Architetture Parallele (Class. Flynn)

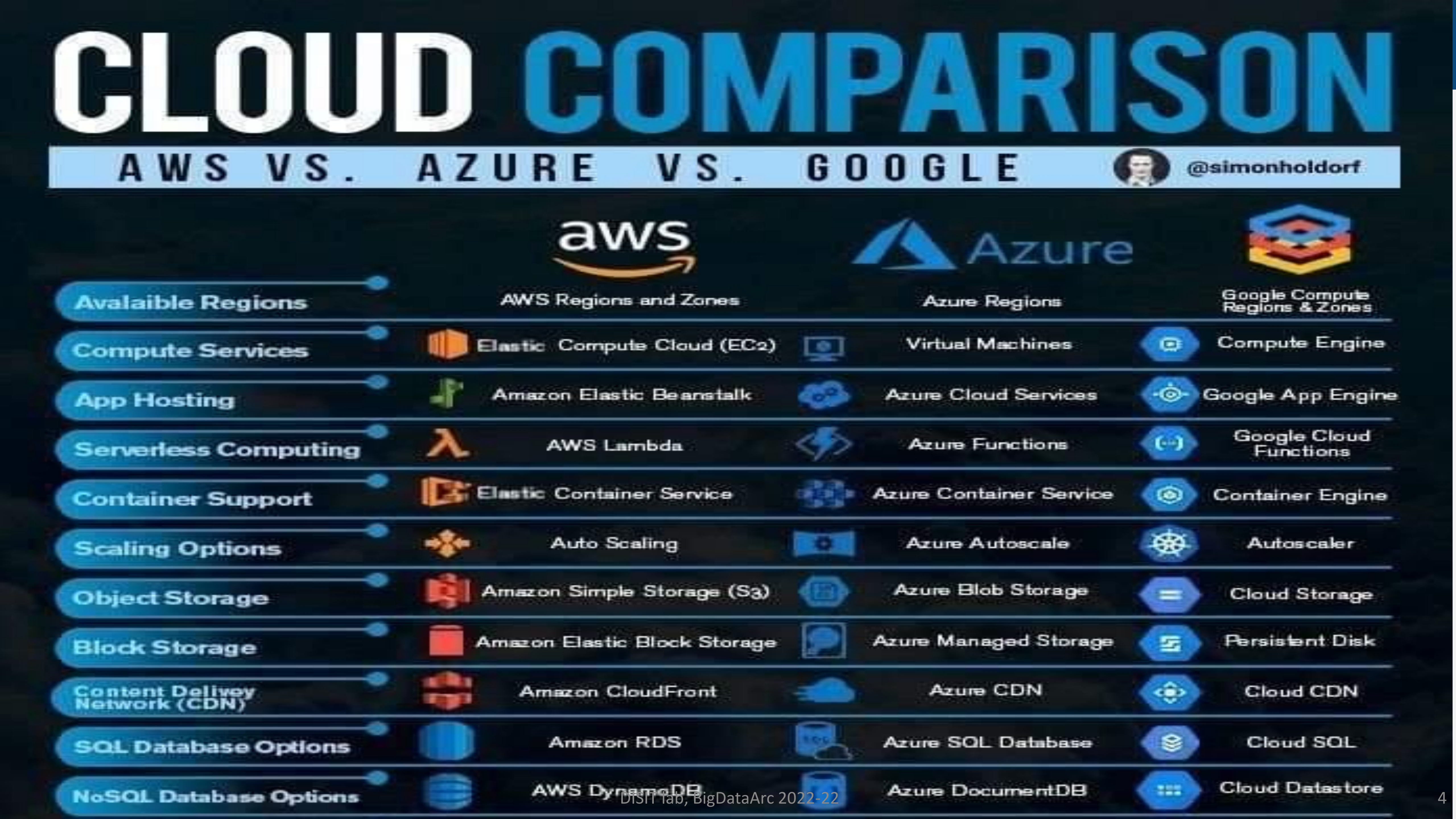
- **SISD**: Single instruction stream, single data stream
  - E.g.: un computer monoprocessore, monocore
- **SIMD**: Single instruction stream, multiple data stream
  - E.g.: le stesse istruzioni su tutti I nodi computazionali allo stesso tempo, ma lavorando su dati diversi, per esempio GPU su immagini
- **MISD**: Multiple instruction stream, single data stream
  - E.g.: ogni nodo puo' eseguire processi indipendenti ma sullo stesso stream di dati, un risultato singolo (poco realistico)
- **MIMD**: Multiple instruction stream, multiple data stream
  - E.g.: ogni nodo puo' eseguire processi indipendenti su dati diversi
  - Tipicamente la soluzione piu' utilizzata per il sistemi general purpose, cloud, etc.

Classificazione

- **SISD**: Single instruction stream, single data stream
  - Von Neumann
- **SIMD**: Single instruction stream, multiple data stream
  - Vector processor, Array processor
- **MISD**: Multiple instruction stream, single data stream
  - poco realistico

MIMD: Multiple instruction stream, multiple data stream

- **MultiProcessore**
  - Parallelismo interno al calcolatore
  - Memoria condivisa, variabili condivise
  - Sincronizzazioni
  - E.g., Uniform Memory Access: XEON
- **MultiComputer**
  - Cloud, architetture parallele
  - Comunicazione tramite canali dedicati,
  - Message passing, Send e Receive
    - ❑ Hypercubes
- **Sistemi Distribuiti**,
  - Cluster, Massive parallel processor
    - GRID, cloud, ..

