CMC SYSTEM INTEGRATION

Towards the digital future

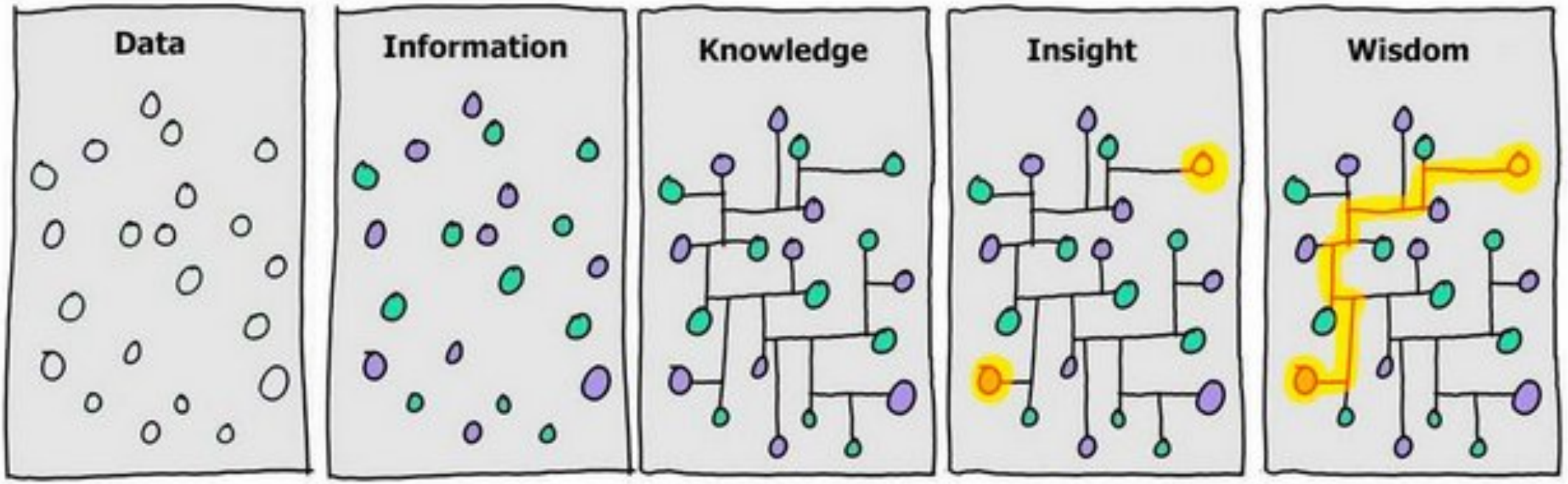
Big Data Reference Architecture on AWS

Author: Thanh Luong

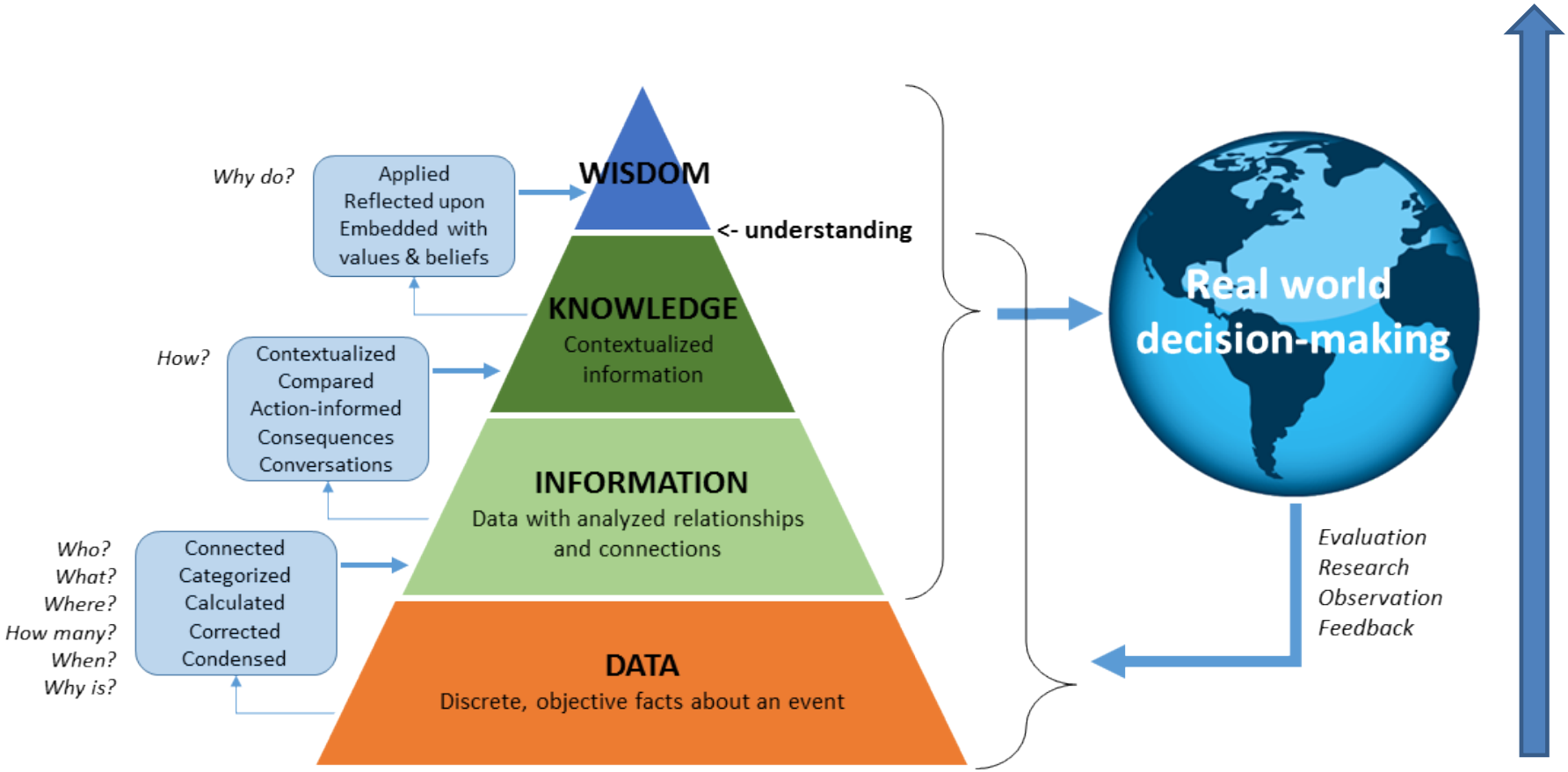
Agenda

- Introduction to Data Mining & Data Science
- Data Driven Architecture for Enterprise
- AWS Practice
- Key takeaway

