Streaming Data Analytics

Saptarshi Pyne
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur, Rajasthan, India 342030

CSL4030 Data Engineering Lectures 36-41
November 15th, 17th, 20th, 22nd, 24th, 26th, 2023

What we discussed in the last class

Query optimization

We learnt various techniques for

- Enumerating all possible query evaluation plans
- Estimating the cost of each plan
- Choosing an optimal or approximately optimal plan

What is streaming data?

Data that is continuously being generated (streamed).

Examples:

- Click-through data (e.g., for Google search, every page request made by any user anywhere)
- Financial trading data
- E-commerce transactions
- Live streams and broadcasts (sports, online gaming, vlogs, podcasts, election coverage, reporting wars, disasters, pandemics, etc.)
- Leaderboards and ‘relative’ leaderboards (gaming apps and gamified apps)

https://uxdesign.cc/building-better-leaderboards-a5013d19cbd7

What is streaming data? (contd.)

Examples: (contd.)

- Sensor data: seismometers, tsunami warning sensors, etc.
- CC TV cameras
- Geo-location trackers (e.g., aeroplanes, Google Maps)
- IoT & smart devices (e.g., smartwatch step counters)
- Network routers (incoming and outgoing packet streams of the IITJ intranet)
- Server monitors (e.g., CPU monitors)
- Satellites orbiting the Earth, Moon, Mars, etc.
- Collaborative work streams (coding, designing, etc.) [1]

[1] https://aws.amazon.com/blogs/media/the-crown-in-the-cloud/

What is NOT a streaming data?

Batch data

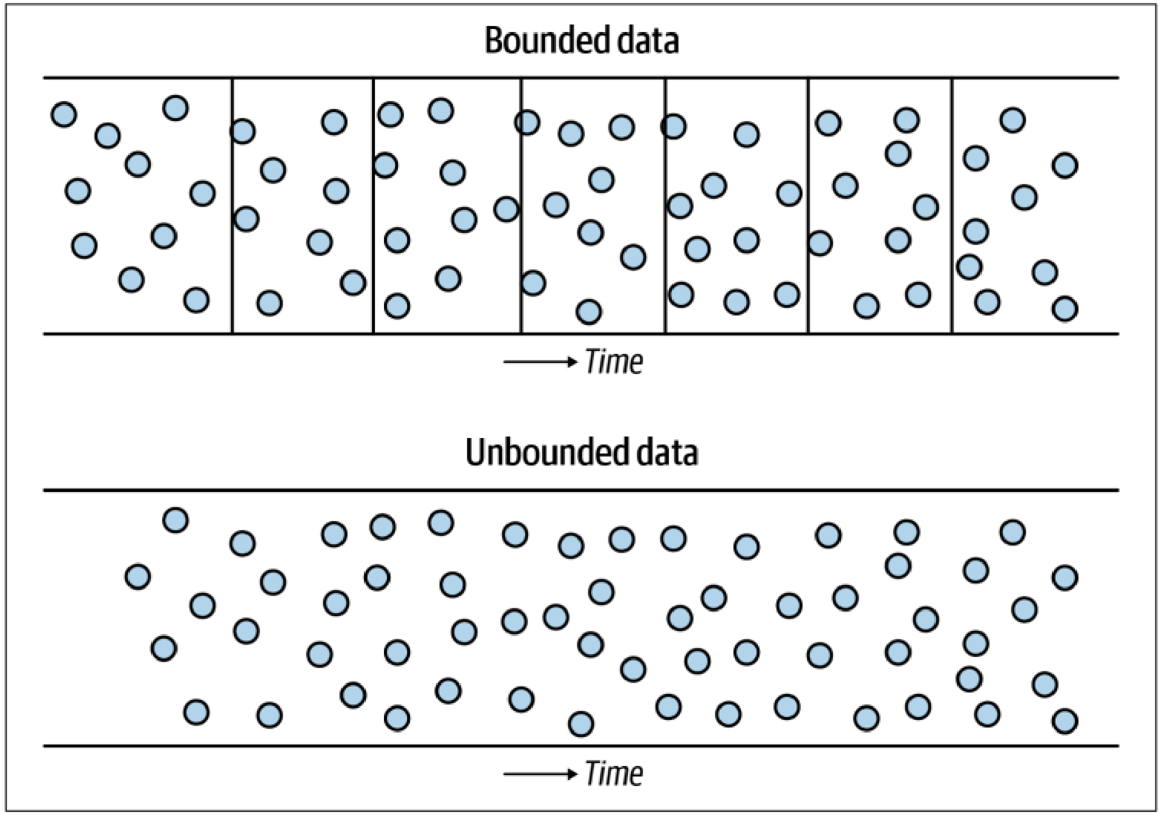


Figure 7-3. Bounded versus unbounded data

“All data is unbounded until it’s bounded” (irrespective of whether they are generated continuously or sporadically).

Boundaries are created for convenience. Such **bounded units of data** are called **batches**.

E.g., a TODO list. The TODO items appear in our mind as an **unbounded** stream of thoughts. Then we write them on a piece of paper which is **bounded**.