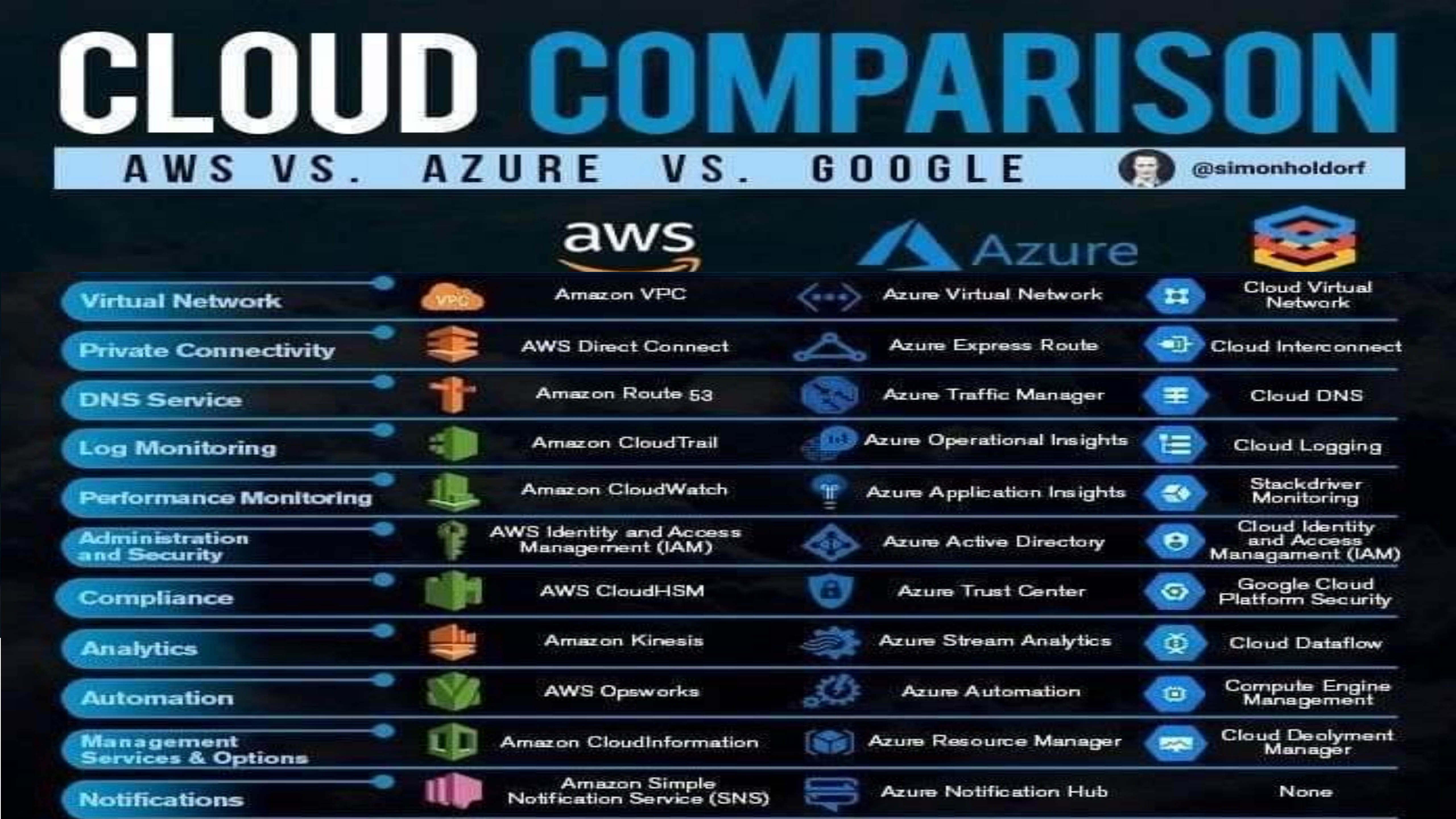


CLOUD COMPARISON

AWS VS. AZURE VS. GOOGLE

@simonholdorf

	aws	Azure	
Avalaible Regions	AWS Regions and Zones	Azure Regions	Google Compute Regions & Zones
Compute Services	Elastic Compute Cloud (EC2)	Virtual Machines	Compute Engine
App Hosting	Amazon Elastic Beanstalk	Azure Cloud Services	Google App Engine
Serverless Computing	AWS Lambda	Azure Functions	Google Cloud Functions
Container Support	Elastic Container Service	Azure Container Service	Container Engine
Scaling Options	Auto Scaling	Azure Autoscale	Autoscaler
Object Storage	Amazon Simple Storage (S3)	Azure Blob Storage	Cloud Storage
Block Storage	Amazon Elastic Block Storage	Azure Managed Storage	Persistent Disk
Content Delivey Network (CDN)	Amazon CloudFront	Azure CDN	Cloud CDN
SQL Database Options	Amazon RDS	Azure SQL Database	Cloud SQL
NoSQL Database Options	AWS DynamoDB	Azure DocumentDB	Cloud Datastore



CLOUD COMPARISON

AWS VS. AZURE VS. GOOGLE

@simonholdorf

	aws	Azure	
Virtual Network	Amazon VPC	Azure Virtual Network	Cloud Virtual Network
Private Connectivity	AWS Direct Connect	Azure Express Route	Cloud Interconnect
DNS Service	Amazon Route 53	Azure Traffic Manager	Cloud DNS
Log Monitoring	Amazon CloudTrail	Azure Operational Insights	Cloud Logging
Performance Monitoring	Amazon CloudWatch	Azure Application Insights	Stackdriver Monitoring
Administration and Security	AWS Identity and Access Management (IAM)	Azure Active Directory	Cloud Identity and Access Managament (IAM)
Compliance	AWS CloudHSM	Azure Trust Center	Google Cloud Platform Security
Analytics	Amazon Kinesis	Azure Stream Analytics	Cloud Dataflow
Automation	AWS Opsworks	Azure Automation	Compute Engine Management
Management Services & Options	Amazon CloudInformation	Azure Resource Manager	Cloud Deolyment Manager
Notifications	Amazon Simple Notification Service (SNS)	Azure Notification Hub	None

UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

https://www.disit.org/

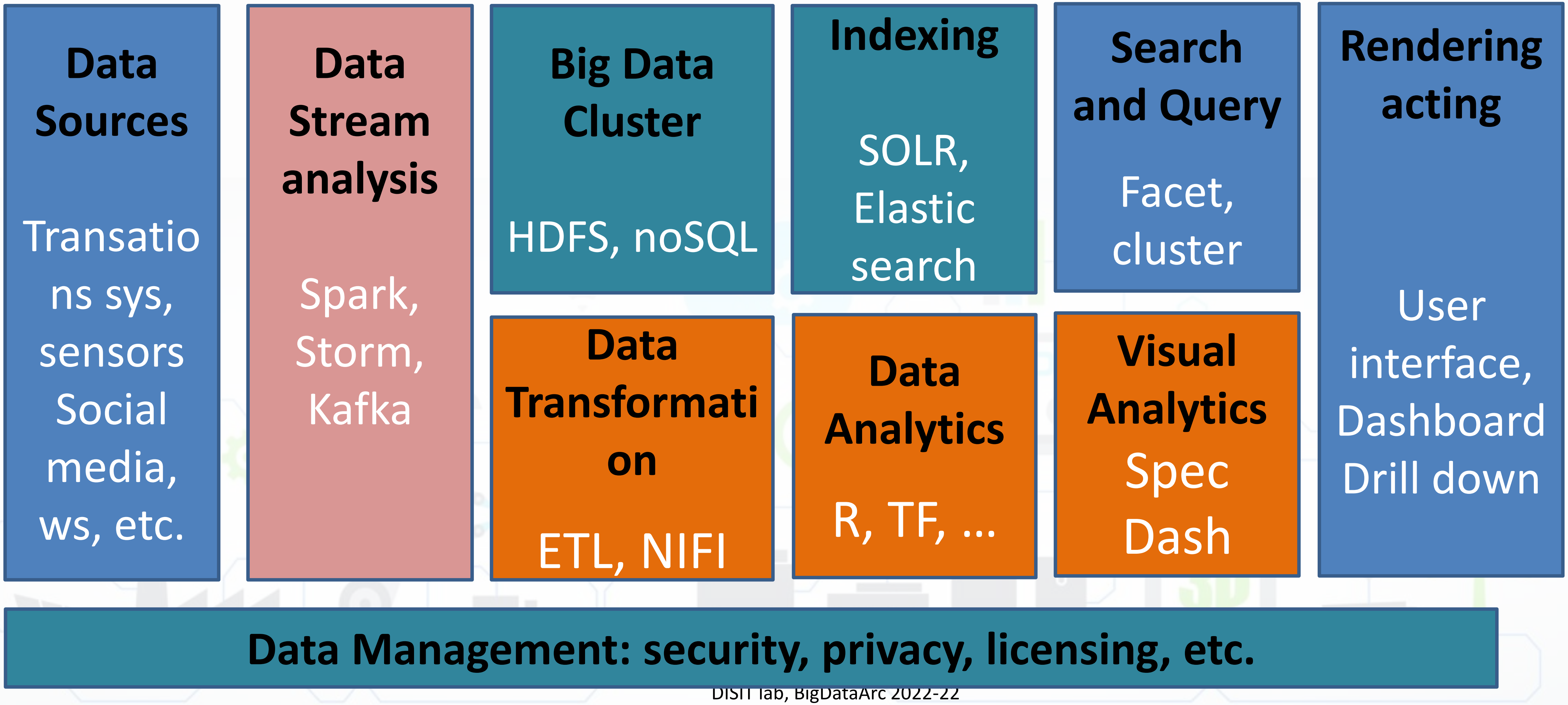
Paolo Nesi, paolo.nesi@unifi.it

Data Lake vs Data Warehouse

https://www.snap4City.org https://www.Km4City.org

Parte: x (2022)

Architettura di base Big Data, IOT, Industry 4.0



Data Sources

Transations sys, sensors Social media, ws, etc.