Data- Information – Knowledge - Wisdom

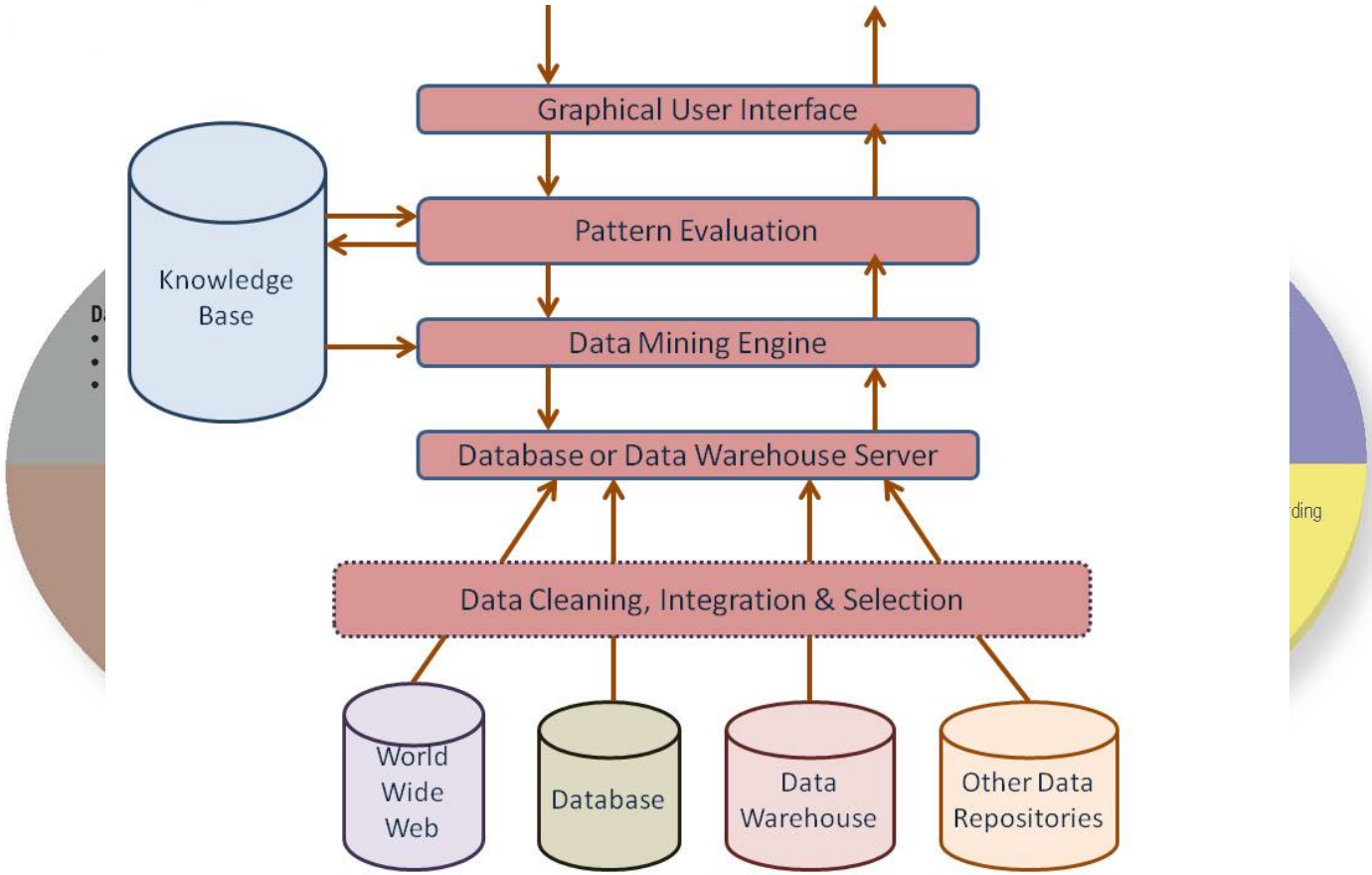


Data, Information and Knowledge

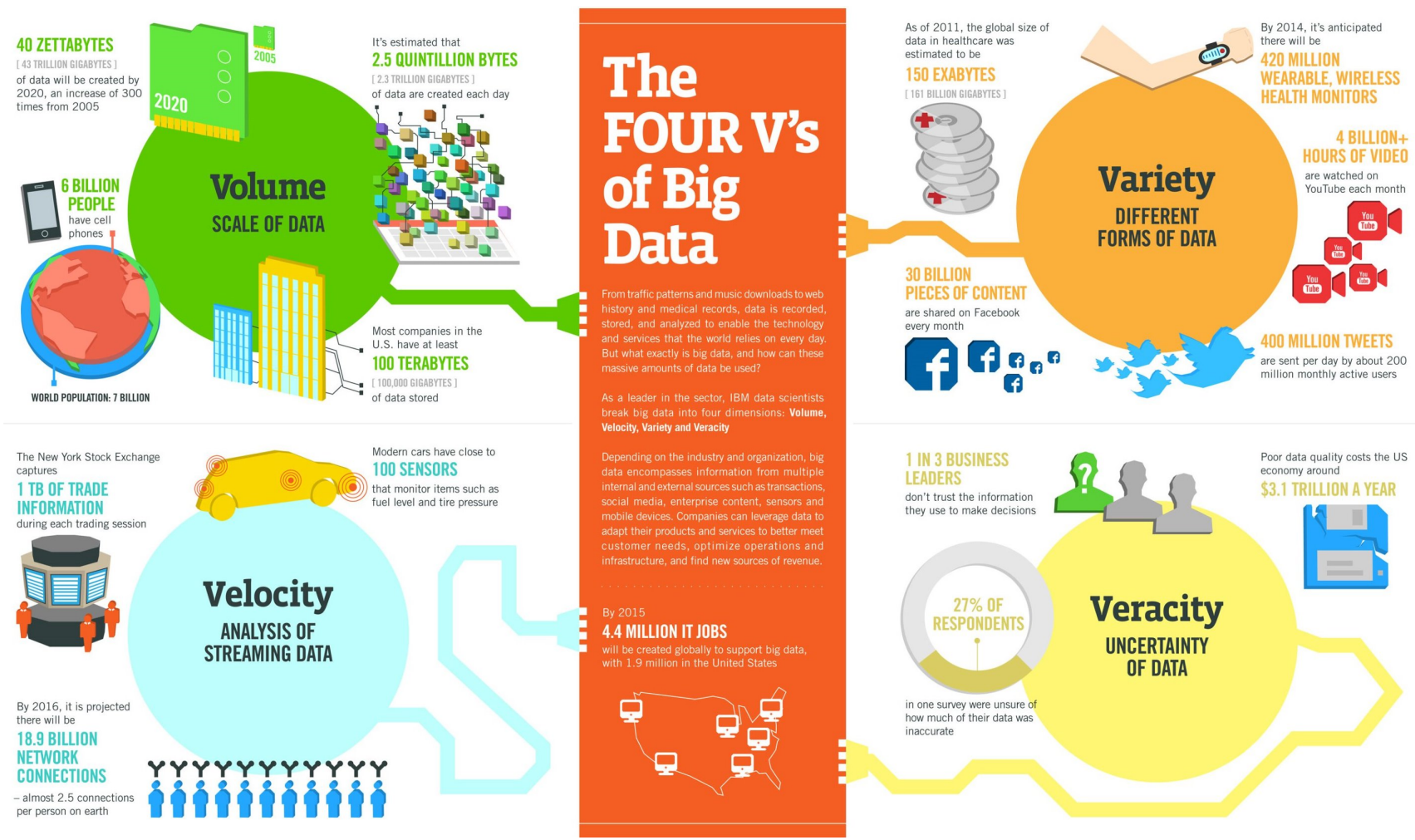


- Statistic
- Machine Learning
- Data Mining
- Data Science

Data Driven Implementation for Enterprise



Big Data


40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005
2005
2020
6 BILLION PEOPLE
have cell phones
WORLD POPULATION: 7 BILLION
Volume
SCALE OF DATA
It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day
Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored
The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session
Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure
Velocity
ANALYSIS OF STREAMING DATA
By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth
The FOUR V's of Big Data
From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?
As a leader in the sector, IBM data scientists break big data into four dimensions: Volume, Velocity, Variety and Veracity
Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.
By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States
As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]
30 BILLION PIECES OF CONTENT
are shared on Facebook every month
Variety
DIFFERENT FORMS OF DATA
By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS
4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month
400 MILLION TWEETS
are sent per day by about 200 million monthly active users
1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions
27% OF RESPONDENTS
in one survey were unsure of how much of their data was inaccurate
Veracity
UNCERTAINTY OF DATA
Poor data quality costs the US economy around
$3.1 TRILLION A YEAR

Data types & Mining Methods

- Data types and models
  - Flat data tables
  - Relational databases
  - Temporal & spatial data
  - Transactional databases
  - Multimedia data
  - Genome databases
  - Materials science data
  - Textual data
  - Web data
  - Etc.

- Mining tasks and method
  - Classification / Prediction
    - ✓ Decision trees
    - ✓ Bayesian classification
    - ✓ Neural networks
    - ✓ Rule induction
    - ✓ Support vector machine (SVM)
    - ✓ Hidden Markov Model
    - ✓ Etc
  - Description
    - ✓ Association analysis
    - ✓ Clustering
    - ✓ Summarization
    - ✓ Etc.

Data types

- Symbolic
  - Indexing
  - Binary
  - Boolean
  - Nominal
  - Ordinal
- Numeric
  - Integer
  - Continuous

- Structured vs Unstructured data
- Semi-structured data
- Supervised vs Unsupervised data

Data Mining techniques

- Supervised Learning

(predictive ability based on past data)

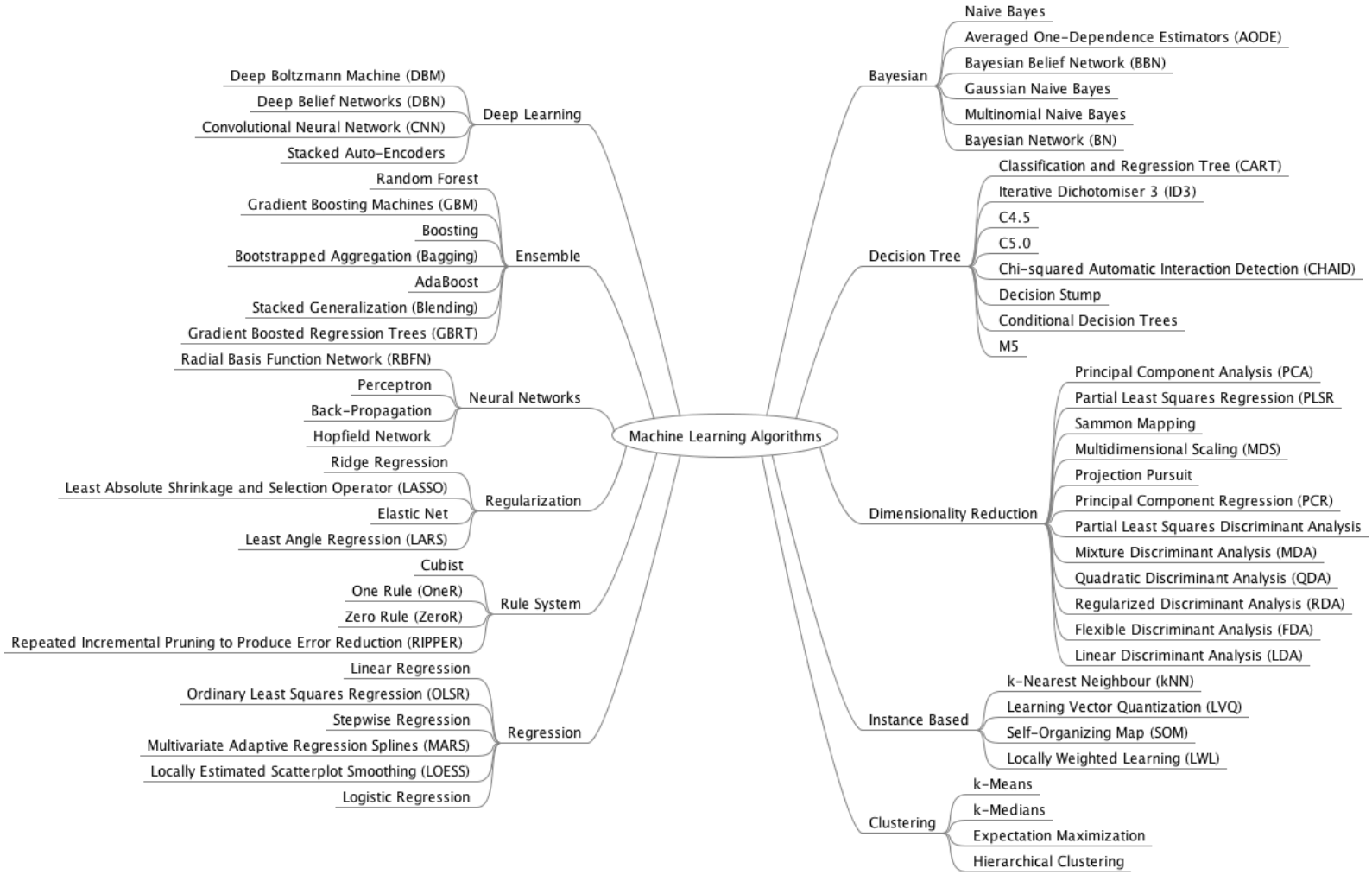
  - Classification Statistics
  - Decision Trees
  - Regression
  - Artificial Neural Networks (ANN)
  - Classification machine learning

- Unsupervised Learning

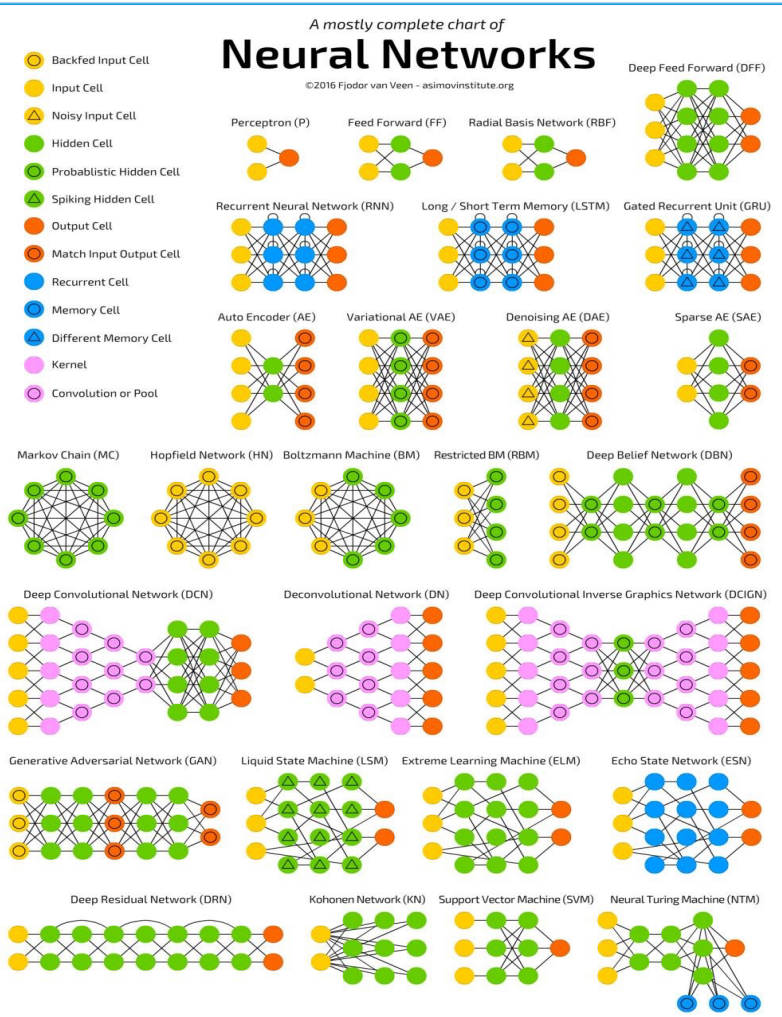
(Exploratory analysis to discover patterns)