Streaming data is 'ingested' in (near) real-time

Data ingestion is the process of **moving data from one application to another application**. For example, moving a live stream data from YouTube to the Google Cloud storage.

Data ingestion can be performed at different frequencies:

- Streaming data is ingested **as soon as it arrives**.
- Sometimes data is ingested in **micro-batches** (e.g., once a minute).

- On the other hand, data **batches** are usually ingested at longer time intervals (e.g., once a day).

Pages 237-238, Reis and Housley

What is a data pipeline?

Data is like **water**.

Once generated at the **source**, it flows like a **stream**.

Sometimes we put dams to consume it in **batches**.

File:SQK Dam DSC 3657.jpg – Wikipedia. Creator: MARTINA NOLTE . Copyright: MARTINA NOLTE

What is a data pipeline? (contd.)

Sometimes we split it and divert it to serve different applications.

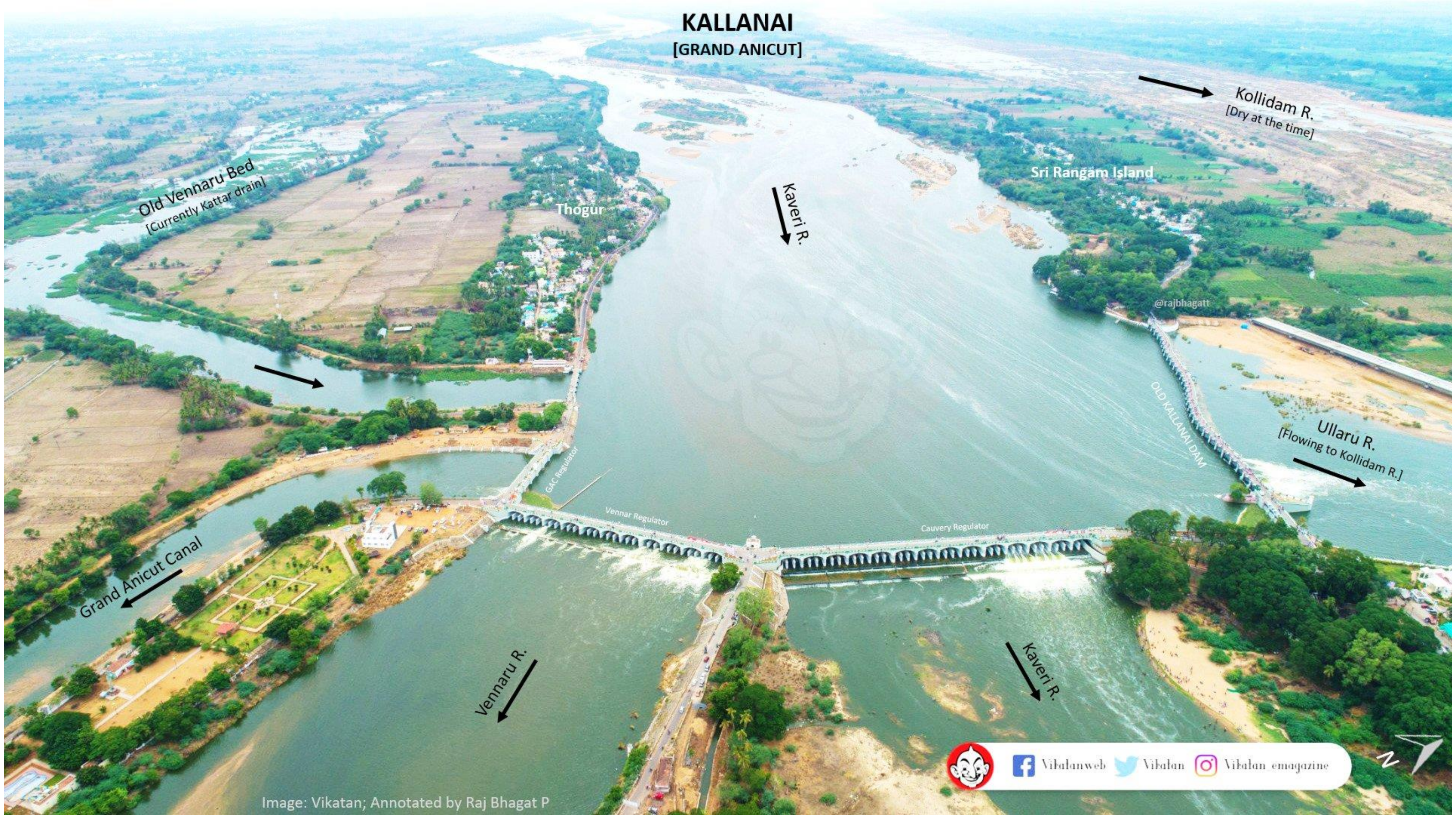


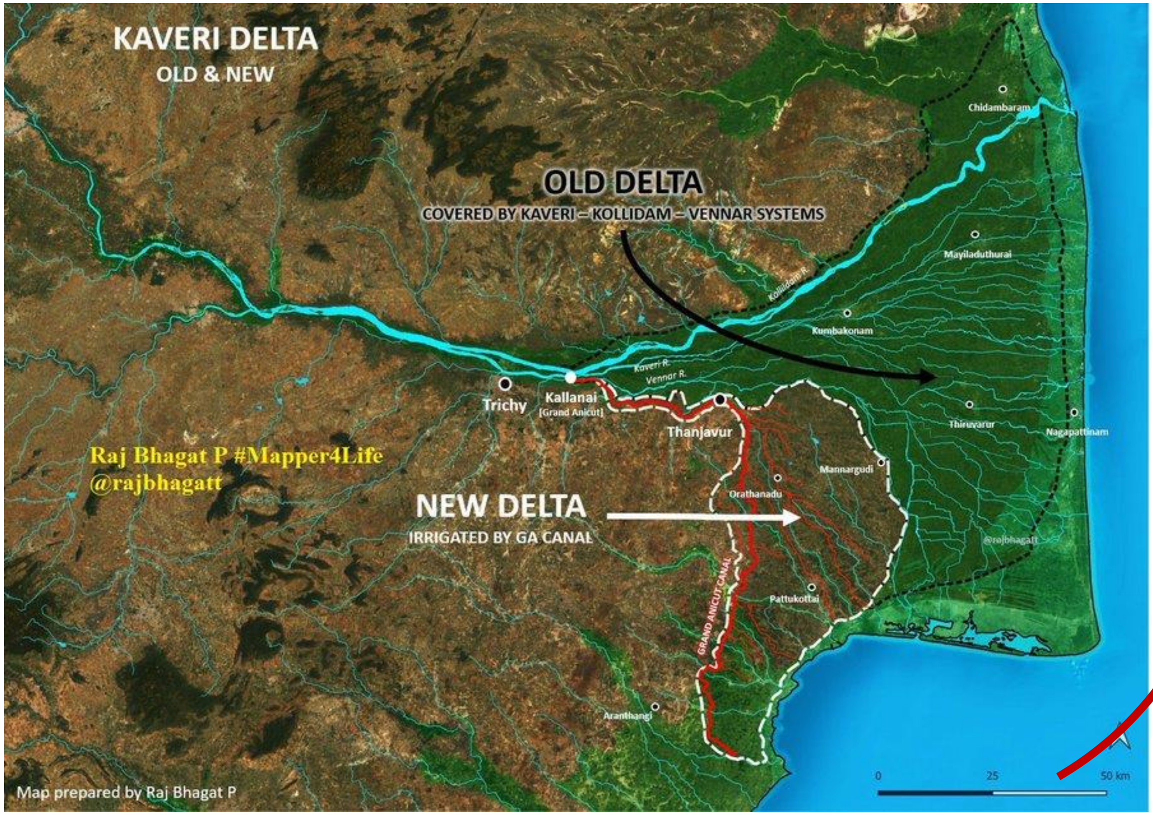
Image: Vikatan; Annotated by Raj Bhagat P

The Kallanai Dam on the Kaveri river in Tamil Nadu. Constructed by King Karikala Chola in 150 AD.

https://www.youtube.com/watch?v=1Tu9Tp1tgM8

What is a data pipeline? (contd.)

Applications such as feeding a lot of people.