Data Stream analysis

Spark, Storm, Kafka

Big Data Cluster

HDFS, noSQL

Data Transformation

ETL, NIFI

Indexing

SOLR, Elastic search

Data Analytics

R, TF, …

Search and Query

Facet, cluster

Visual Analytics

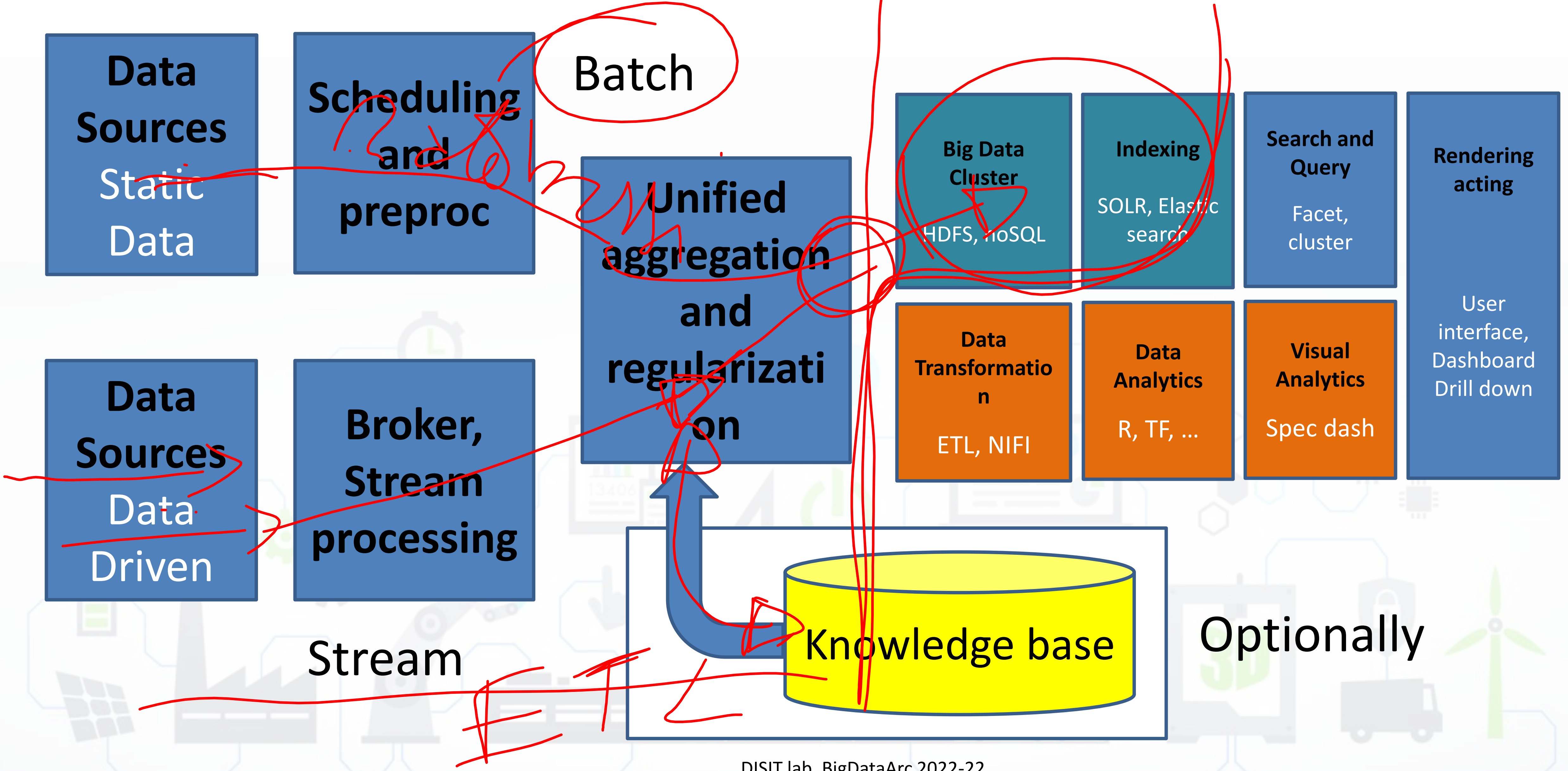
Spec Dash

Rendering acting

User interface, Dashboard Drill down

Data Management: security, privacy, licensing, etc.