  - Clustering Analysis
  - Association Rules

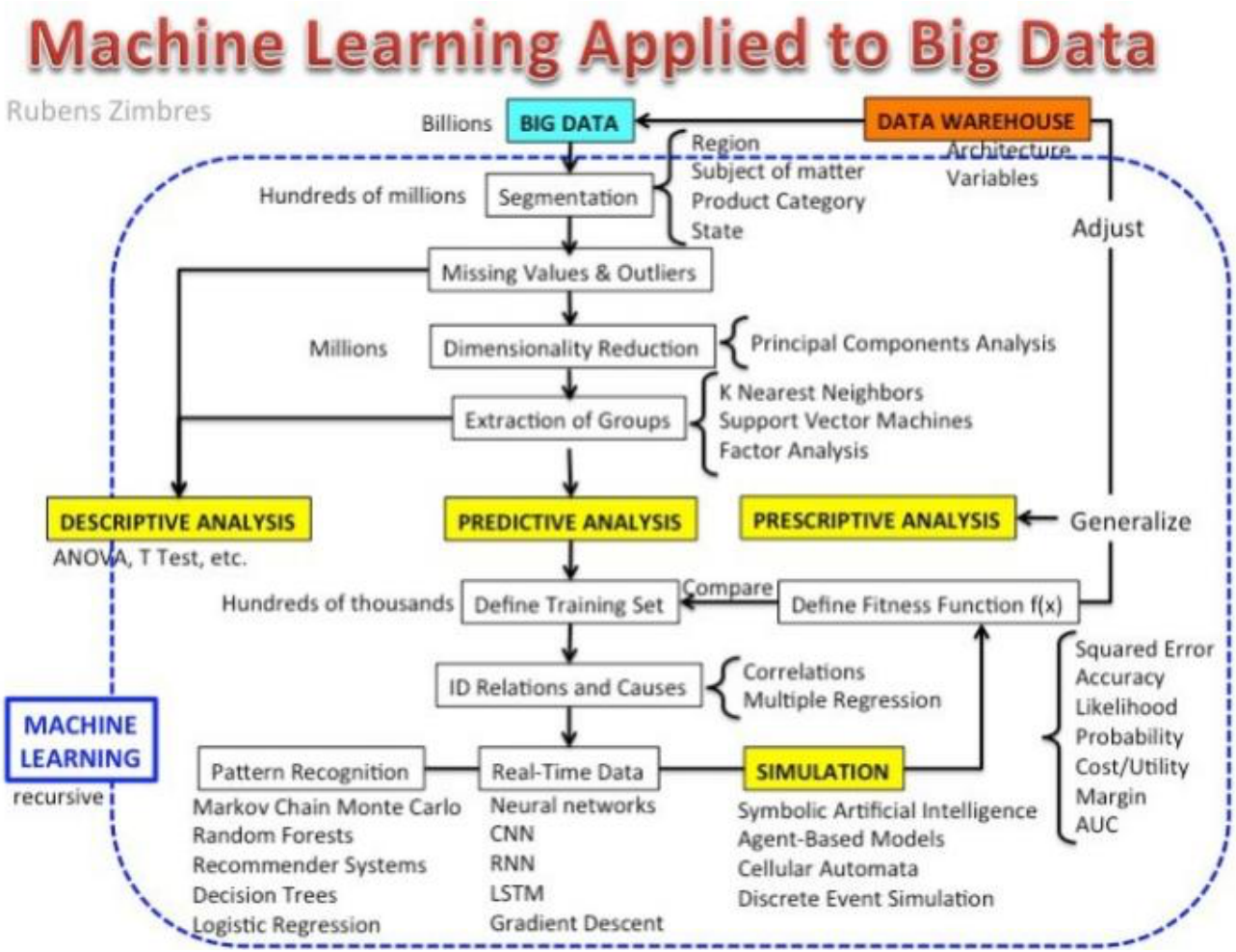
Machine Learning



Neural Networks



From Kdnuggets

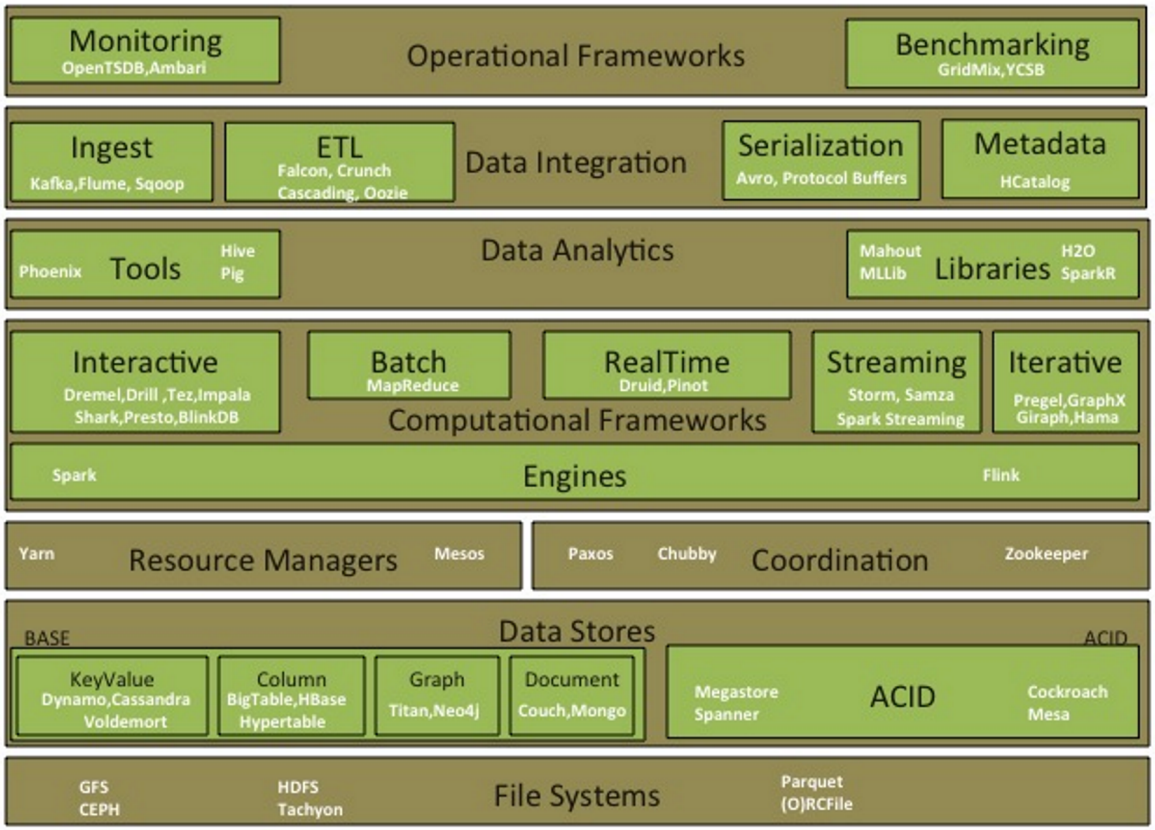


CMC SYSTEM INTEGRATION
Towards the digital future

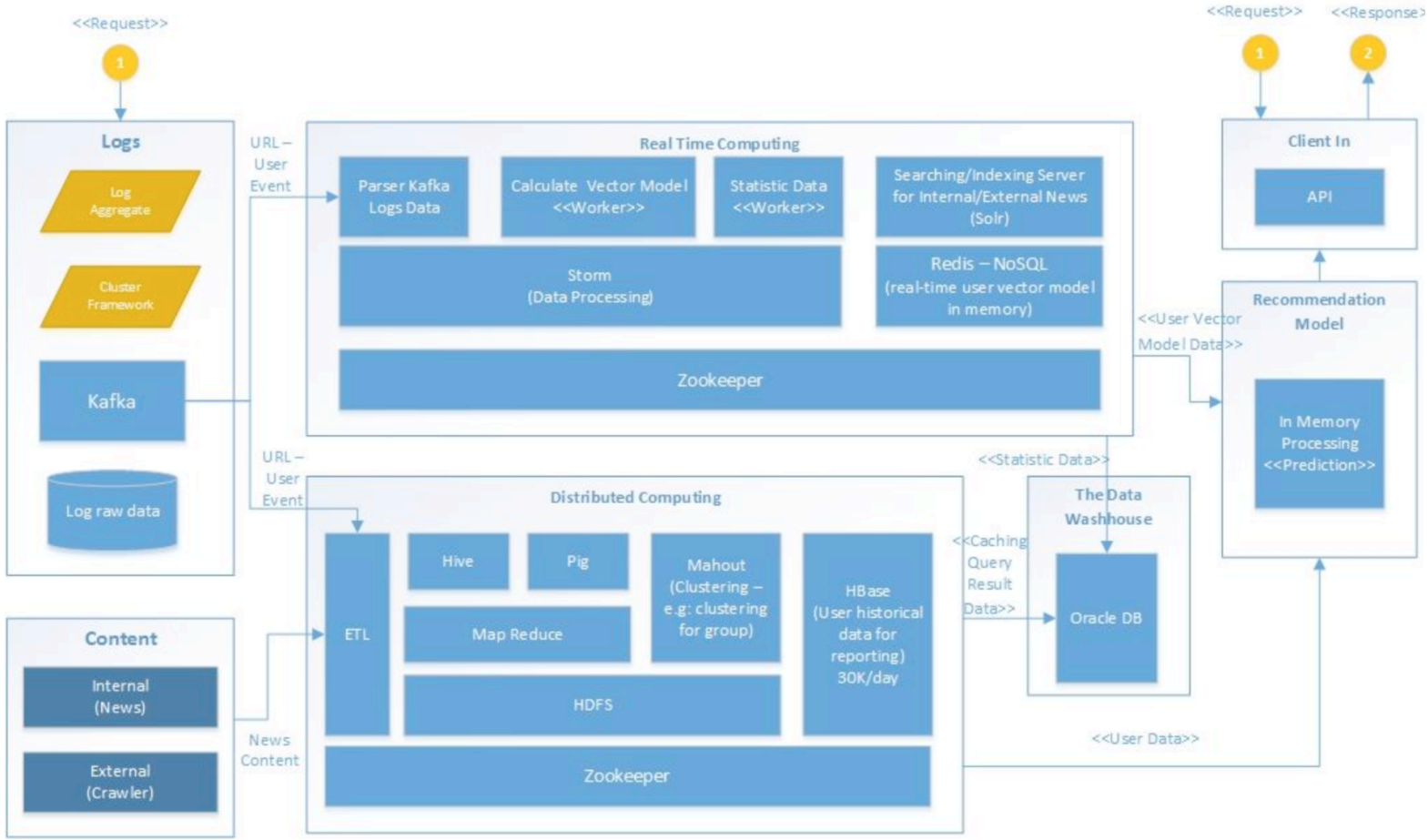
Data Driven Architecture for Enterprise

Too many tools of Big Data

Reference Architecture- Open Source Tools



Data Platform Reference Architecture–Open Source

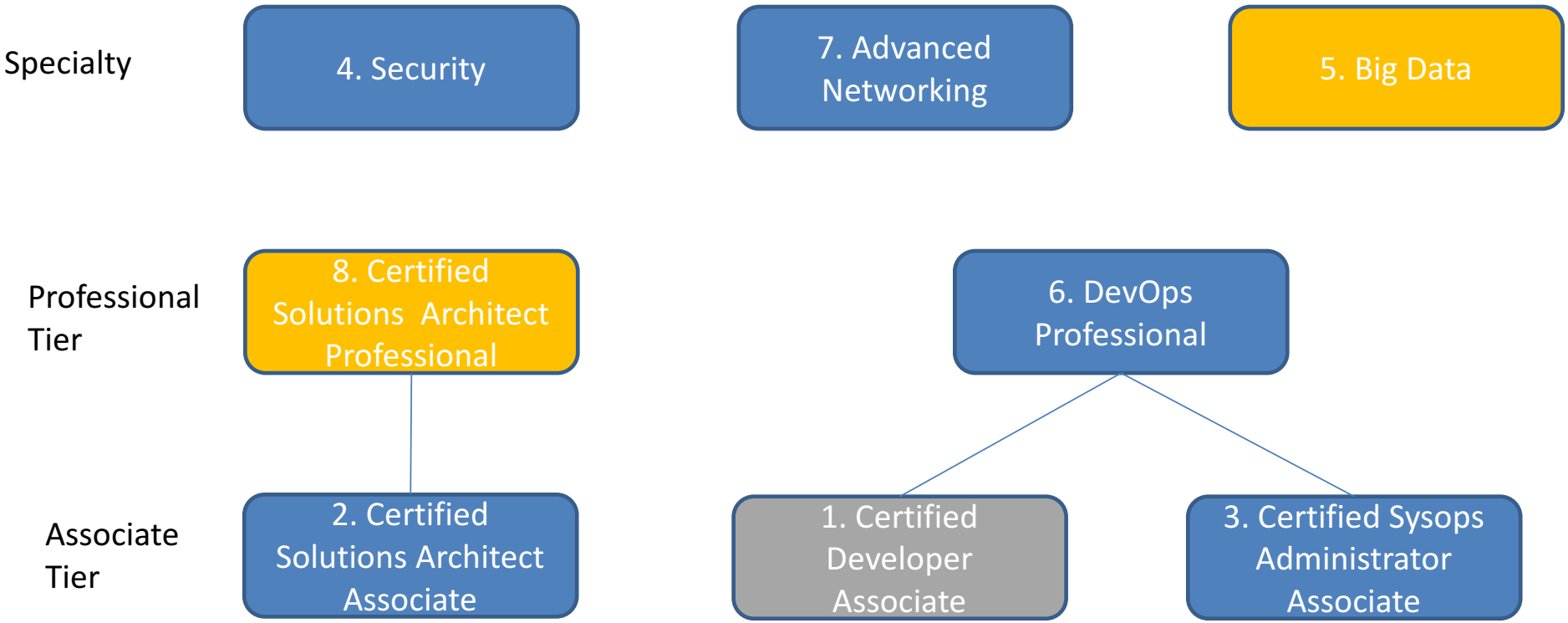


CMC SYSTEM INTEGRATION

Towards the digital future

AWS Big Data Reference Architecture

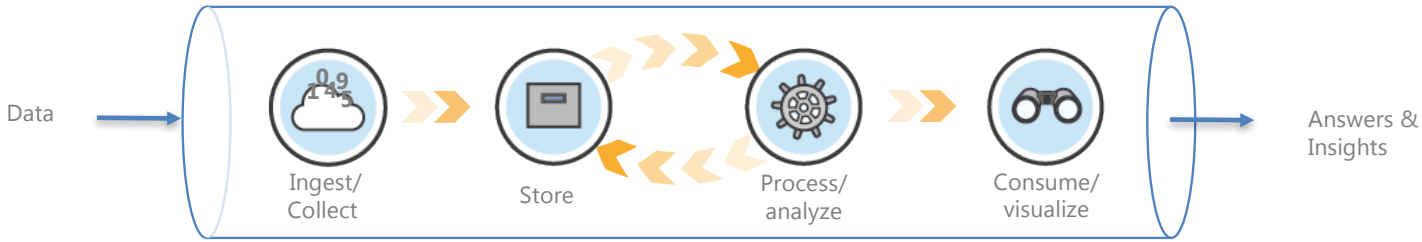
Amazon Web Services – Technical Skills



Architectural Principles - AWS

- Build decoupled systems
  - **Data → Store → Process → Store → Analyze → Answers**
- Use the right tool for the job
  - Data structure, latency, throughput, access patterns
- Leverage AWS managed services
  - Scalable/elastic, available, reliable, secure, no/low admin
- Use Lambda architecture ideas
  - Immutable (append-only) log, batch/speed/serving layer
- Be cost-conscious
  - Big data ≠ big cost

