processing-stream definition and management
- Parameters
- Scheduling
- Monitoring
- Communication
- Alerting

- The operations management layer is the common location where you store any of the processing chains you have created for your data science.

AUDIT,BALANCE AND CONTROL LAYER

- The audit, balance, and control layer, represented in Figure 3-2, is the area from which you can observe what is currently running within your data science environment.
- It records
- • Process-execution statistics
- • Balancing and controls
- • Rejects and error-handling
- • Codes management

FUNCTIONAL LAYER

- The functional layer of the data science ecosystem is the main layer of programming required.

- The functional layer is the part of the ecosystem that executes the comprehensive data science. It consists of several structures.
- • Data models
- • Processing algorithms
- • Provisioning of infrastructure

BUSINESS LAYER

- The business layer does not belong to the data scientist 100%, and normally, its success represents a joint effort among such professionals as business subject matter experts, business analysts, hardware architects, and data scientists.

- The business layer is where we record the interactions with the business. This is where we convert business requirements into data science requirements.

FUNCTIONAL REQUIREMENTS

- These requirements are the business's view of the system, which can also be described as the "Will of the Business."

- " The MoSCoW method is a prioritization technique, to indicate how important each requirement is to the business.

Table 4-1. MoSCoW Options

Must have	Requirements with the priority "must have" are critical to the current delivery cycle.
Should have	Requirements with the priority "should have" are important but not necessary to the current delivery cycle.
Could have	Requirements prioritized as "could have" are those that are desirable but not necessary, that is, nice to have to improve user experience for the current delivery cycle.
Won't have	Requirements with a "won't have" priority are thoseidentified by stakeholders as the least critical, lowest payback requests, or just not appropriate at that time in the delivery cycle.