Lambda Architecture



Data Sources Static Data

Scheduling and preproc

Batch

Data Sources Data Driven

Broker, Stream processing

Stream

Unified aggregation and regularization

Big Data Cluster

HDFS, noSQL

Indexing

SOLR, Elastic search

Search and Query

Facet, cluster

Rendering acting

User interface, Dashboard Drill down

Data Transformation

ETL, NIFI

Data Analytics

R, TF, …

Visual Analytics

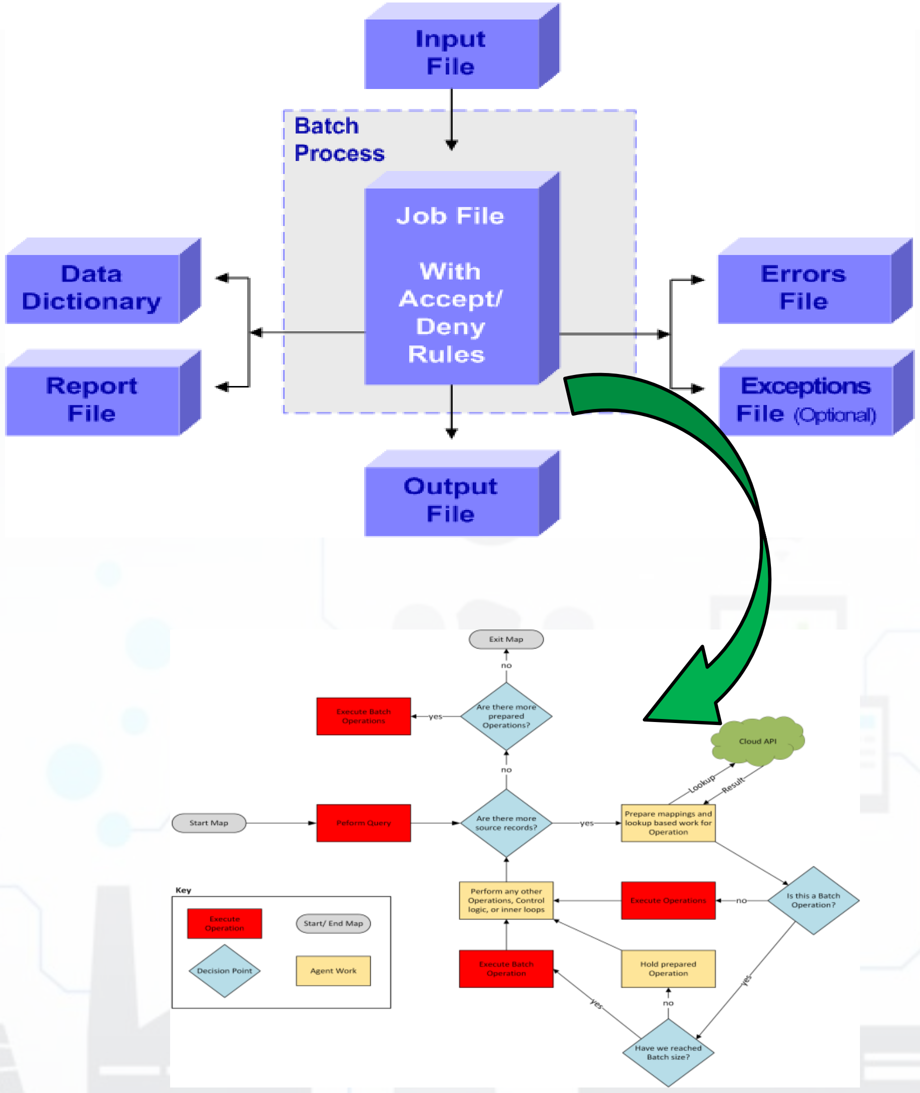
Spec dash

Knowledge base

Optionally

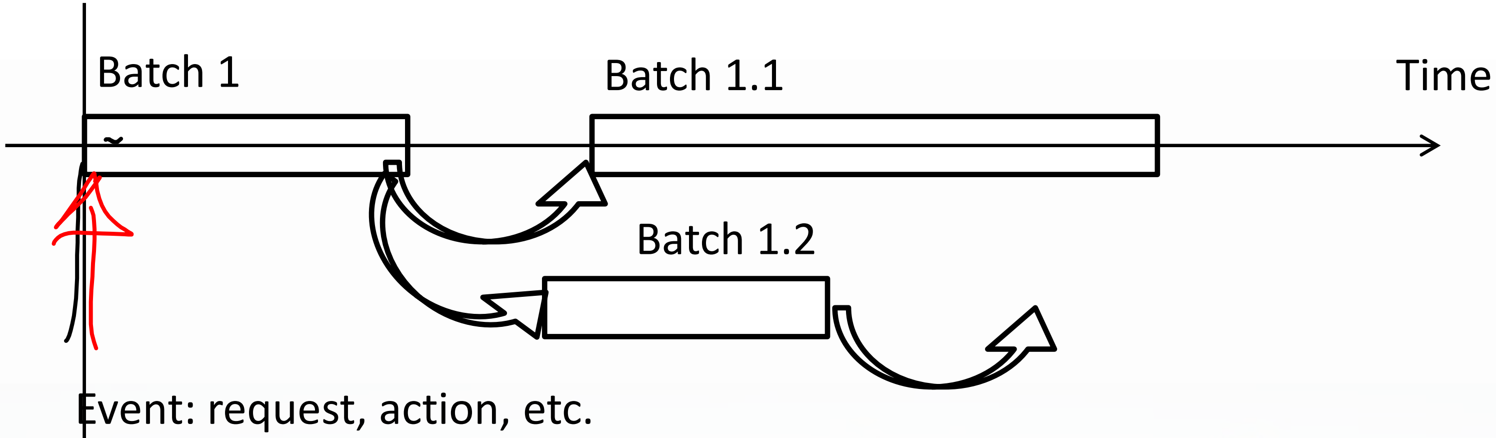
ETL

Batch Processing


Input File
Batch Process
Job File With Accept/ Deny Rules
Data Dictionary
Report File
Errors File
Exceptions File (Optional)
Output File
Exit Map
no
Execute Batch Operations
yes
Are there more prepared Operations?
no
Cloud API
Lookup
Result
Start Map
Peform Query
Are there more source records?
yes
Prepare mappings and lookup based work for Operation
Key
Execute Operation
Start/ End Map
Decision Point
Agent Work
Perform any other Operations, Control logic, or inner loops
Execute Operations
no
Is this a Batch Operation?
Execute Batch Operation
Hold prepared Operation
yes
no
yes
Have we reached Batch size?

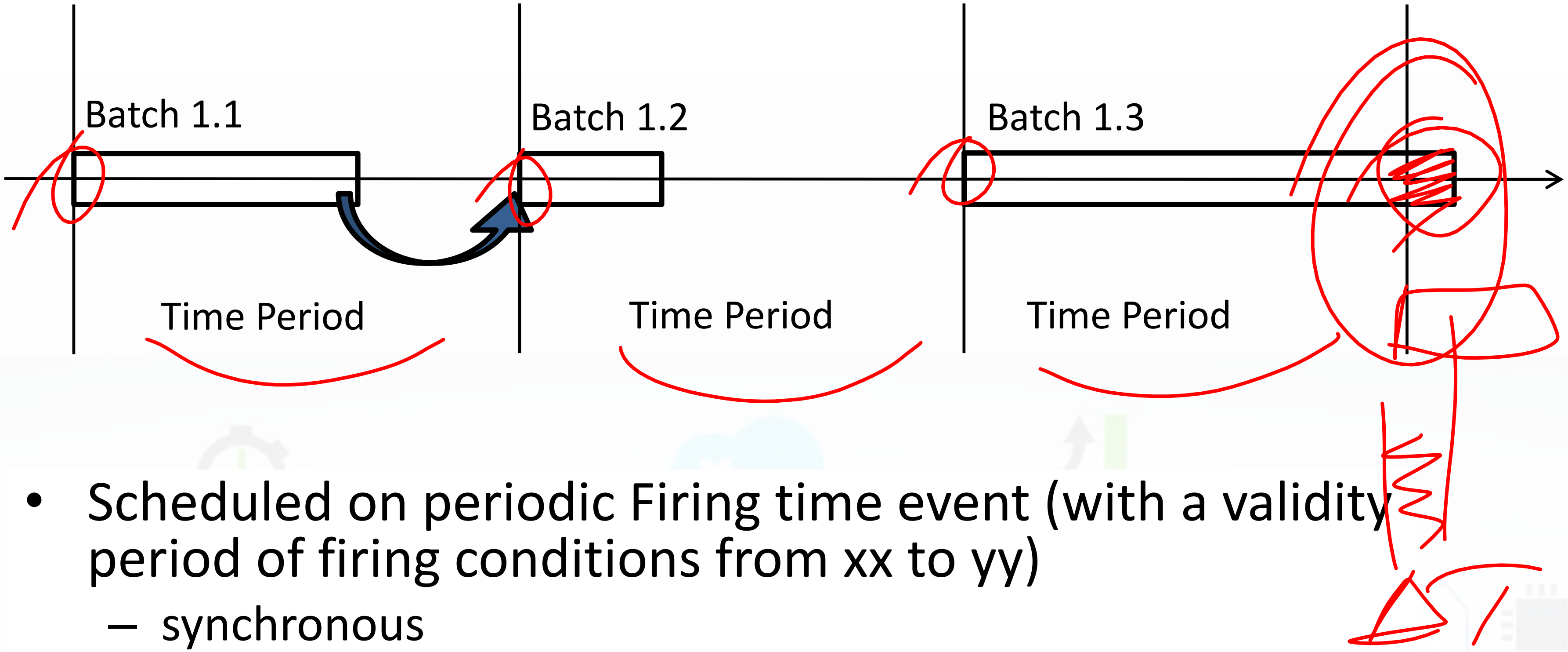
- Script language
- Smilar to workflow or flowchart
- Sequence of Commands
- Intermediate status on disk or memory
- Executed command driven:
  - On demand, sporadically
  - Periodically

aPeriodic Batch processing



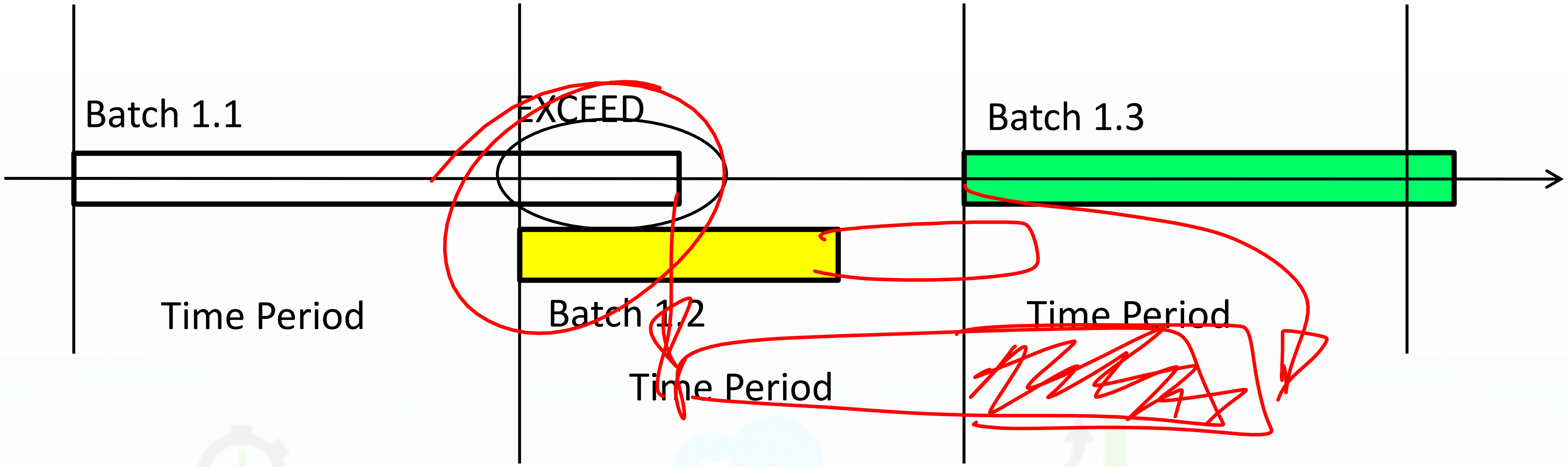
- Activated/Fired/Triggered by events, on demand, etc..
  - Synchronous with something
  - chained or not
- May fire/generate (ask to do or directly do) other jobs/batches:
  - on the same or different computers
  - Identical or different
- Activation may be asked to a third party manager
- Duration of execution depending on data !