Simplify Big Data Processing

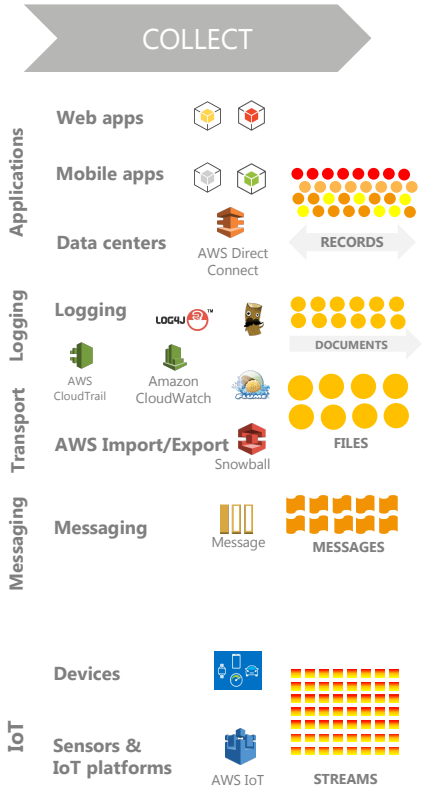


CMC SYSTEM INTEGRATION

Towards the digital future

Ingest/Collect

Data Ingestion & Collection



Types of Data

In-memory data structures
Database records
→ Transactions

Search documents
Log files
→ Files

Messages
→ Events

Data streams
→ Events

Data Characteristics: Hot, Warm, Cold

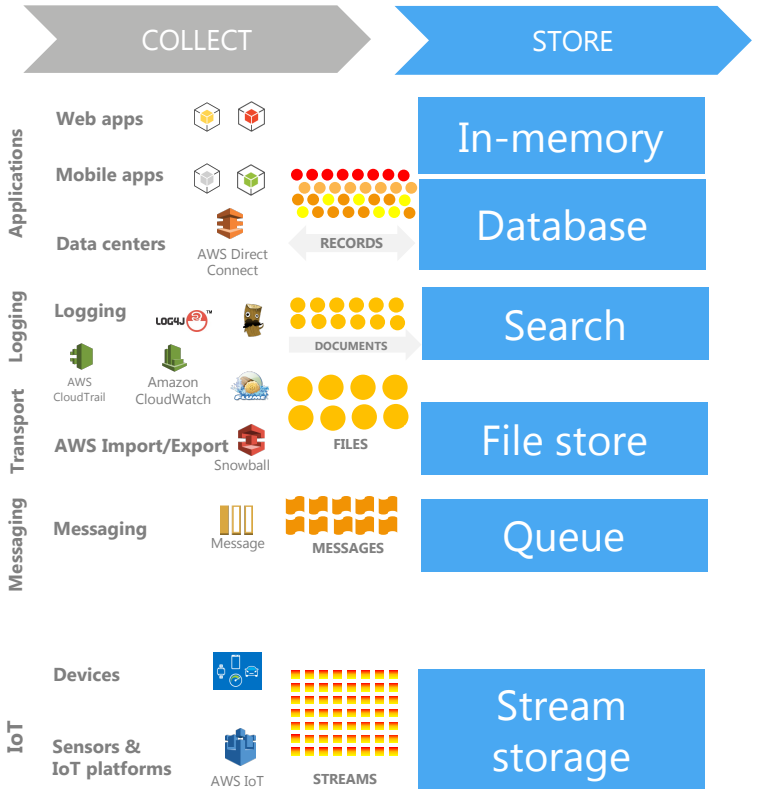
	Hot	Warm	Cold
Volume	MB–GB	GB–TB	PB–EB
Item size	B–KB	KB–MB	KB–TB
Latency	ms	ms, sec	min, hrs
Durability	Low–high	High	Very high
Request rate	Very high	High	Low
Cost/GB	$$-$	$-¢¢	¢

CMC SYSTEM INTEGRATION

Towards the digital future

Store

Data Store



Types of Data Stores

Caches, data structure servers

SQL & NoSQL databases

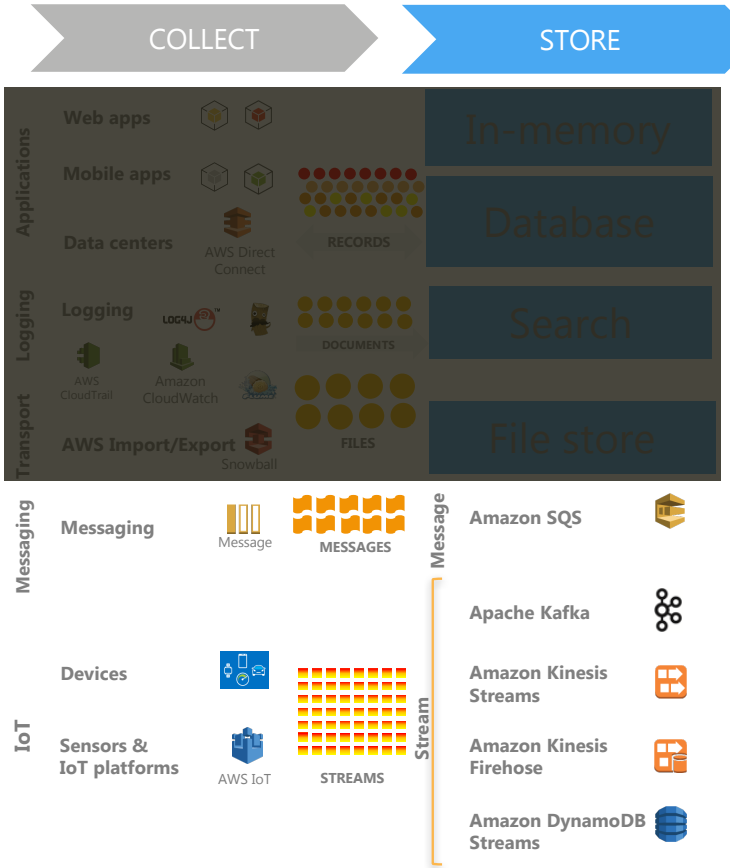
Search engines

File systems

Message queues

Pub/sub message queues

Data Store



Message & Stream Storage

Amazon SQS

- Managed message queue service

Apache Kafka

- High throughput distributed streaming platform

Amazon Kinesis Streams

- Managed stream storage + processing

Amazon Kinesis Firehose

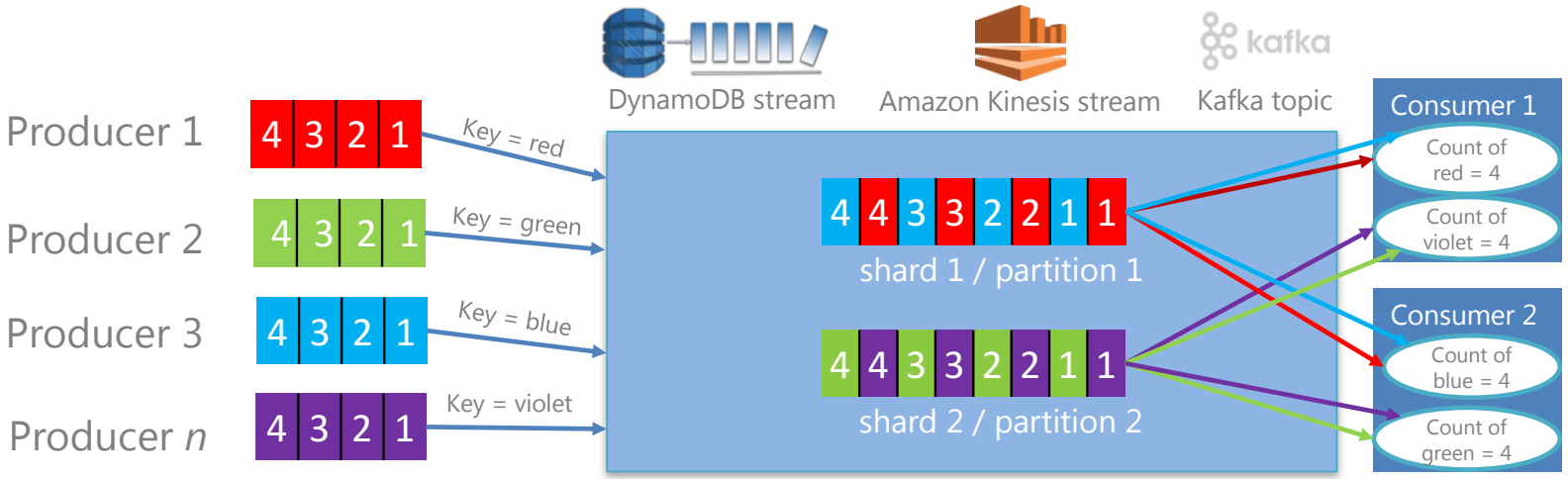
- Managed data delivery

Amazon DynamoDB

- Managed NoSQL database
- Tables can be stream-enabled

Why Stream Storage?

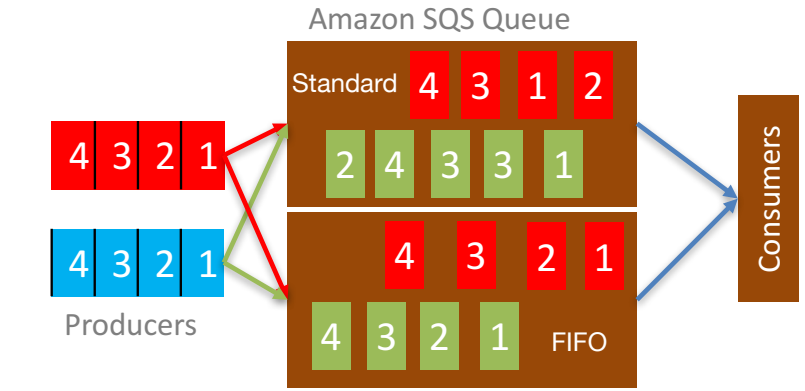
- Decouple producers & consumers
- Persistent buffer
- Collect multiple streams
- Preserve client ordering
- Parallel consumption
- Streaming MapReduce

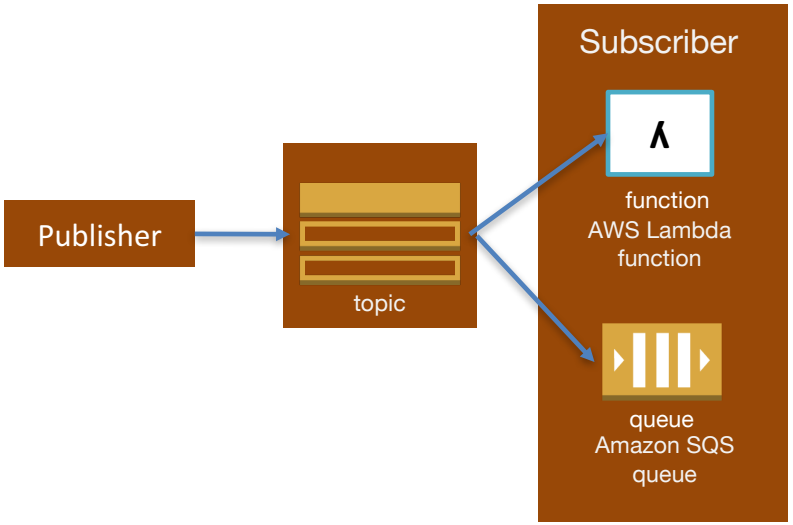


AWS SQS

- Decouple producers & consumers
- Persistent buffer
- Collect multiple streams

- **No** client ordering (Standard)
  - FIFO queue preserves client ordering
- **No** streaming MapReduce
- **No** parallel consumption
  - Amazon SNS can publish to multiple SNS subscribers (queues or λ functions)

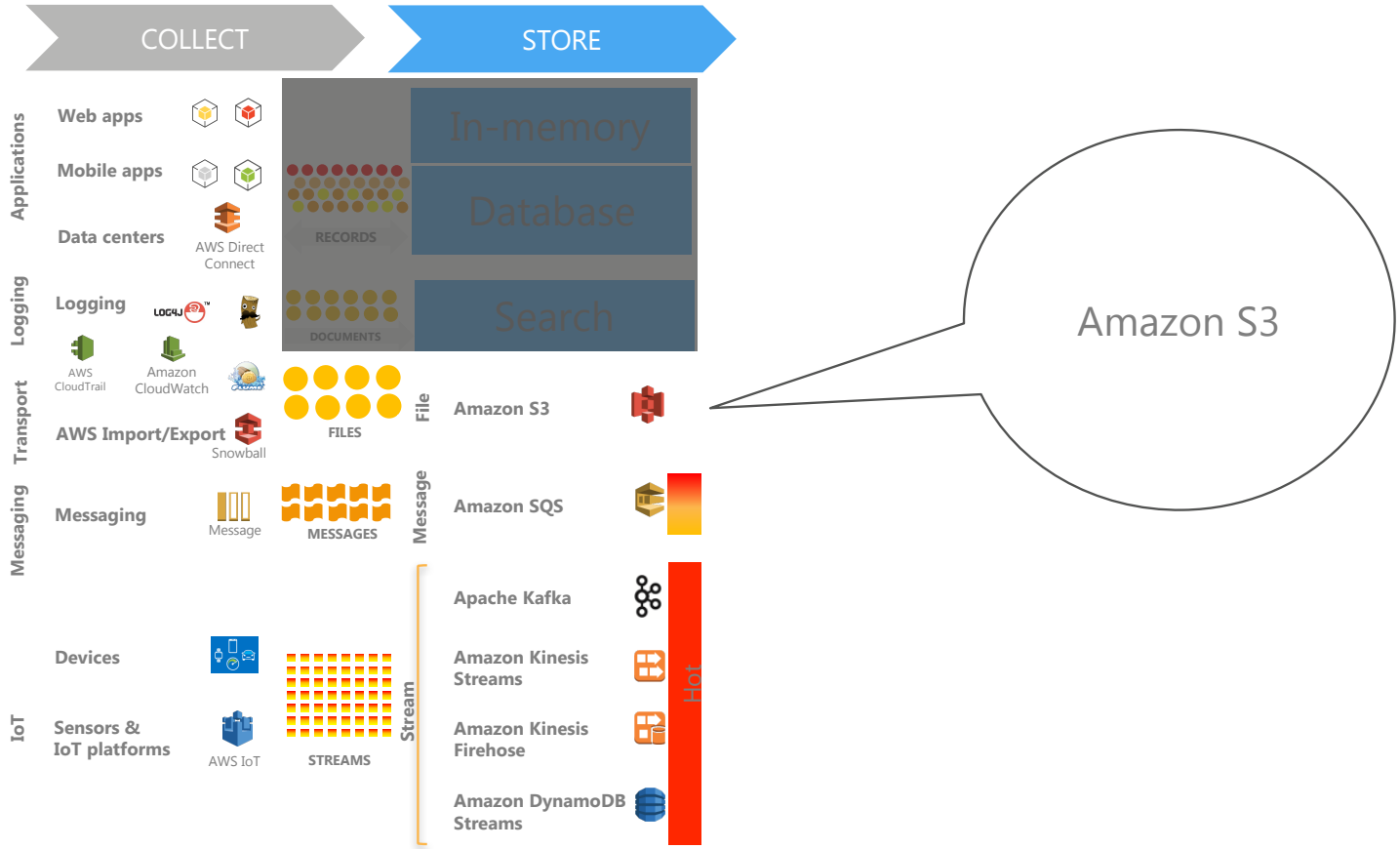




Which Stream/Message Storage should we use?