SPECIFIC FUNCTIONAL REQUIREMENTS

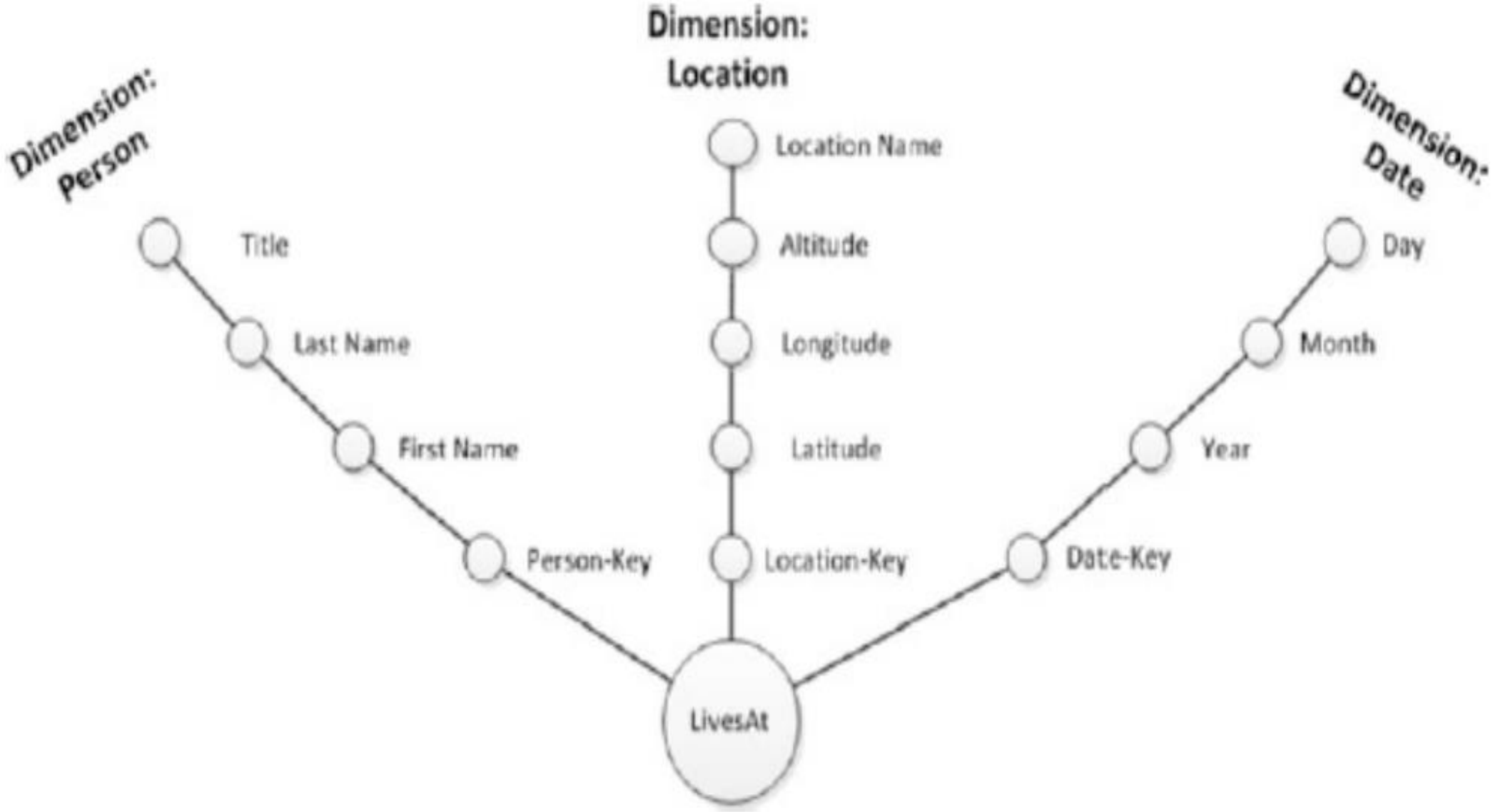
- DATA MAPPING MATRIX

- The data mapping matrix is one of the core functional requirement recording techniques used in data science. It tracks every data item that is available in the data sources.

SUN MODELS

- The sun models is a requirement mapping technique that assists you in recording requirements at a level that allows your nontechnical users to understand the intent of your analysis, while providing you with an easy transition to the detailed technical modeling of your data scientist and data engineer.

- This sun model is for fact LivesAt and supports three dimensions: person, location, and date.