Periodic Batch processing



- Scheduled on periodic Firing time event (with a validity period of firing conditions from xx to yy)
  - synchronous
- Execution time duration depending on data or other..
  - Execution may exceed the Time Period
- Each single execution MAY or MAY NOT depend on the preceding one

Periodic Batch processing



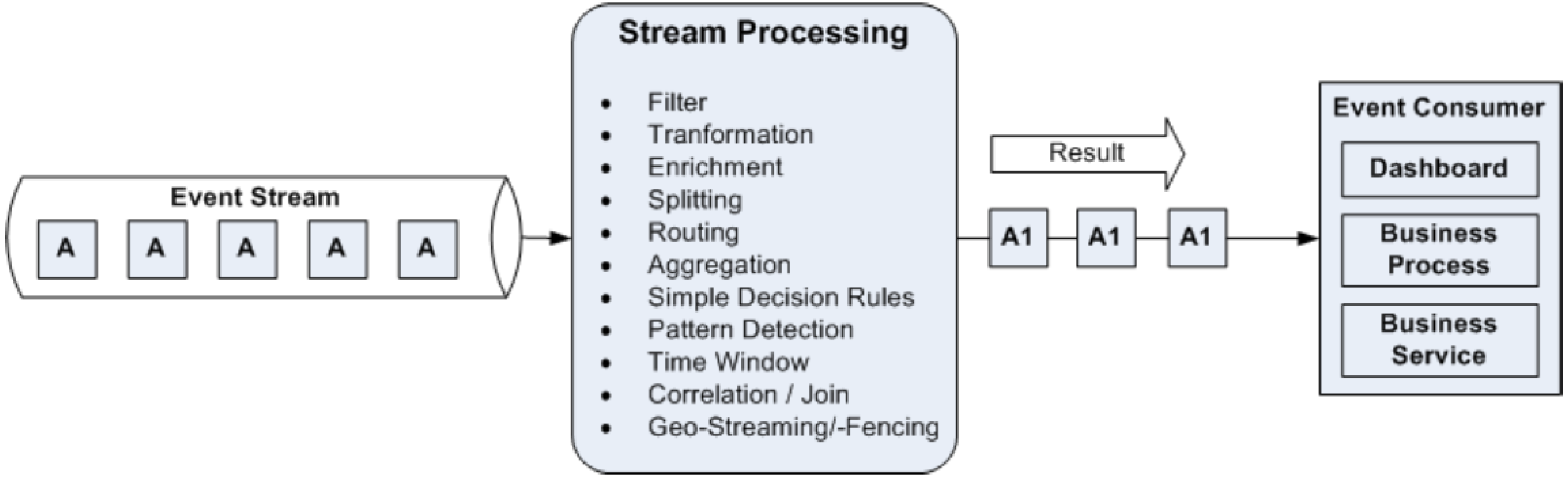
- If the Execution Time Duration exceeds the Time Period
  - A) the successive execution (Batch 1.2) is overlapped (yellow)
    - If it happens systematically: the number of tasks grow indefinitely consuming all resources → until crash
  - B) the successive execution (Batch 1.2) is canceled to wait for the next one (Batch 1.3, green),
    - skipping the second execution, Batch 1.2

Stream Processing

- paradigma di programmazione parallela
  - Detto anche di real time processing
- Lavora con:
  - Dati parziali e non su tutto l'insieme dei dati.
  - Per esempio: valutare il contenuto spettrale di un segnale:
    - Su tutto il segnale
    - Su una finestra temporale di 30 secondi, per ogni secondo un nuovo valore, small delay/latency, (but present , see pipeline)
- L'attivazione del processo corrisponde spesso all'arrivo del dato nello stream, nella pipeline,
  - si ha sincronizzazione e comunicazione in un solo colpo.
  - Tipicamente una sequenza di azioni semplici attivate da
    - l'arrivo delle condizioni e dei dati per calcolare i risultati
    - ed in modo asincrono
    - tipicamente senza avere necessità di memorizzare il dato
  - → elimina alcuni problemi della programmazione parallela.

Stream Processing

- Example on languages
  - SISAL (Streams and Iteration in a Single Assignment Language)
  - CUDA (Compute Unified Device Architecture) for NVIDIA GPU, SIMD
- Applications:
  - IOT, media recognition, log processing, social media enrichment, indexing,



Main Purpose

- To store a large amount of data, **big data**, and they can be **structured and un-structured**, several different kind of data:
  - **Direct Data**: Time series, geolocated data, events, shapes, measures, social media posts, video, files, logs, etc.
    - Most of them may have multiple features, e.g.: geolocated events with shape
  - **Derived Data**: predictions, typical trends, trajectories, flows, heatmaps, 3D reconstructions, traffic reconstruction, planning, simulations, etc.
- for **exploiting them** for producing:
  - Deductions, hints, early warning,
  - Derived Data as well, in real time

Main Functions

- **Data Extraction:**
  - gathering, harvesting, ingestion, reception in push, ….
- **Data Transformation:**
  - Adaptation, mapping, formatting, conversion, enrich
  - Cleaning or leaving as it is
- **Data Loading** and Refreshing: saving in the storage
  - As it is, converted and ready to use, etc.
- **Data Usage:**
  - As it is from Storage (faster, more rigid schema, higher volume in access)
  - Transformed on the fly (slower, more flexible, moderate volume in access)

DW vs DL

- ETL-Usage vs ELT-Usage
- DW:
  - – More complex data ingestion
    - Simplify data ingestion for sporadically used data
  - – Faster date usage unforeseen patterns/combinations
- DL:
  - – Faster data ingestion
  - – More complex data rendering since all combination are unplanned
    - Prepare data rendering for well known patterns

Parameters	Data Lake	Data Warehouse
Storage	In the data lake, all data is kept irrespective of the source and its structure. Data is kept in its raw form. It is only transformed when it is ready to be used.	A data warehouse will consist of data that is extracted from transactional systems or data which consists of quantitative metrics with their attributes. The data is cleaned and transformed
History	Big data technologies used in data lakes is relatively new.	Data warehouse concept, unlike big data, had been used for decades.
Data Capturing	Captures all kinds of data and structures, semi-structured and unstructured in their original form from source systems.	Captures structured information and organizes them in schemas as defined for data warehouse purposes
Data Timeline	can retain all data. This includes not only the data that is in use but also data that it might use in the future. Also, data is kept for all time, to go back in time and do an analysis.	In the data warehouse development process, significant time is spent on analyzing various data sources.
Users	ideal for the users who indulge in deep analysis. Such users include data scientists who need advanced analytical tools with capabilities such as predictive modeling and statistical analysis.	The data warehouse is ideal for operational users because of being well structured, easy to use and understand.
Storage Costs	Data storing in big data technologies are relatively inexpensive then storing data in a data warehouse.	Storing data in Data warehouse is costlier and time-consuming.

https://www.guru99.com/data-lake-vs-data-warehouse.html

Parameters	Data Lake	Data Warehouse
Task	can contain all data and data types; it empowers users to access data prior the process of transformed, cleansed and structured.	Data warehouses can provide insights into pre-defined questions for pre-defined data types.
Processing time	empower users to access data before it has been transformed, cleansed and structured. Thus, it allows users to get to their result more quickly compares to the traditional data warehouse.	Data warehouses offer insights into pre-defined questions for pre-defined data types. So, any changes to the data warehouse needed more time.
Position of Schema	Typically, the schema is defined after data is stored. This offers high agility and ease of data capture but requires work at the end of the process	Typically schema is defined before data is stored. Requires work at the start of the process, but offers performance, security, and integration.
Data processing	use of the ELT (**Extract Load Transform**) process.	Data warehouse uses a traditional ETL (**Extract Transform Load**) process.
Complain	Data is kept in its raw form. It is only transformed when it is ready to be used.	The chief complaint against data warehouses is the inability, or the problem faced when trying to make change in in them.
Key Benefits	They integrate different types of data to come up with entirely new questions as these users not likely to use data warehouses because they may need to go beyond its capabilities.	Most users in an organization are operational. These type of users only care about reports and key performance metrics.