	Amazon DynamoDB Streams	Amazon Kinesis Streams	Amazon Kinesis Firehose	Apache Kafka	Amazon SQS (Standard)	Amazon SQS (FIFO)
AWS managed	Yes	Yes	Yes	No	Yes	Yes
Guaranteed ordering	Yes	Yes	No	Yes	No	Yes
Delivery (deduping)	Exactly-once	At-least-once	At-least-once	At-least-once	At-least-once	Exactly-once
Data retention period	24 hours	7 days	N/A	Configurable	14 days	14 days
Availability	3 AZ	3 AZ	3 AZ	Configurable	3 AZ	3 AZ
Scale / throughput	No limit / ~ table IOPS	No limit / ~ shards	No limit / automatic	No limit / ~ nodes	No limits / automatic	300 TPS / queue
Parallel consumption	Yes	Yes	No	Yes	No	No
Stream MapReduce	Yes	Yes	N/A	Yes	N/A	N/A
Row/object size	400 KB	1 MB	Destination row/object size	Configurable	256 KB	256 KB
Cost	Higher (table cost)	Low	Low	Low (+admin)	Low-medium	Low-medium
	Hot					Warm

File Storage



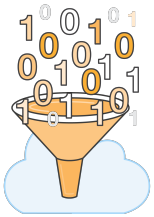
Why is AWS S3 good for Big Data?

- Very high bandwidth, designed for 99.99% availability & 99.999999999% durability
- Natively supported by big data frameworks (Spark, Hive, Presto, etc.)
- No need to run compute clusters for storage (unlike HDFS)
- Multiple & heterogeneous analysis clusters can use the same data
- Tiered-storage (Standard, IA, Amazon Glacier) via life-cycle policies
- Secure – SSL, client/server-side encryption at rest, KMS
- Low cost

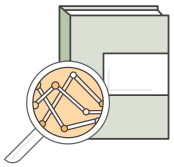
S3 Analytics, Object Tagging, Inventory, and Metrics

- S3 Analytics
- S3 Object Tagging
- S3 Inventory
- S3 CloudWatch Metrics

Data Lake



Data Ingestion



Catalogue & Search

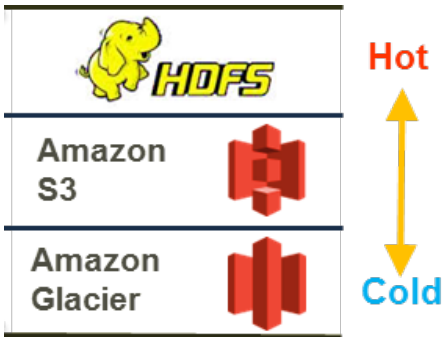
Protect & Secure

Access & User Interface

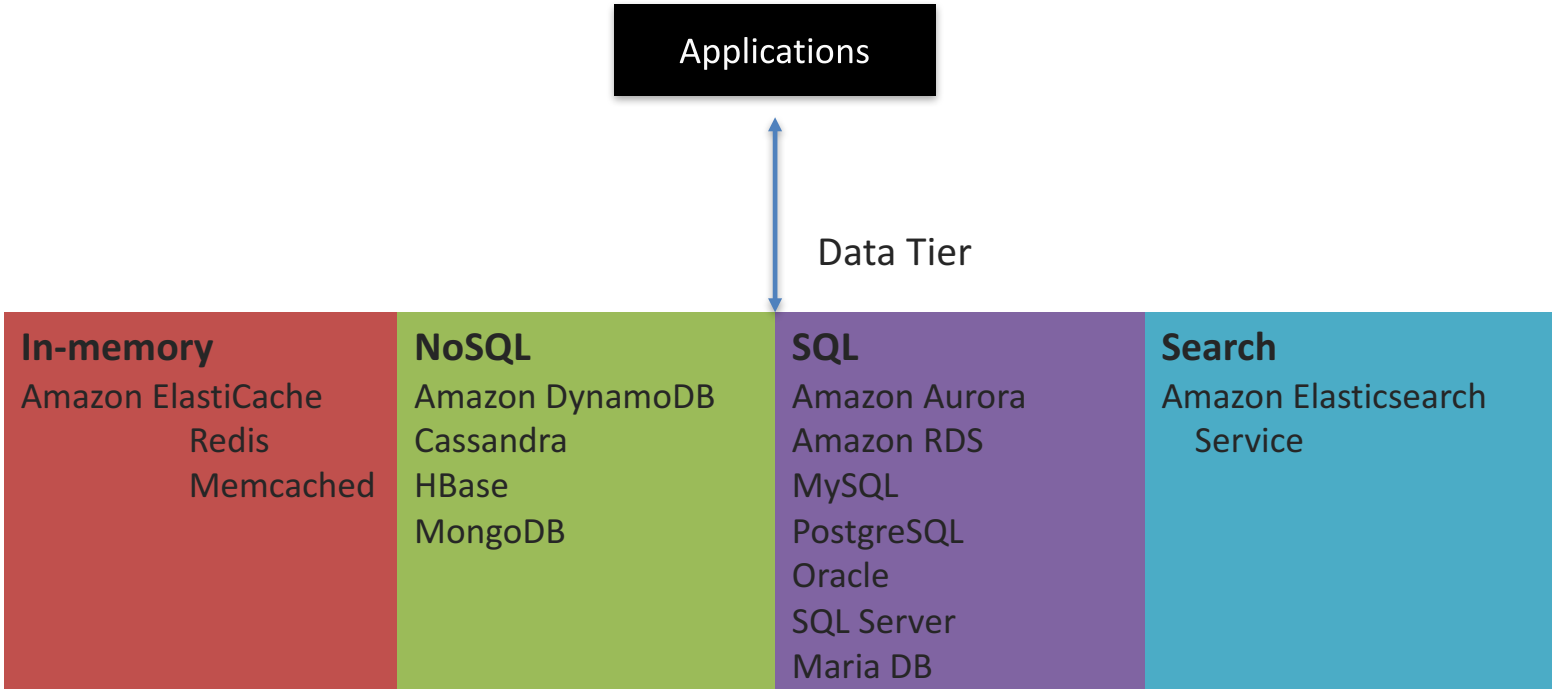
Build a Data Lake on AWS

HDFS & Data Tiering

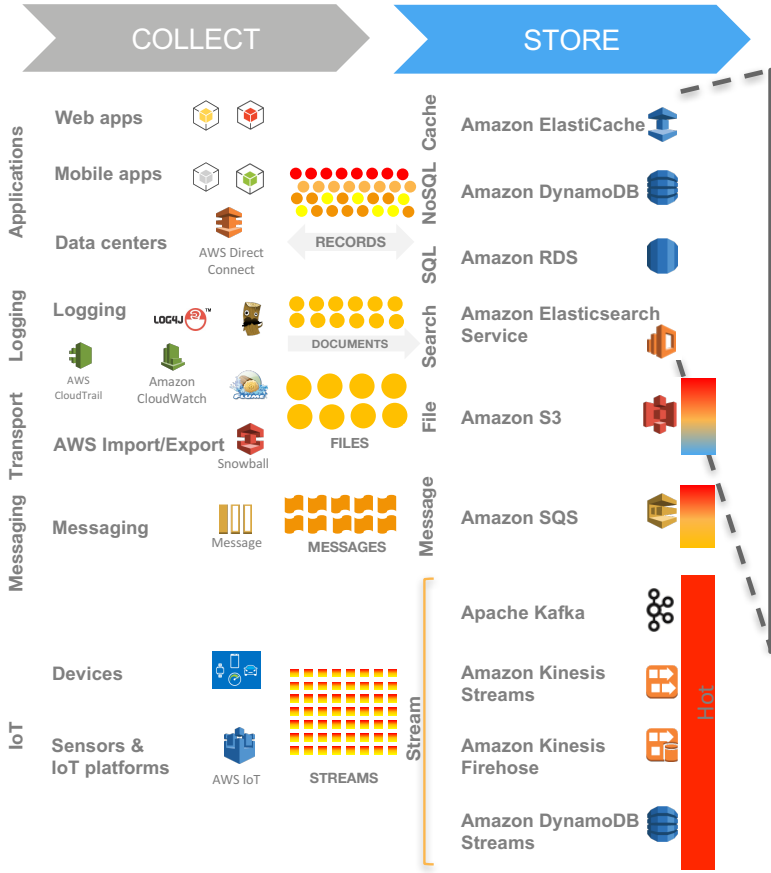
- Use HDFS for very frequently accessed (hot) data
- Use Amazon S3 Standard for frequently accessed data
- Use Amazon S3 Standard – IA for less frequently accessed data
- Use Amazon Glacier for archiving cold data



Right tool for the right job



Data Tier



Amazon ElastiCache

- Managed Memcached or Redis service

Amazon DynamoDB

- Managed NoSQL database service

Amazon RDS

- Managed relational database service

Amazon Elasticsearch Service

- Managed Elasticsearch service

Which data store should we use?

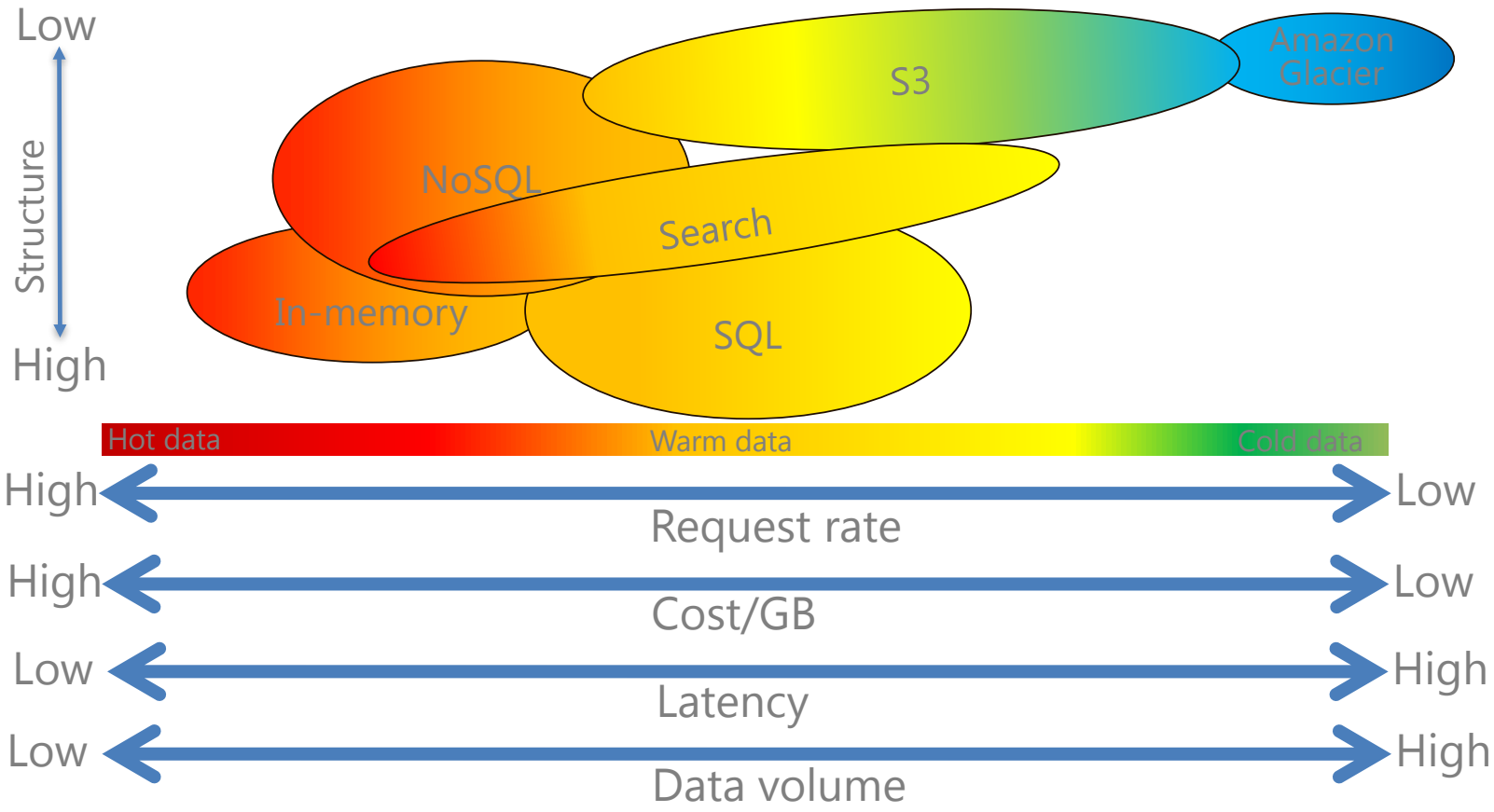
- Data structure → Fixed schema, JSON, key-value
- Access patterns → Store data in the format you will access it
- Data characteristics → Hot, warm, cold
- Cost → Right cost

Data Structure and Access Patterns

Access Patterns	What to use?
Put/Get (key, value)	In-memory, NoSQL
Simple relationships → 1:N, M:N	NoSQL
Multi-table joins, transaction, SQL	SQL
Faceting, search	Search

Data Structure	What to use?
Fixed schema	SQL, NoSQL
Schema-free (JSON)	NoSQL, Search
(Key, value)	In-memory, NoSQL

Data Store by characteristic



Which data store should we use?