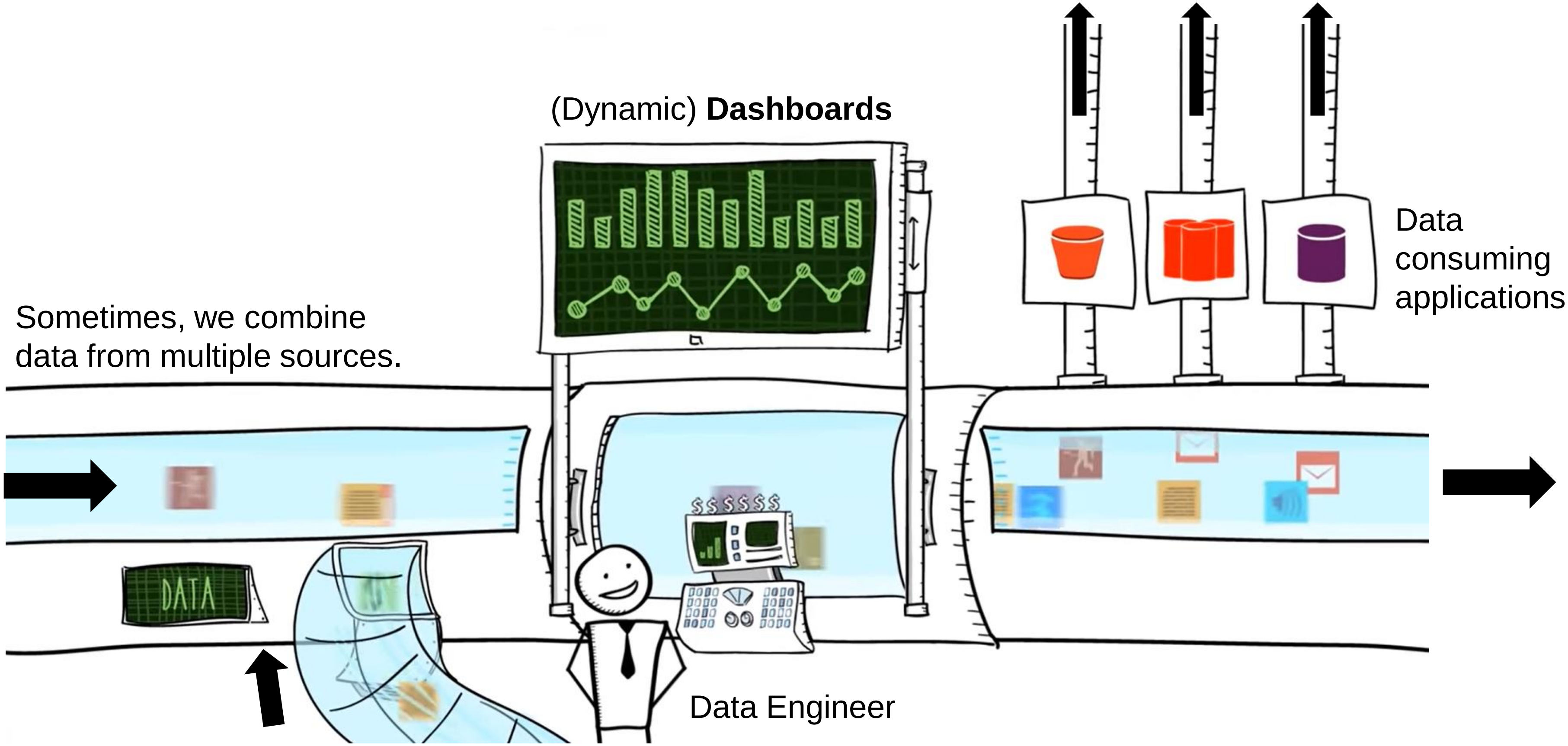


Natural course of the Kaveri

All these people would never have water otherwise.

What is a data pipeline? (contd.)

Similarly, we channelize data into multiple applications.



(Dynamic) **Dashboards**

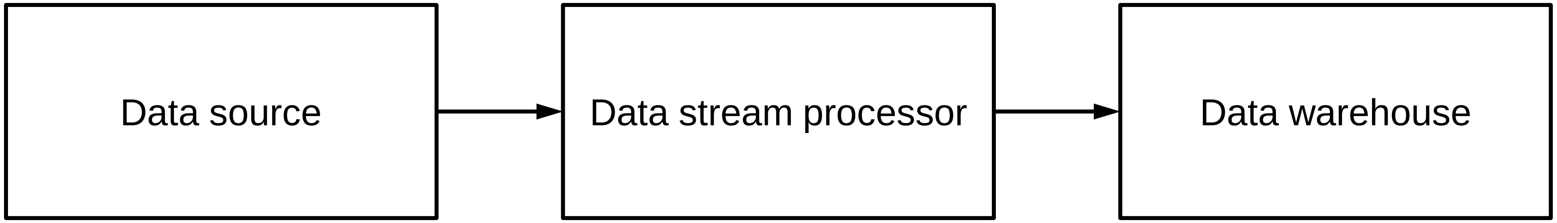
Sometimes, we combine data from multiple sources.

Data consuming applications

Data Engineer

https://www.youtube.com/watch?v=MbEfiX4sMXc

Example of a traditional data pipeline: Old-day financial trading floor



Data source

Data stream processor

Data warehouse

A GUI software installed on on-premise desktop computers of the company. Employees are constantly making new trades.

A software ingesting and processing (extracting-transforming-loading) transactional trading data

A software that stores the data into domain-specific warehouses and produces domain-specific business intelligence reports at the end of every day.