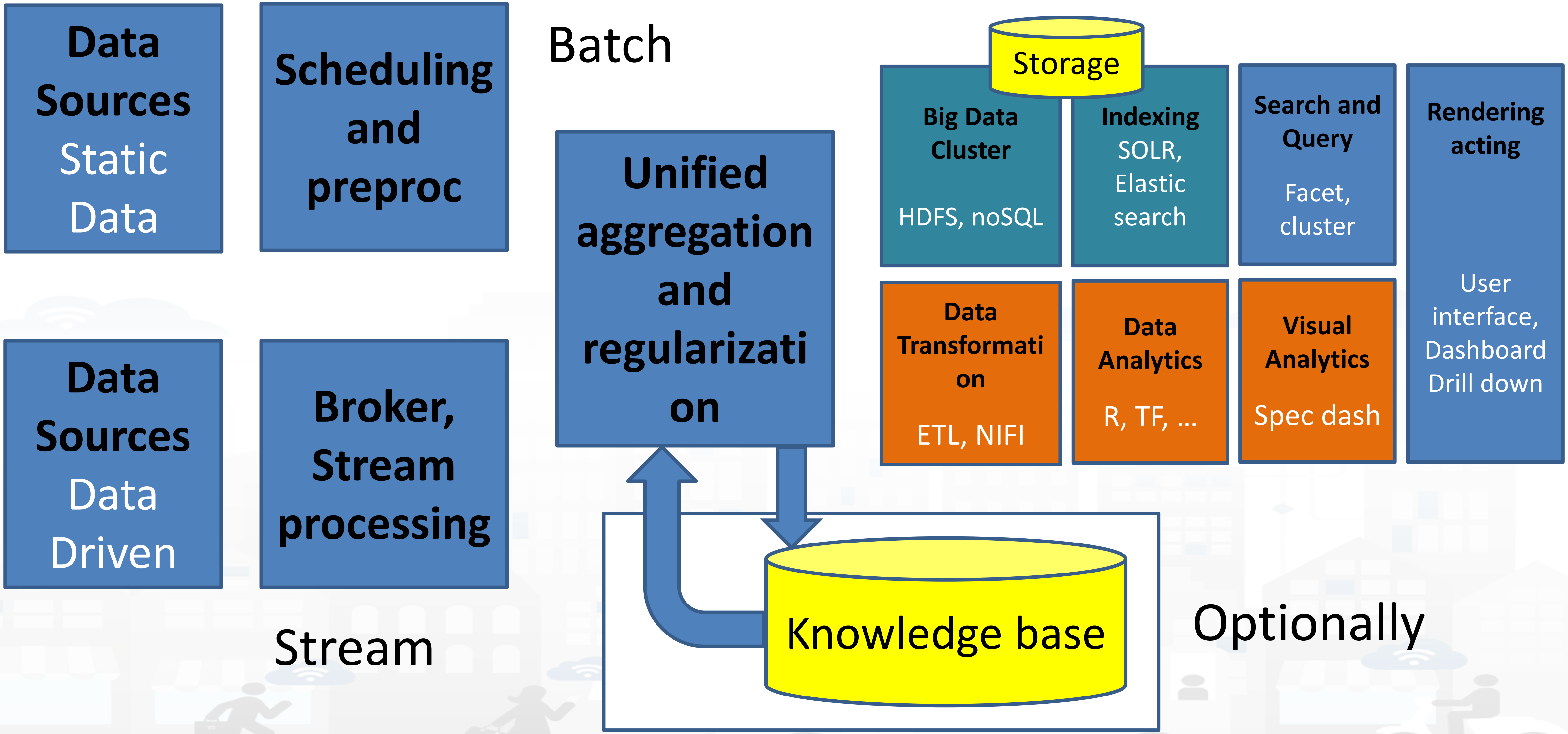
https://www.guru99.com/data-lake-vs-data-warehouse.html

Pros and Cons

- **Data Lake**
  - Original data preserved, structured and unstructured
  - Lower costs of ingestion
  - Lower performance in the usage
  - Security: possible control at source
  - More difficult to extend in usage, simpler in storage
  - More scientist oriented
    - Moderated results in access

- **Data Warehouse**
  - Original data transformed and prepared for mainly structured or semi-structured
  - Higher cost of ingestion
  - Higher performance in the usage
  - Security: Control on organized data
  - More difficult to extend in storage, simpler in usage
  - More Business and Purpose Oriented
    - Large volume of accesses

DWH



Data Sources Static Data

Scheduling and preproc

Batch

Data Sources Data Driven

Broker, Stream processing

Stream

Unified aggregation and regularization

Knowledge base

Storage

Big Data Cluster

HDFS, noSQL

Indexing SOLR, Elastic search

Search and Query

Facet, cluster

Rendering acting

User interface, Dashboard Drill down

Data Transformation

ETL, NIFI

Data Analytics

R, TF, …

Visual Analytics

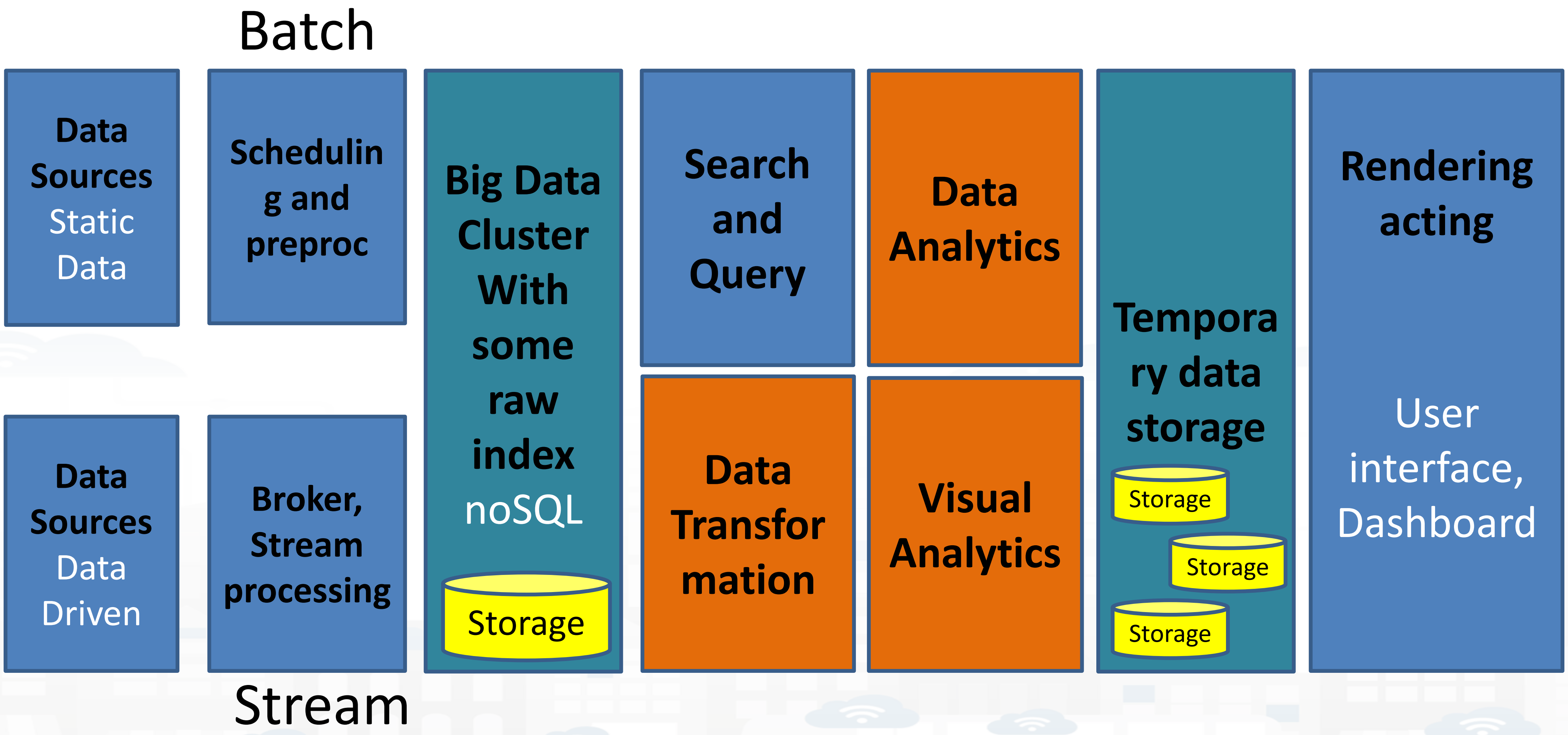
Spec dash

Optionally

Top Cloud Data Warehouses at a Glance

	Amazon Redshift	Microsoft Azure Synapse	Google BigQuery	Snowflake Cloud Data Platform
Initial Release	2012	2016	2010	2014
Separates Storage and Compute	No	Yes	Yes	Yes
Multi-Cloud	No	No	No	Yes
Query Language	Amazon Redshift SQL	TSQL	Standard SQL 2011 & BigQuery SQL	Snowflake SQL
Elasticity	Yes - Manual	Yes – Manual and Automatic	Yes – Automatic	Yes – Automatic
MPP	Yes	Yes	Yes	Yes
Columnar	Yes	Yes	Yes	Yes
Foreign Keys	Yes	Yes	No	Yes
Transaction	ACID	ACID	ACID	ACID
Concurrency	Yes	Yes	Yes	Yes
Durability	Yes	Yes	Yes	Yes
Automation	No	No	No	No
Website	Link	Link	Link	Link
Free Trial	Yes	Yes	Yes	Yes