	Amazon ElastiCache	Amazon DynamoDB	Amazon RDS/Aurora	Amazon ES	Amazon S3	Amazon Glacier
Average latency	ms	ms	ms, sec	ms,sec	ms,sec,min (~ size)	hrs
Typical data stored	GB	GB–TBs (no limit)	GB–TB (64 TB max)	GB–TB	MB–PB (no limit)	GB–PB (no limit)
Typical item size	B-KB	KB (400 KB max)	KB (64 KB max)	B-KB (2 GB max)	KB-TB (5 TB max)	GB (40 TB max)
Request Rate	High – very high	Very high (no limit)	High	High	Low – high (no limit)	Very low
Storage cost GB/month	$$	¢¢	¢¢	¢¢	¢	¢4/10
Durability	Low - moderate	Very high	Very high	High	Very high	Very high
Availability	High 2 AZ	Very high 3 AZ	Very high 3 AZ	High 2 AZ	Very high 3 AZ	Very high 3 AZ
	Hot data		Warm data			Cold data

CMC SYSTEM INTEGRATION

Towards the digital future

Process - Analyze

Analytics Type & Frameworks

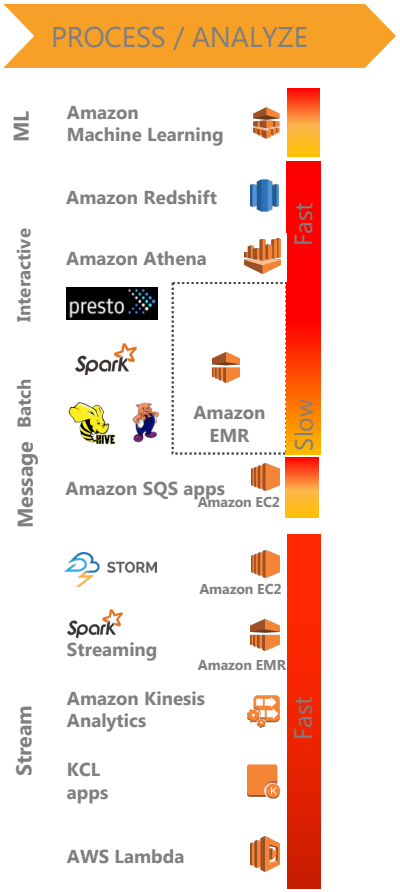
Batch
- Takes minutes to hours
- Example: Daily/weekly/monthly reports
- Amazon EMR (MapReduce, Hive, Pig, Spark)

Interactive
- Takes seconds
- Example: Self-service dashboards
- Amazon Redshift, Amazon Athena, Amazon EMR (Presto, Spark)

Message
- Takes milliseconds to seconds
- Example: Message processing
- Amazon SQS applications on Amazon EC2

Stream
- Takes milliseconds to seconds
- Example: Fraud alerts, 1 minute metrics
- Amazon EMR (Spark Streaming), Amazon Kinesis Analytics, KCL, Storm, AWS Lambda

Machine Learning
- Takes milliseconds to minutes
- Example: Fraud detection, forecast demand
- Amazon ML, Amazon EMR (Spark ML)



Which Stream & Message processing should we use?

	Amazon EMR (Spark Streaming)	Apache Storm	KCL Application	Amazon Kinesis Analytics	AWS Lambda	Amazon SQS Application
AWS managed	Yes (Amazon EMR)	No (Do it yourself)	No (EC2 + Auto Scaling)	Yes	Yes	No (EC2 + Auto Scaling)
Serverless	No	No	No	Yes	Yes	No
Scale / throughput	No limits / ~ nodes	No limits / ~ nodes	No limits / ~ nodes	Up to 8 KPU / automatic	No limits / automatic	No limits / ~ nodes
Availability	Single AZ	Configurable	Multi-AZ	Multi-AZ	Multi-AZ	Multi-AZ
Programming languages	Java, Python, Scala	Almost any language via Thrift	Java, others via MultiLangDaemon	ANSI SQL with extensions	Node.js, Java, Python	AWS SDK languages (Java, .NET, Python, ...)
Uses	Multistage processing	Multistage processing	Single stage processing	Multistage processing	Simple event-based triggers	Simple event based triggers
Reliability	KCL and Spark checkpoints	Framework managed	Managed by KCL	Managed by Amazon Kinesis Analytics	Managed by AWS Lambda	Managed by SQS Visibility Timeout

Fast

Which Analysis tool should we use?

	Amazon Redshift	Amazon Athena	Amazon EMR		
			Presto	Spark	Hive
Use case	Optimized for data warehousing	Ad-hoc Interactive Queries	Interactive Query	General purpose (iterative ML, RT, ..)	Batch
Scale/throughput	~Nodes	Automatic / No limits	~ Nodes		
AWS Managed Service	Yes	Yes, Serverless	Yes		
Storage	Local storage	Amazon S3	Amazon S3, HDFS		
Optimization	Columnar storage, data compression, and zone maps	CSV, TSV, JSON, Parquet, ORC, Apache Web log, AVRO	Framework dependent		
Metadata	Amazon Redshift managed	Athena Catalog Manager	Hive Meta-store		
BI tools supports	Yes (JDBC/ODBC)	Yes (JDBC)	Yes (JDBC/ODBC & Custom)		
Access controls	Users, groups, and access controls	AWS IAM	Integration with LDAP		
UDF support	Yes (Scalar)	No	Yes		

Fast ... Slow

ETL (Extract, Transform & Load) Tool?

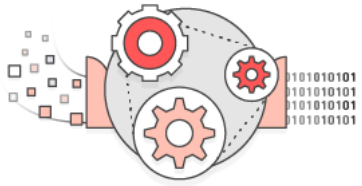
Data Integration Partners

Reduce the effort to move, cleanse, synchronize, manage, and automatize data related processes.

alteryx · ATTUNITY · informatica · ironSource .atom · MATILLION · snaplogic · bryte · talend · alooma · Boomi

https://aws.amazon.com/big-data/partner-solutions/

AWS Glue

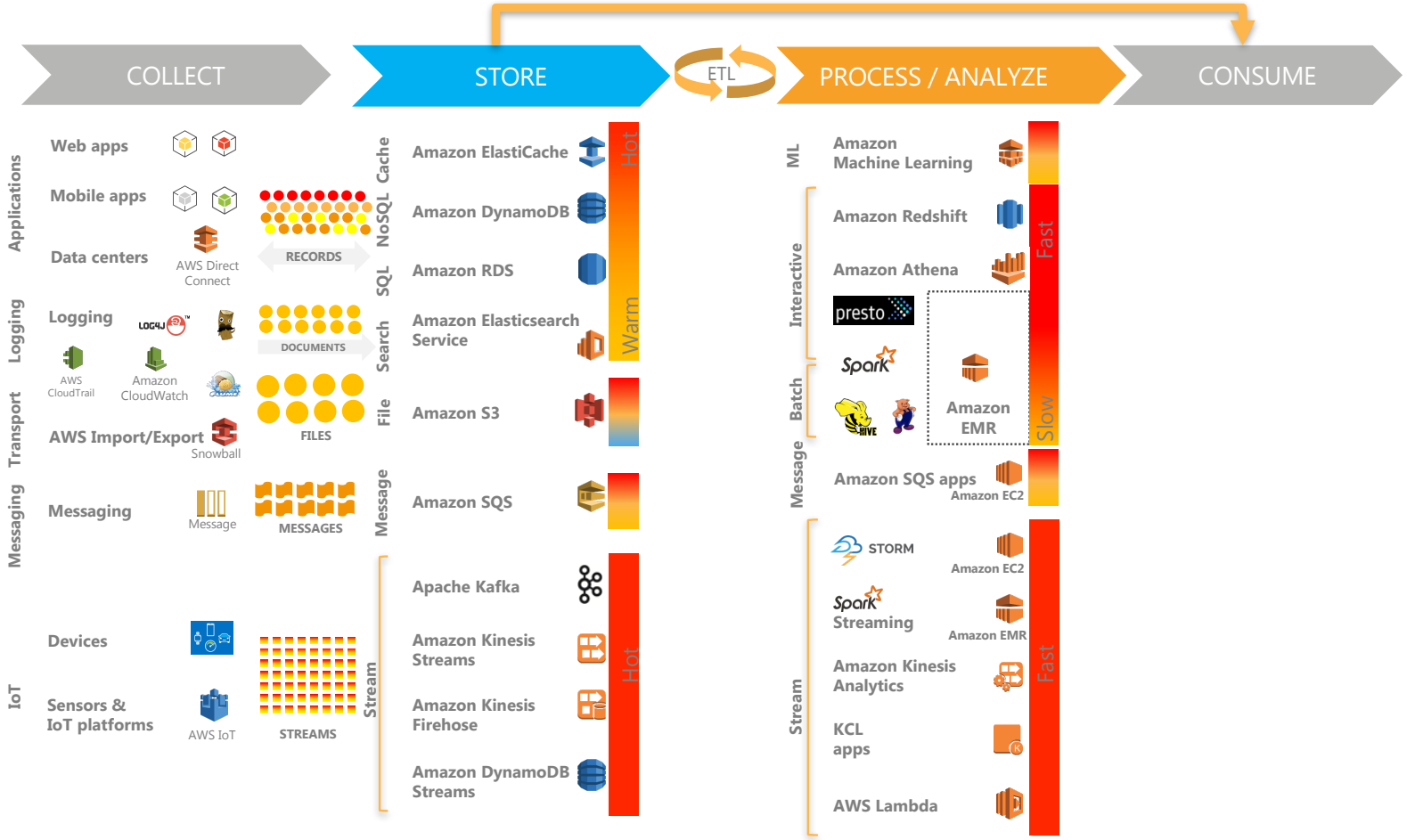


AWS Glue is a fully managed ETL service that makes it easy to understand your data sources, prepare the data, and move it reliably between data stores

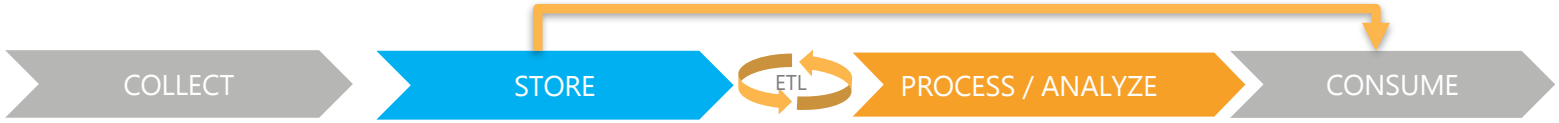
CMC SYSTEM INTEGRATION
Towards the digital future

Consume / Visualize / Analyze / Share

Consume


COLLECT
STORE
ETL
PROCESS / ANALYZE
CONSUME
Applications
Web apps
Mobile apps
Data centers
AWS Direct Connect
Logging
Logging
AWS CloudTrail
Amazon CloudWatch
Transport
AWS Import/Export
Snowball
Messaging
Messaging
Message
IoT
Devices
Sensors & IoT platforms
AWS IoT
RECORDS
DOCUMENTS
FILES
MESSAGES
STREAMS
Cache
NoSQL
SQL
Search
File
Message
Stream
Amazon ElastiCache
Amazon DynamoDB
Amazon RDS
Amazon Elasticsearch Service
Amazon S3
Amazon SQS
Apache Kafka
Amazon Kinesis Streams
Amazon Kinesis Firehose
Amazon DynamoDB Streams
Hot
Warm
Hot
ML
Amazon Machine Learning
Interactive
Amazon Redshift
Amazon Athena
presto
Batch
Spark
HIVE
Amazon EMR
Message
Amazon SQS apps
Amazon EC2
Stream
STORM
Amazon EC2
Spark Streaming
Amazon EMR
Amazon Kinesis Analytics
KCL apps
AWS Lambda
Fast
Slow
Fast