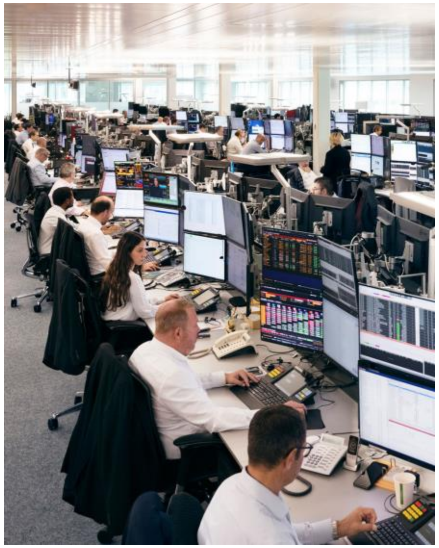
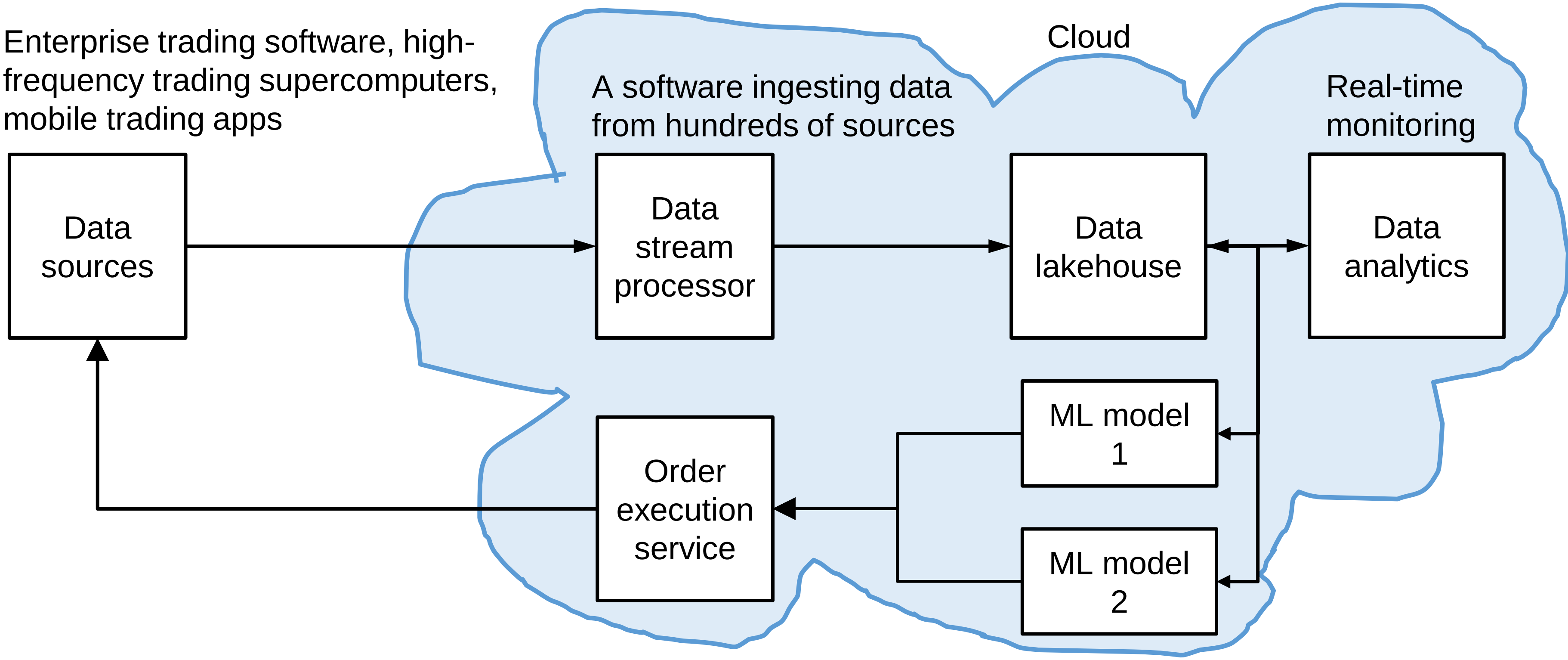


Image courtesy: https://www.pictet.com/uk/en/trading-and-sales

Example of a modern data pipeline: Modern-day financial trading floor



Enterprise trading software, high-frequency trading supercomputers, mobile trading apps

Cloud

A software ingesting data from hundreds of sources

Real-time monitoring

Data sources

Data stream processor

Data lakehouse

Data analytics

ML model 1

ML model 2

Order execution service

Image courtesy (L-R): https://forextrader.live/2016/06/14/forex-trading-floors/ , https://towardsdatascience.com/assembling-an-entry-level-high-frequency-trading-hft-system-e7538545b2a9 , https://www.reliancesmartmoney.com/stocks/trading-platforms/tick-pro

Case study of commercial data pipelines: Amazon Kinesis

Amazon Kinesis is a family of services offered by Amazon Web Services (AWS) for processing and analysing streaming data.

The main services of this family are:

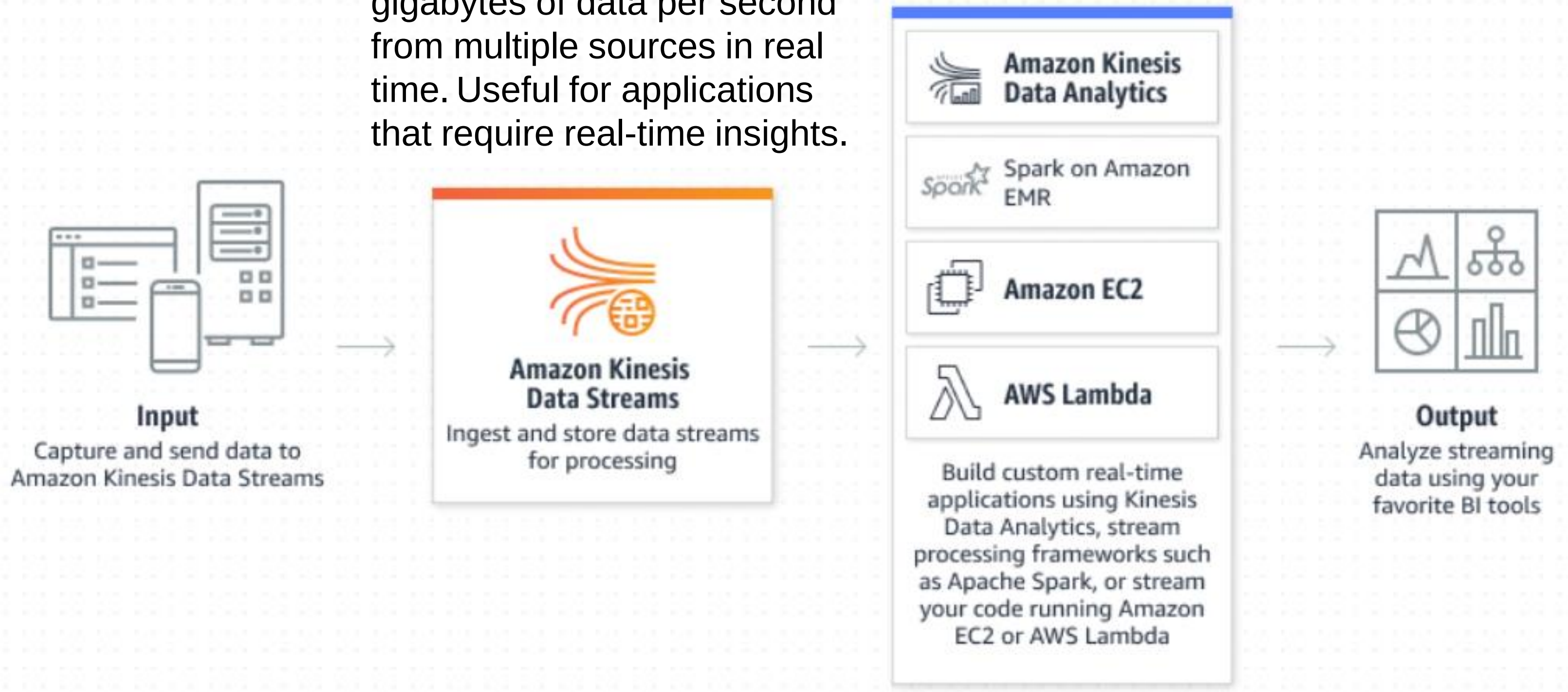
- Kinesis Data Streams
- Kinesis Data Firehose
- Kinesis Video Streams
- Kinesis Data Analytics

https://aws.amazon.com/pm/kinesis/

Case study of commercial data pipelines: Amazon Kinesis Data Streams

Amazon EMR, Amazon Elastic Compute Cloud (EC2), and AWS Lambda are different types of cloud computing services.

Ingests and processes gigabytes of data per second from multiple sources in real time. Useful for applications that require real-time insights.