DL



Batch

Data Sources Static Data

Scheduling and preproc

Data Sources Data Driven

Broker, Stream processing

Stream

Big Data Cluster With some raw index noSQL

Storage

Search and Query

Data Transformation

Data Analytics

Visual Analytics

Temporary data storage

Storage

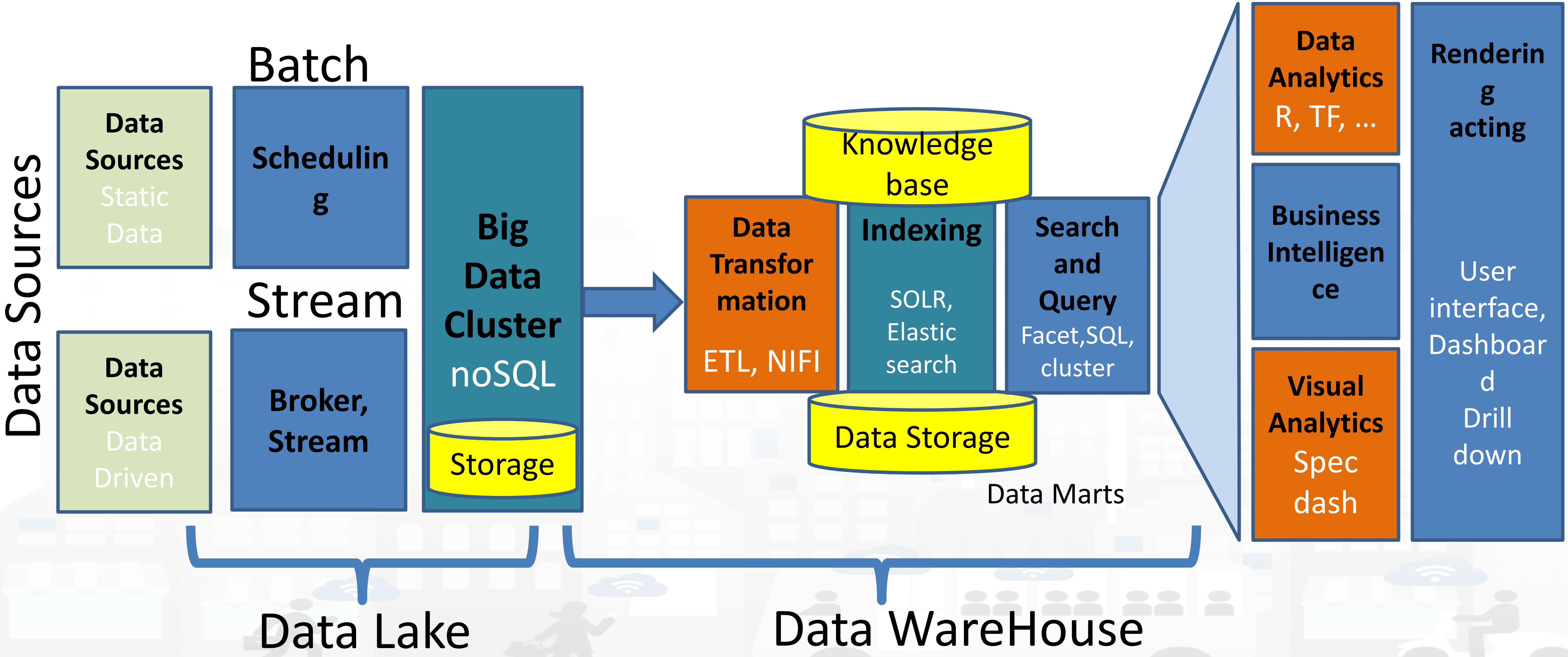
Storage

Storage

Rendering acting

User interface, Dashboard

Combined Solutions



Data Sources

Data Sources Static Data

Data Sources Data Driven

Batch

Scheduling

Stream

Broker, Stream

Big Data Cluster noSQL

Storage

Knowledge base

Data Transformation ETL, NIFI

Indexing SOLR, Elastic search

Search and Query Facet,SQL, cluster

Data Storage

Data Marts

Data Analytics R, TF, …

Business Intelligence

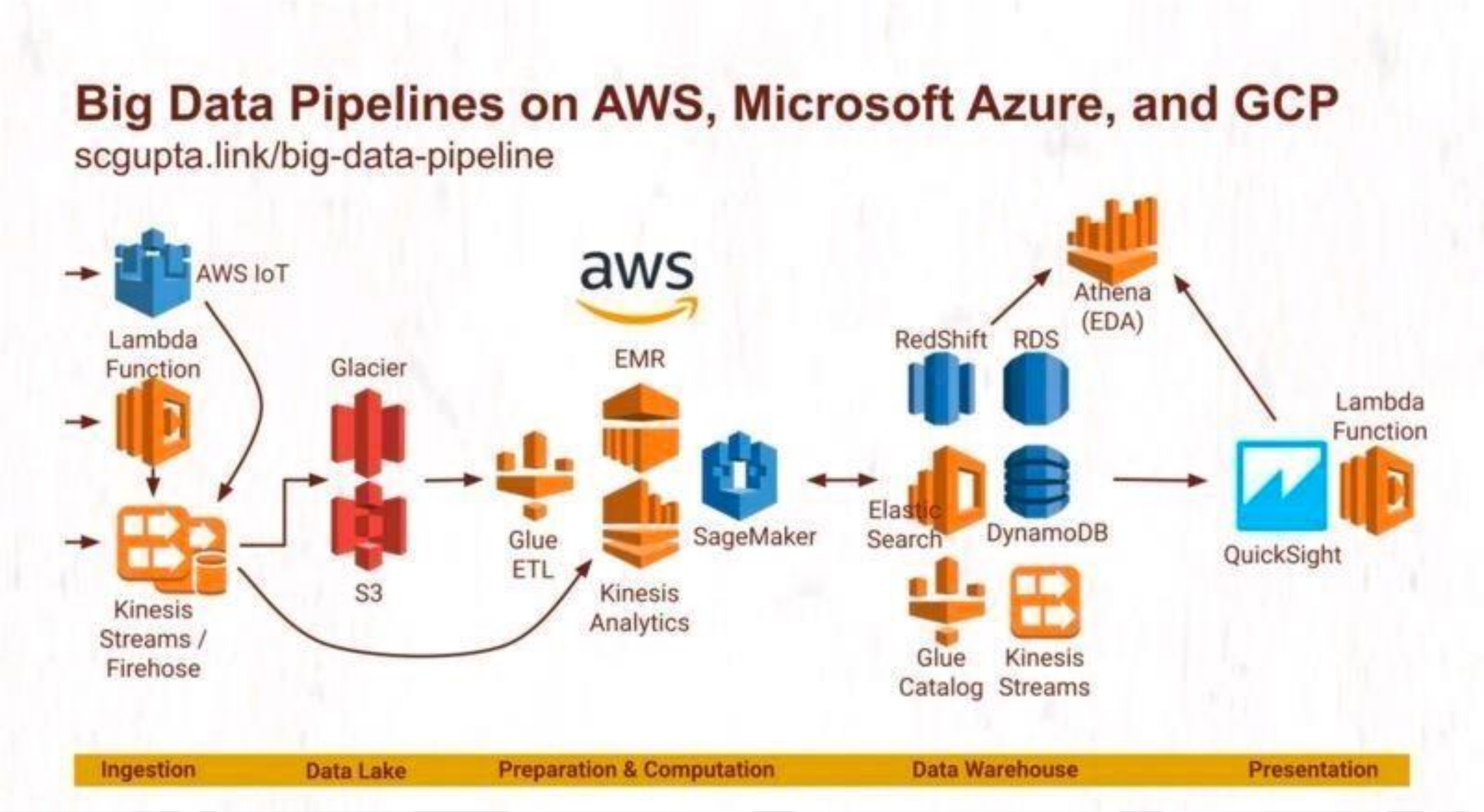
Visual Analytics Spec dash

Rendering acting

User interface, Dashboard Drill down