Consume

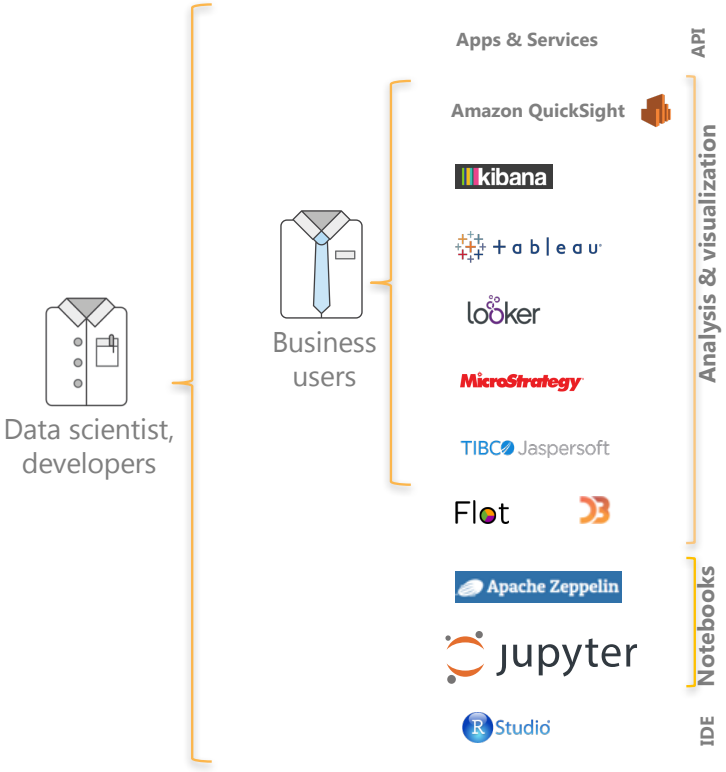


Applications & API

Analysis and visualization

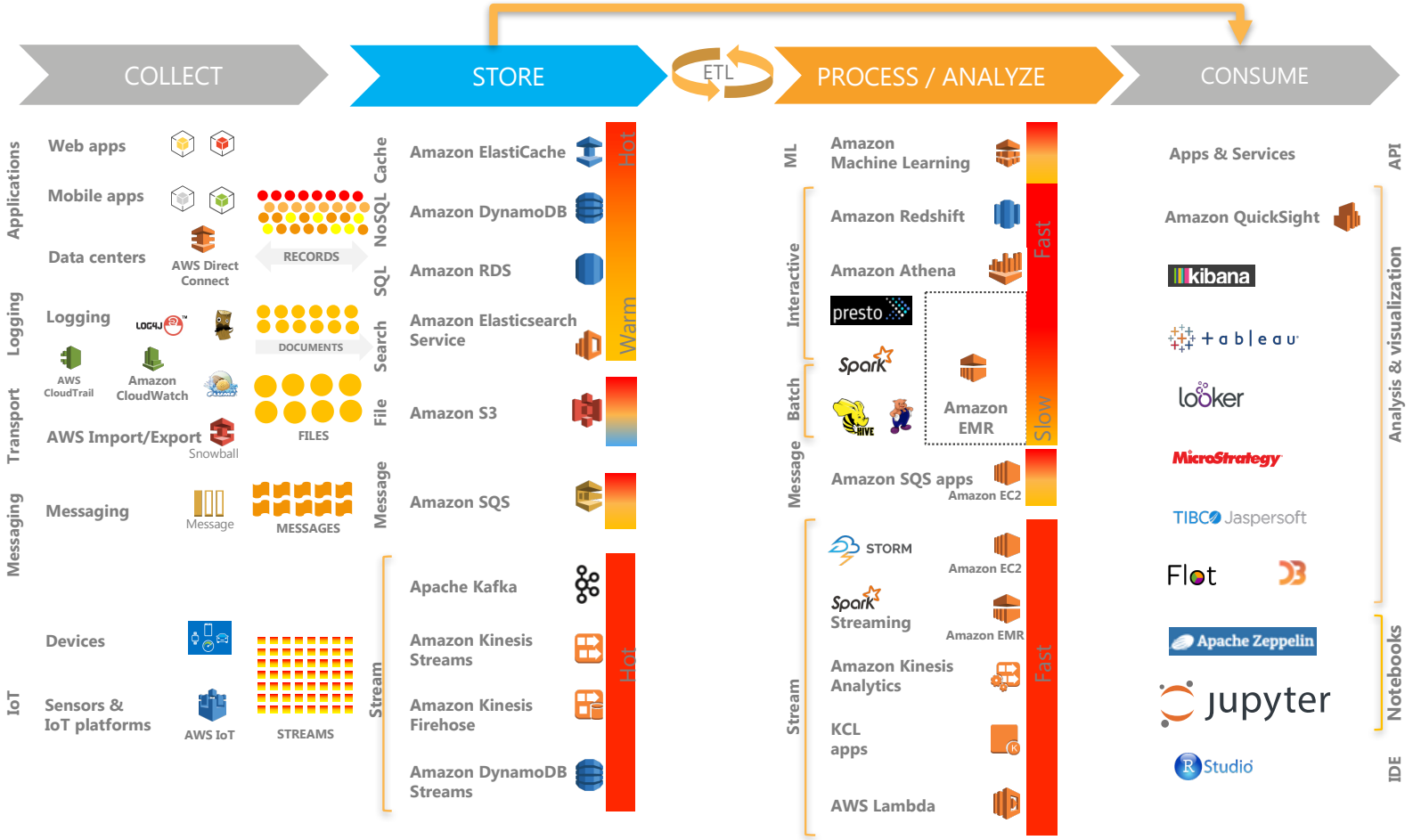
Notebooks

IDE



CMC SI
Towards the digital future

Big Data Reference Architecture on AWS



CMC SYSTEM INTEGRATION

Towards the digital future

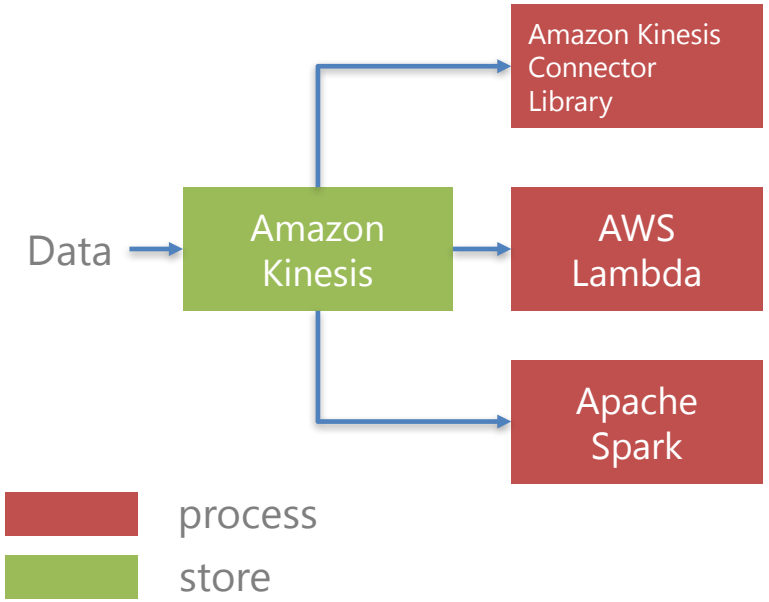
Design Patterns

Primitive: Decoupled Data Bus

- Storage decoupled from processing
- Multiple stages

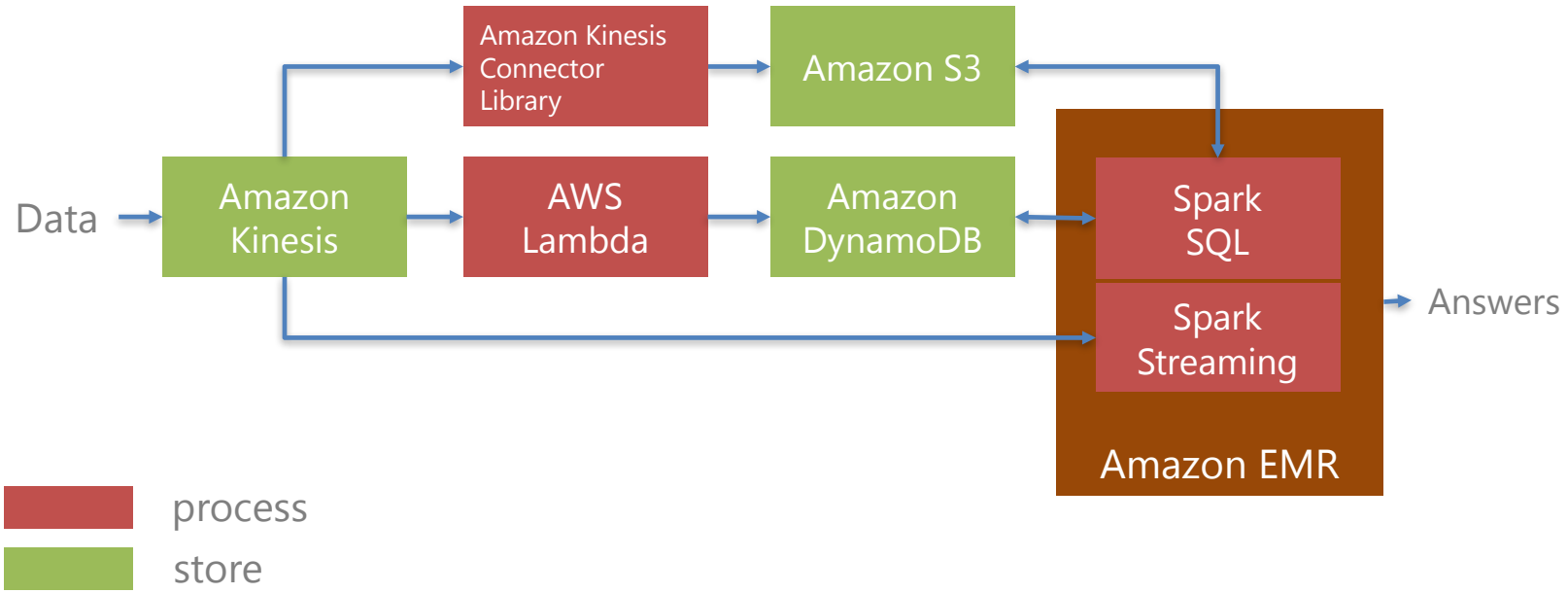
Primitive: Pub/Sub

- Parallel stream consumption/processing

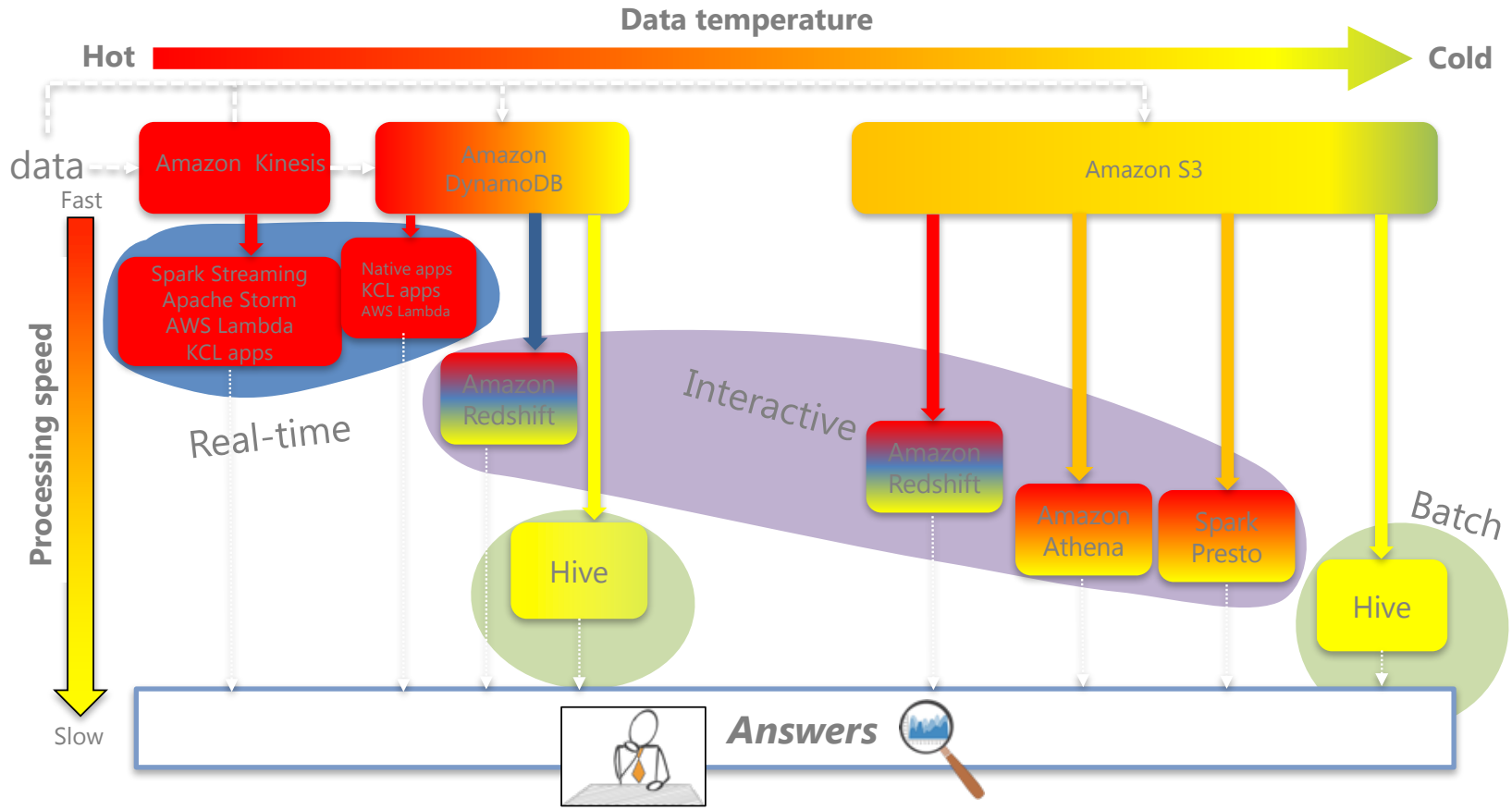


Primitive: Materialized View

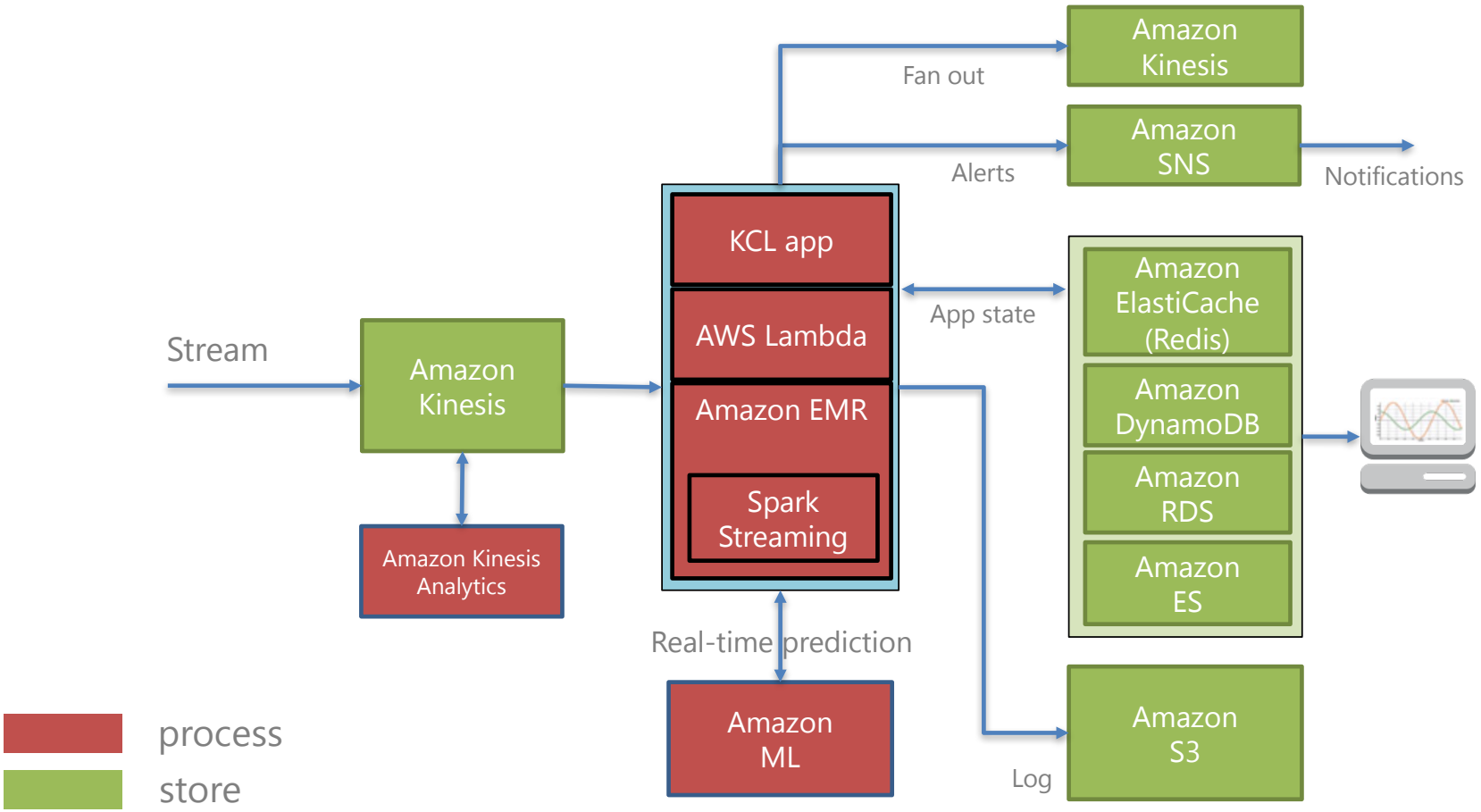
- Analysis framework reads from or writes to multiple data stores