Input

Capture and send data to Amazon Kinesis Data Streams

Amazon Kinesis Data Streams

Ingest and store data streams for processing

Amazon Kinesis Data Analytics

Spark on Amazon EMR

Amazon EC2

AWS Lambda

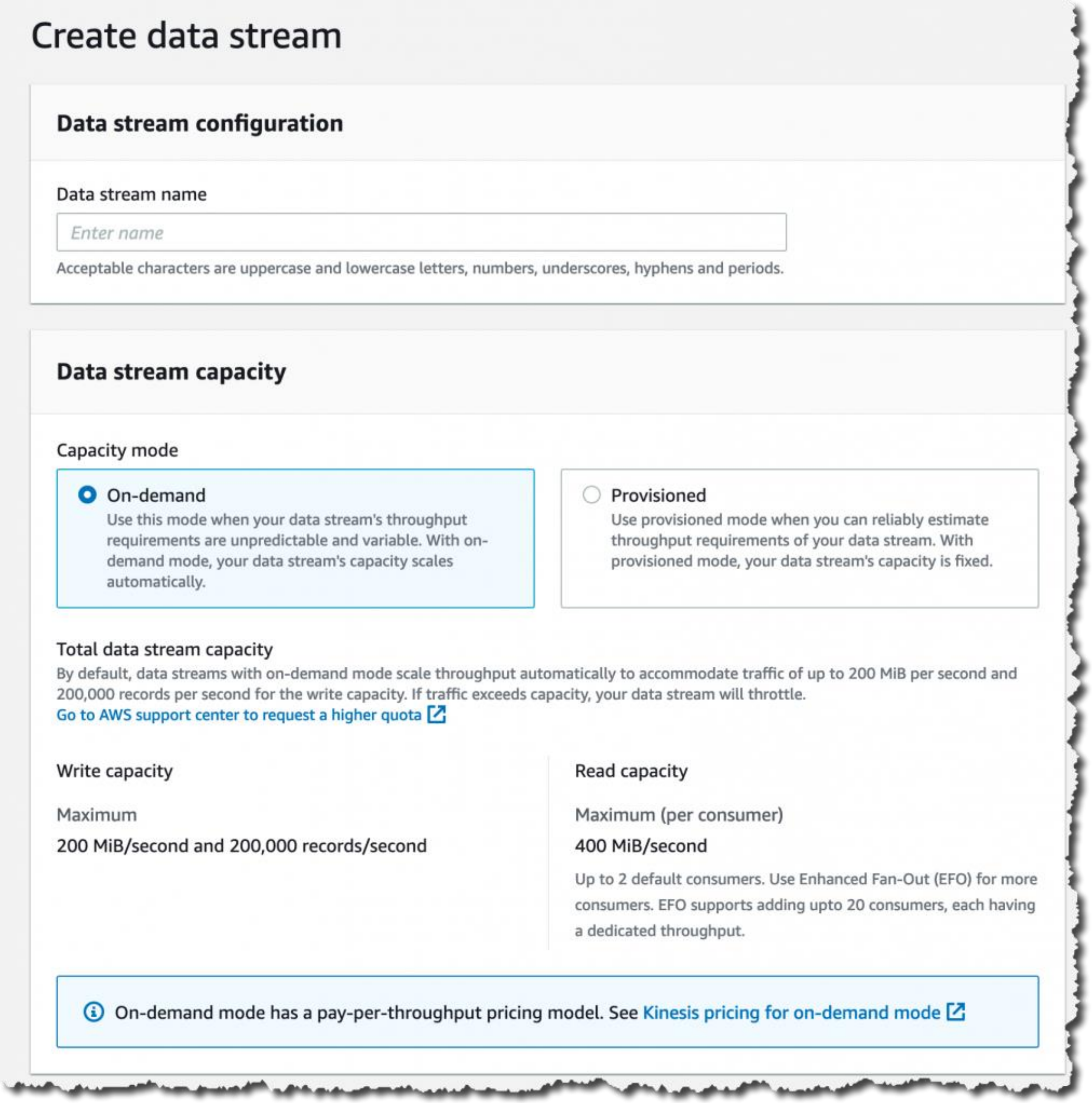
Build custom real-time applications using Kinesis Data Analytics, stream processing frameworks such as Apache Spark, or stream your code running Amazon EC2 or AWS Lambda