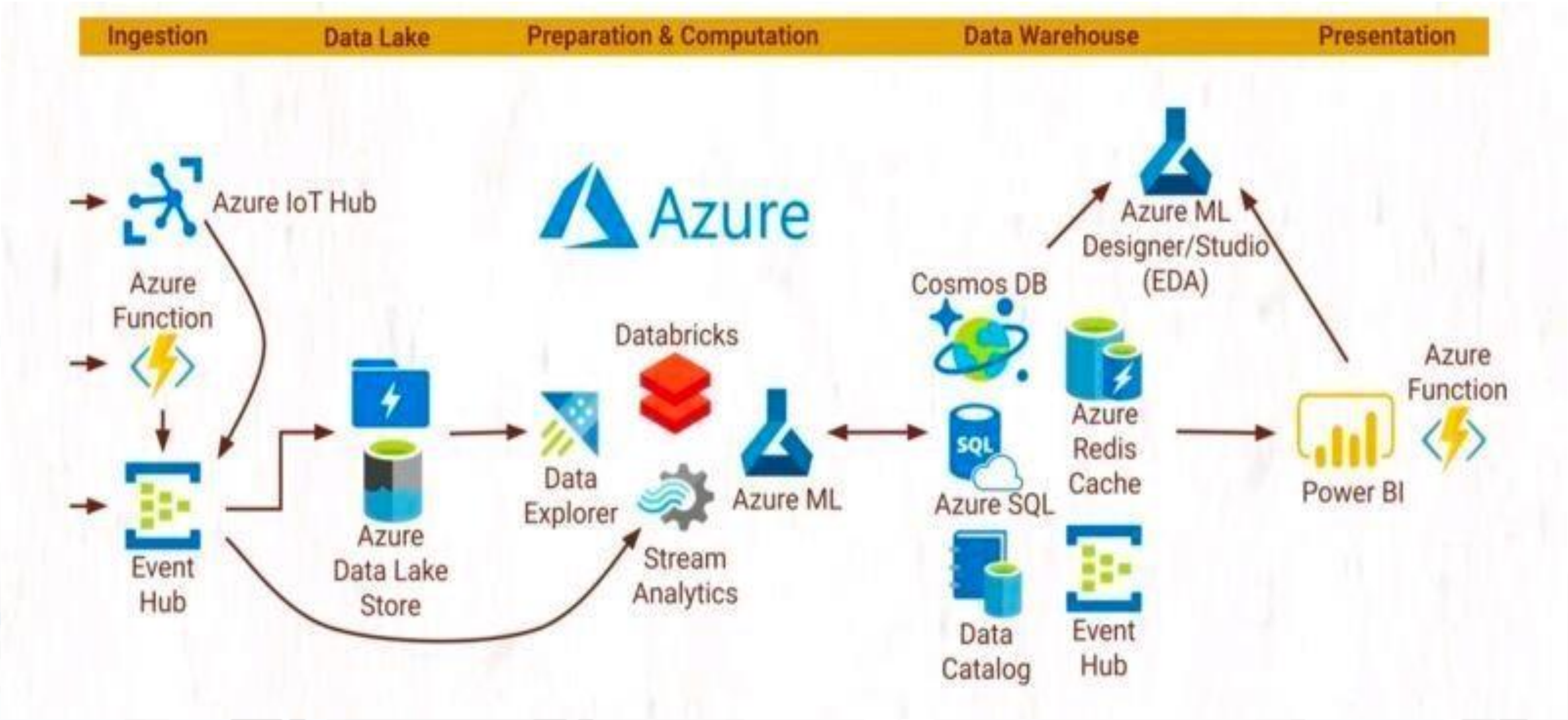
Data Lake

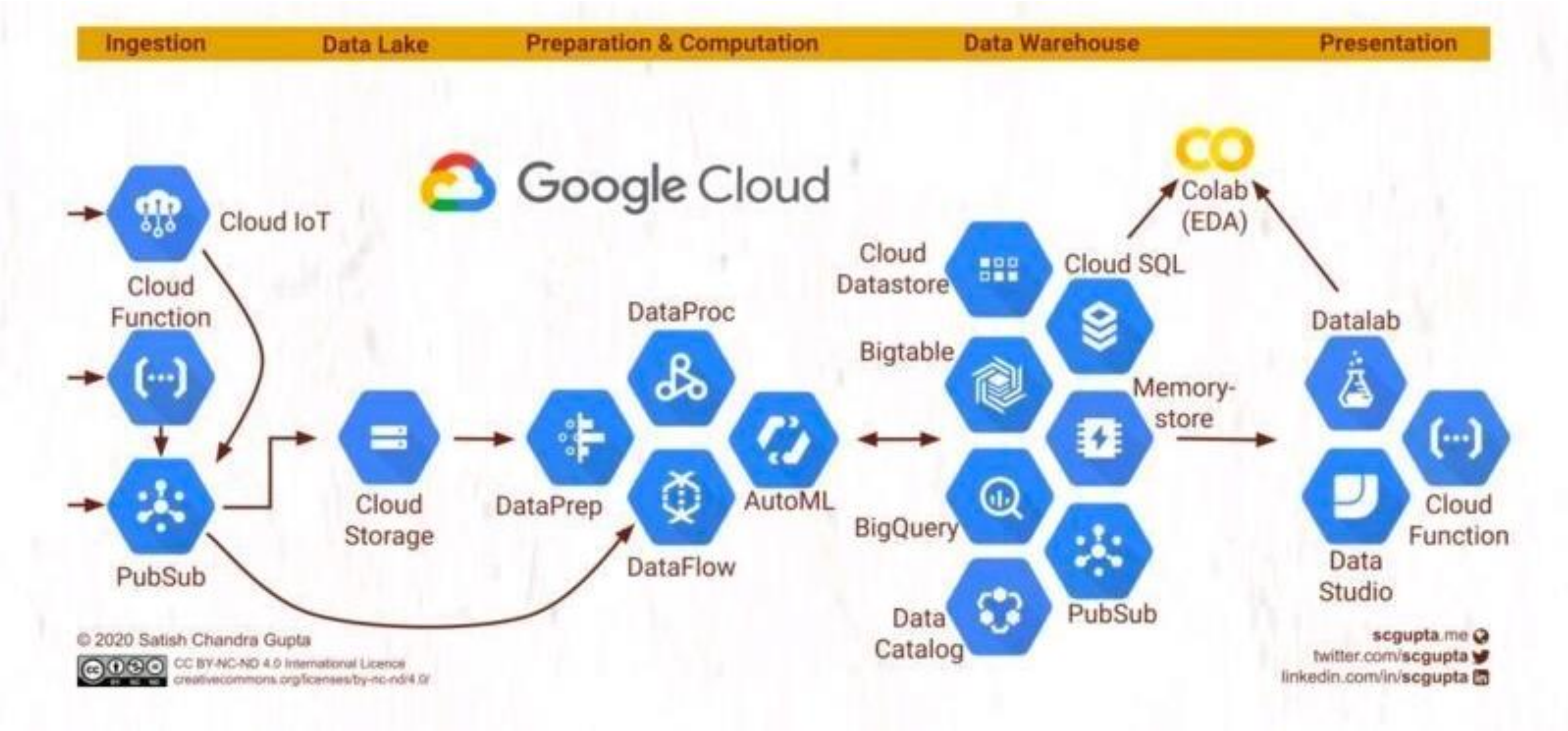
Data WareHouse



Big Data Pipelines on AWS, Microsoft Azure, and GCP

scgupta.link/big-data-pipeline


Ingestion
Data Lake
Preparation & Computation
Data Warehouse
Presentation
Azure
Azure IoT Hub
Azure Function
Event Hub
Azure Data Lake Store
Data Explorer
Databricks
Stream Analytics
Azure ML
Cosmos DB
Azure SQL
Data Catalog
Azure Redis Cache
Event Hub
Azure ML Designer/Studio (EDA)
Power BI
Azure Function


Ingestion
Data Lake
Preparation & Computation
Data Warehouse
Presentation
Google Cloud
Cloud IoT
Cloud Function
PubSub
Cloud Storage
DataPrep
DataProc
DataFlow
AutoML
Cloud Datastore
Bigtable
BigQuery
Data Catalog
Cloud SQL
Memory-store
PubSub
Colab (EDA)
Datalab
Data Studio
Cloud Function
scgupta.me
twitter.com/scgupta
linkedin.com/in/scgupta