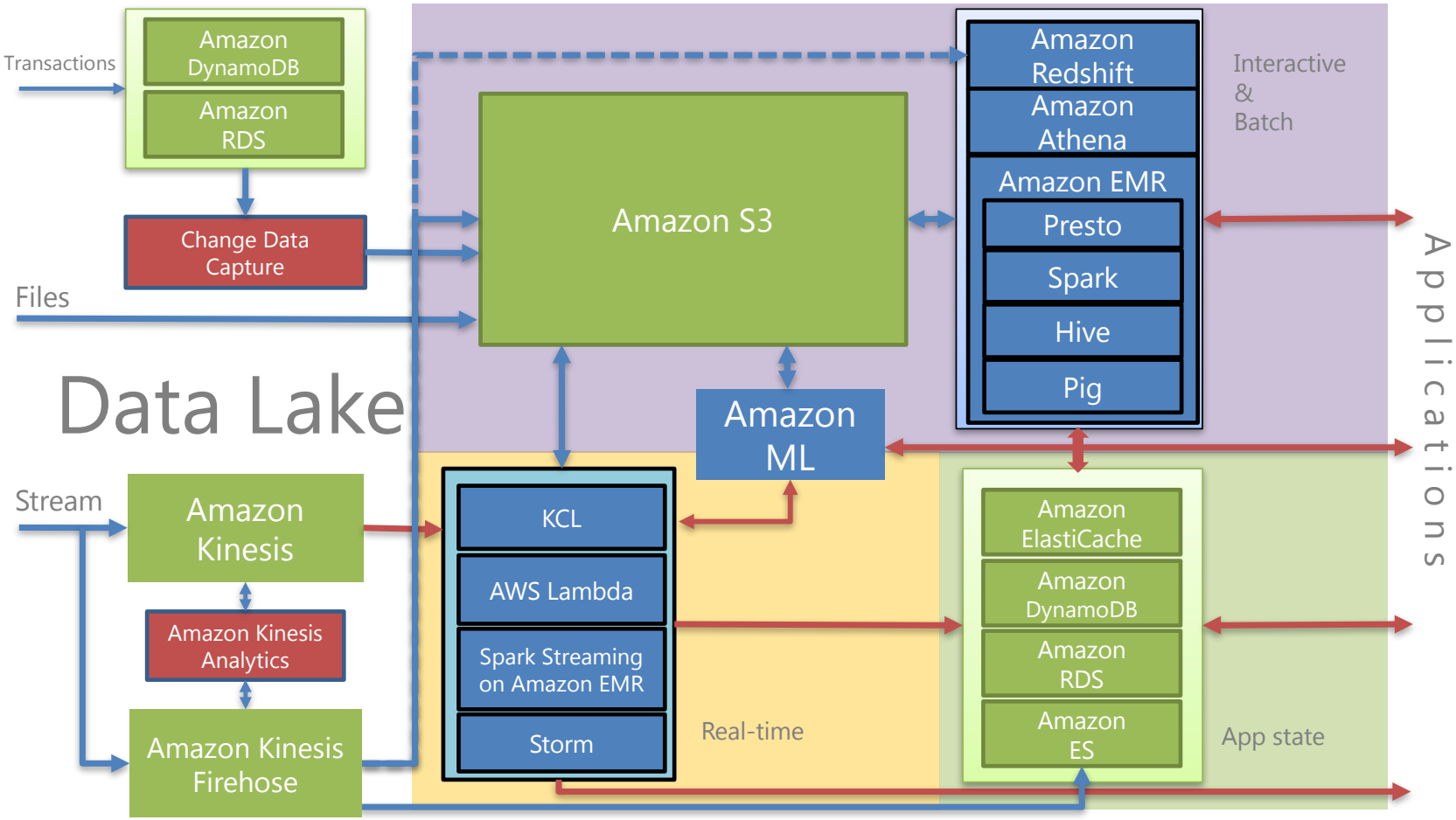
Map on Data Temperature by processing speed



Real-time Analytics



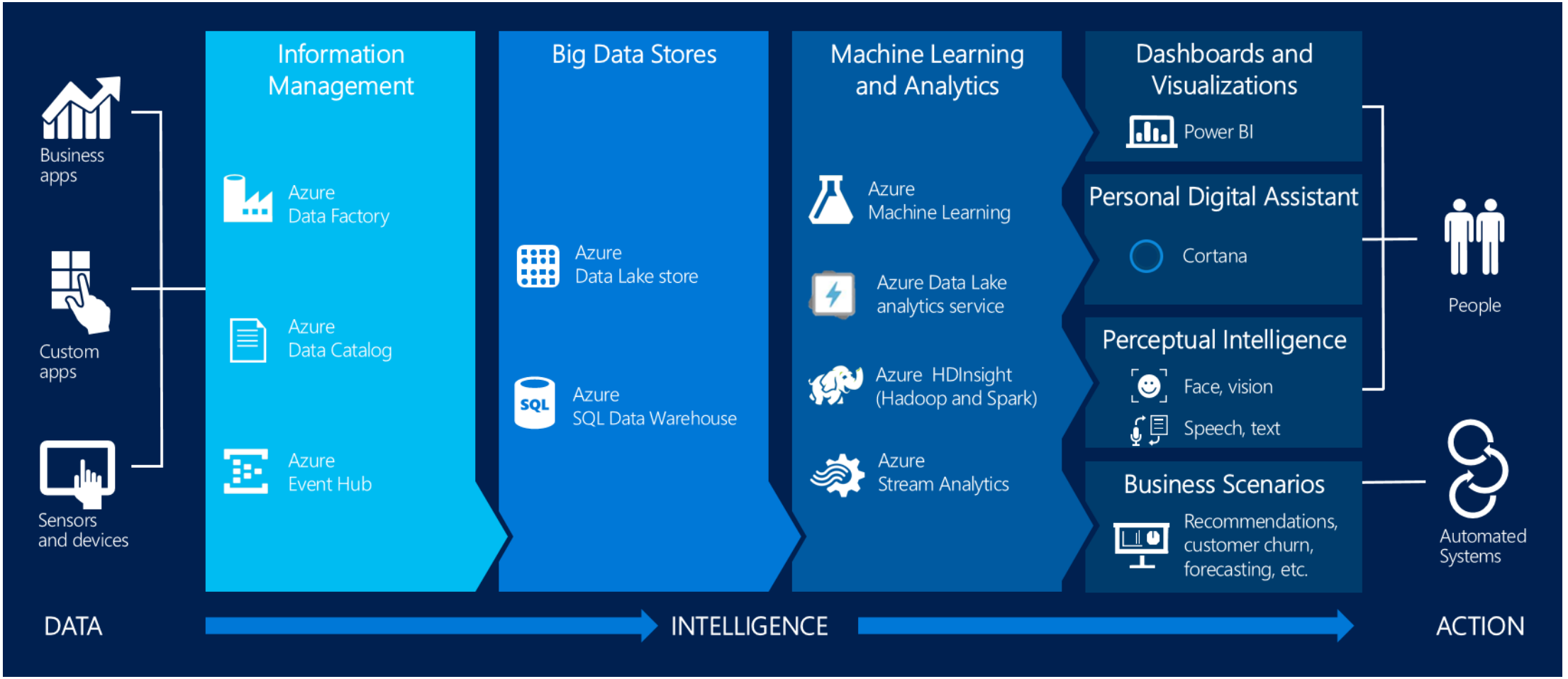
AWS Data Lake



Key Takeaway

- Build decoupled systems
  - **Data → Store → Process → Store → Analyze → Answers**
- Use the right tool for the job
  - Data structure, latency, throughput, access patterns
- Leverage AWS managed services
  - Scalable/elastic, available, reliable, secure, no/low admin
- Use log-centric design patterns
  - Immutable log, batch, interactive & real-time views
- Be cost-conscious
  - Big data ≠ big cost

Ref: Big Data Architecture on Azure



CMC SYSTEM INTEGRATION
Towards the digital future

THANK YOU!