Output

Analyze streaming data using your favorite BI tools

https://aws.amazon.com/kinesis/
https://www.infoq.com/news/2021/12/kinesis-data-streams-ondemand/

Case study of commercial data pipelines: Amazon Kinesis Data Streams (contd.)



Create data stream

Data stream configuration

Data stream name

Enter name

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

Data stream capacity

Capacity mode

● On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

○ Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

Total data stream capacity
By default, data streams with on-demand mode scale throughput automatically to accommodate traffic of up to 200 MiB per second and 200,000 records per second for the write capacity. If traffic exceeds capacity, your data stream will throttle.
Go to AWS support center to request a higher quota

Write capacity

Maximum

200 MiB/second and 200,000 records/second

Read capacity

Maximum (per consumer)

400 MiB/second

Up to 2 default consumers. Use Enhanced Fan-Out (EFO) for more consumers. EFO supports adding upto 20 consumers, each having a dedicated throughput.

ⓘ On-demand mode has a pay-per-throughput pricing model. See Kinesis pricing for on-demand mode

The infrastructure is on the cloud (AWS). Hence, we can **scale to virtually unlimited volume of data stream**.
Moreover, **we pay for what we use**.

That means we do not have to manage the servers that our streaming based app is using. For this reason, such pay-as-you-go cloud services are also known as **serverless** services.

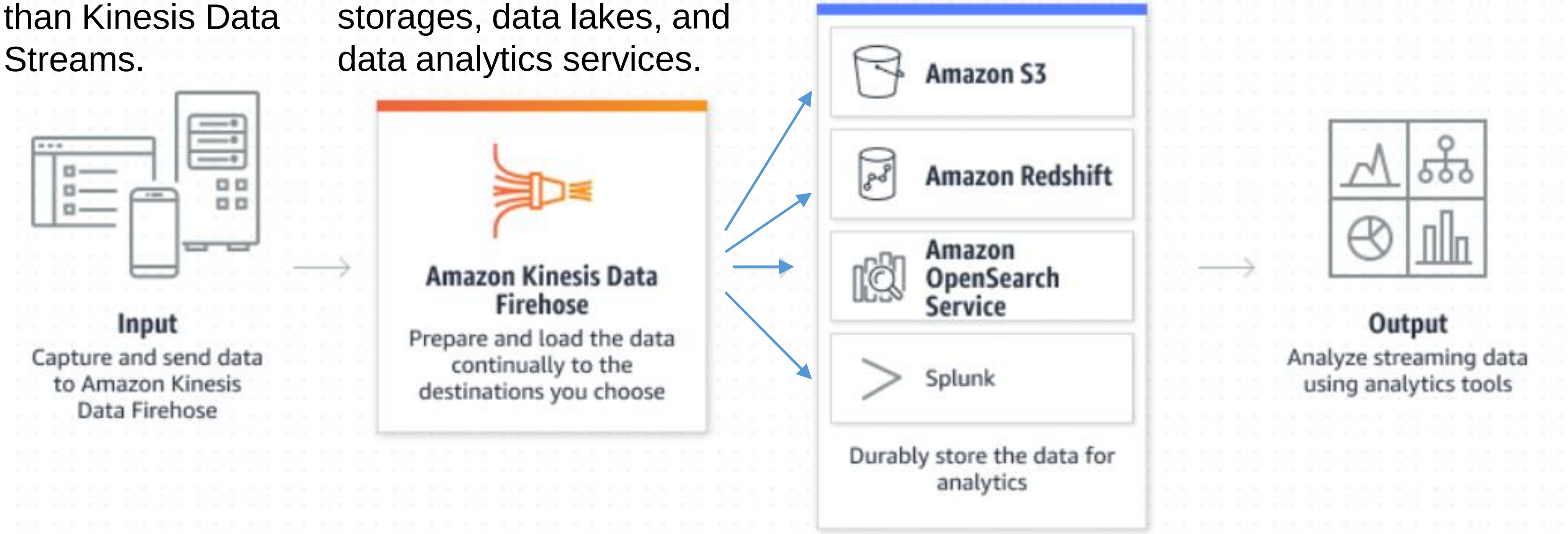
https://www.infoq.com/news/2021/12/kinesis-data-streams-ondemand/

Case study of commercial data pipelines: Amazon Kinesis Data Firehose

Input to Kinesis Data Firehose is often none other than Kinesis Data Streams.