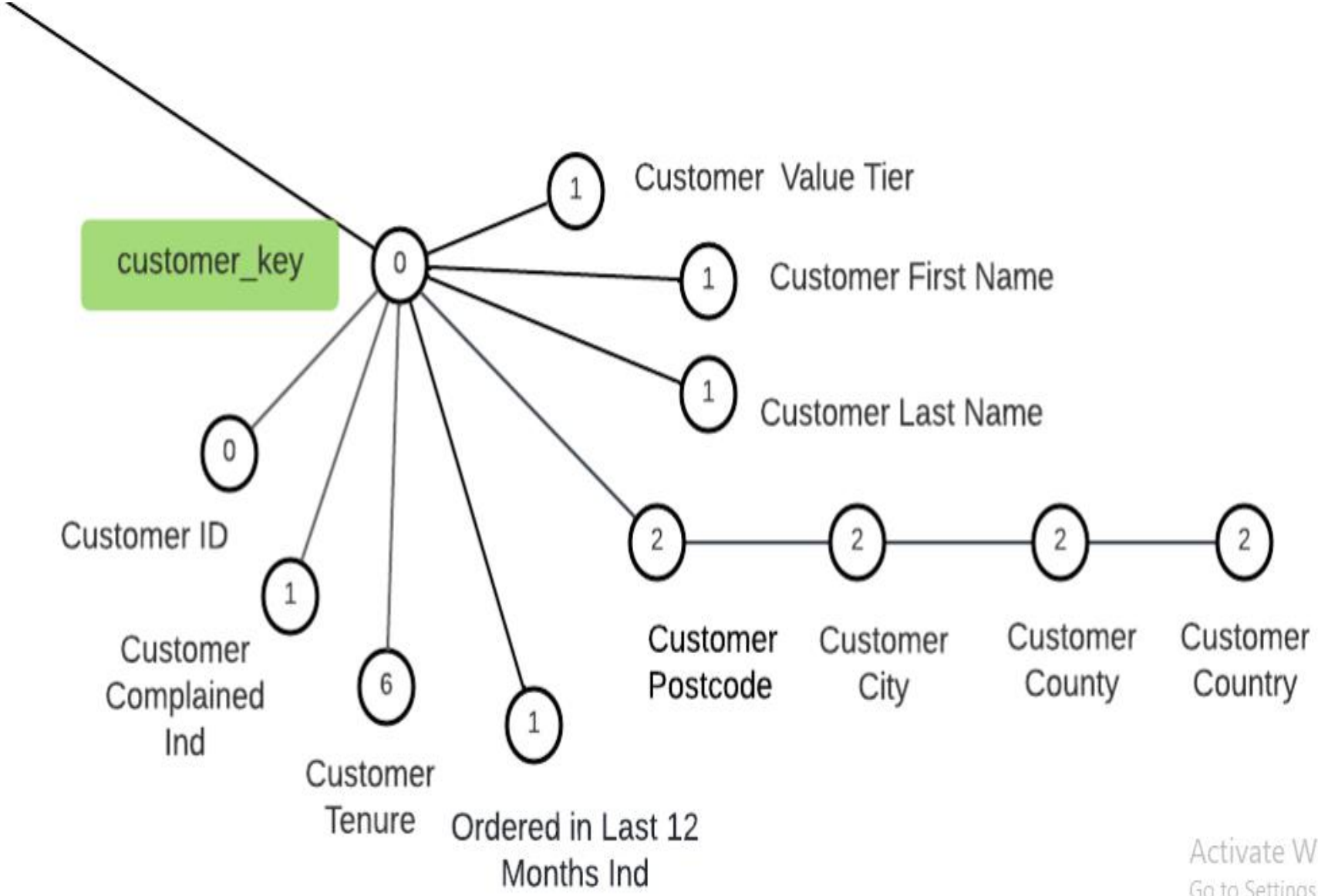

Dimension: Person
Title
Last Name
First Name
Person-Key
Dimension: Location
Location Name
Altitude
Longitude
Latitude
Location-Key
Dimension: Date
Day
Month
Year
Date-Key
LivesAt

DIMENSIONS

- A dimension is a structure that categorizes facts and measures, to enable you to respond to business questions. A slowly changing dimension is a data structure that stores the complete history of the data loads in the dimension structure over the life cycle of the data lake.

FACTS

- A fact is a measurement that symbolizes a fact about the managed entity in the real world.



0 = Type 0 – Take the original value and ignore any changes

1 = Type 1 – Show the current value

2 = Type 2 – Show the value based on the time of the event

3 – Type 3 – 2 attributes for the original and current

6 = Type 6 – (1+2+3 combined) 2 attributes one for the current and one based on the time of the event

Non-Functional Requirements

- Nonfunctional requirements record the precise criteria that must be used to appraise the operation of a data science ecosystem.

Non-Functional Requirements

- Accessibility Requirements Accessibility can be viewed as the “ability to access” and benefit from some system or entity.

- The concept focuses on enabling access for people with disabilities, or special needs, or enabling access through assistive technology.

Audit and Control Requirements

- Audit is the ability to investigate the use of the system and report any violations of the system's data and processing rules.

- Control is making sure the system is used in the manner and by whom it is preapproved to be used. An approach called role-based access control (RBAC) is the most commonly used approach to restricting system access to authorized users of your system.

Availability Requirements

- Availability is as a ratio of the expected uptime of a system to the aggregate of the downtime of the system.

- You should encourage your customers to specify availability requirements with precision. Consider the difference between an uptime of 99.70 percent and an uptime of 99.95 percent.

An uptime of 99.70 percent means the network is down 30 minutes per week, which is not acceptable to many customers. An uptime of 99.95 percent means the network is down 5 minutes per week, which may be acceptable, depending on the type of business.

Availability requirements should be specified with at least two digits following the decimal point.

Backup Requirements

- A backup, or the process of backing up, refers to the archiving of the data lake and all the data science programming code, programming libraries, algorithms, and data models, with the sole purpose of restoring these to a known good state of the system, after a data loss or corruption event. R

COMMON PITFALLS WITH REQUIREMENTS

- Users must easily access the system.
- What is “easily”?
- • Use reliable technology.
- What is “reliable”?

UTILITY LAYER

- The utility layer is a central storehouse for keeping all one’s solutions utilities in one place. Having a central store for all utilities ensures that you do not use out-of-date or duplicate algorithms in your solutions.

European Union General Data Protection Regulation (GDPR)

- The GDPR has the following rules:
- You must have valid consent as a legal basis for processing
- You must assure transparency, with clear information about what data is collected and how it is processed.
- You must support the right to accurate personal data.
- Utilities must use only the latest accurate data
- You must support the right to have personal data erased.
- Utilities must support the removal of all information on a specific person.

THANKS