Firehose is a highly configurable and scalable service for loading data into compatible cloud storages, data lakes, and data analytics services.

Amazon OpenSearch is an open-source data and web search-cum-analytics service.



Input
Capture and send data to Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose
Prepare and load the data continually to the destinations you choose

Amazon S3
Amazon Redshift
Amazon OpenSearch Service
Splunk
Durably store the data for analytics

Output
Analyze streaming data using analytics tools

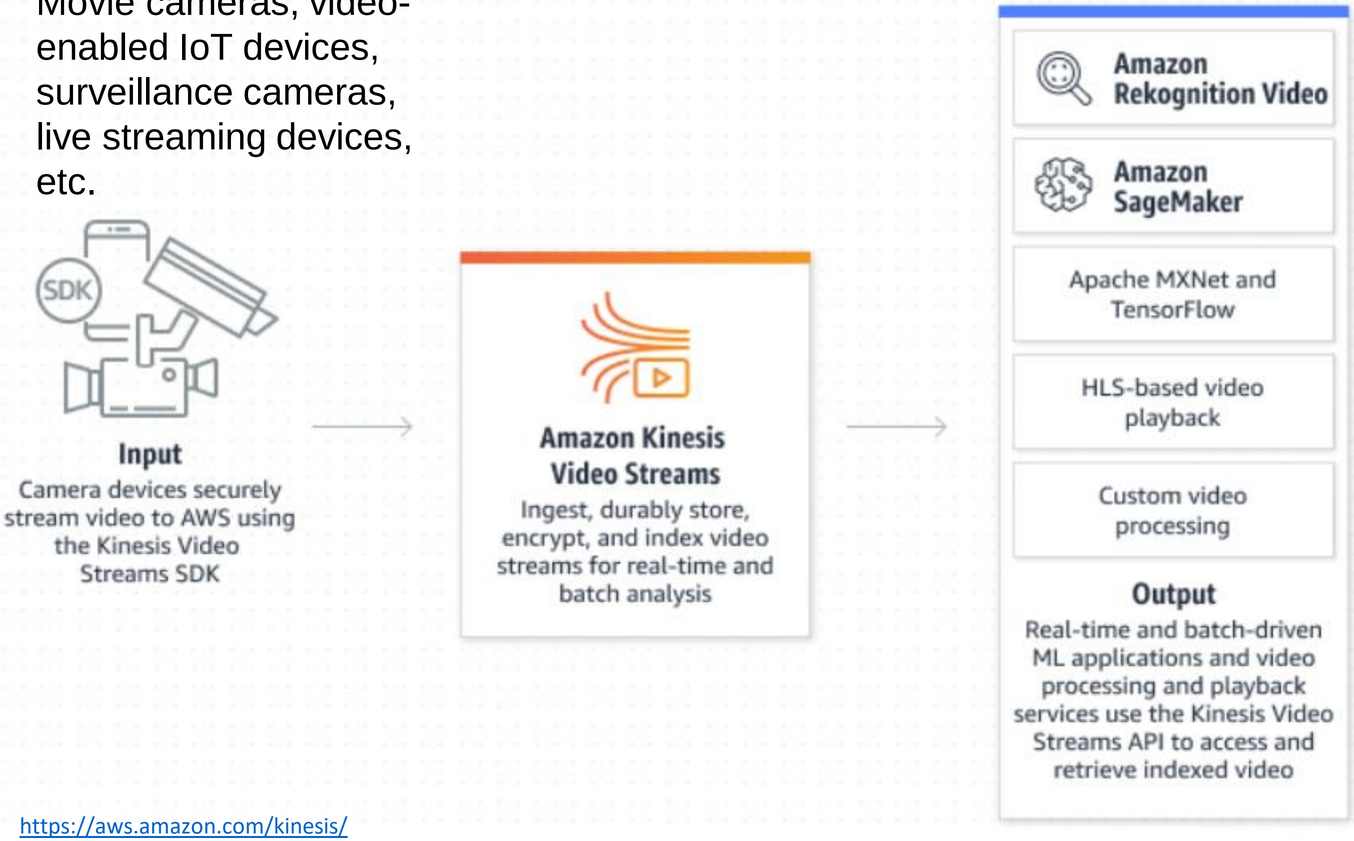
Splunk indexes and correlates streaming-cum-web data into a searchable repository in real time. This year, Cisco has offered a $28 billion all-cash deal to acquire Splunk. It is the most expensive deal in Cisco's history.

https://aws.amazon.com/kinesis/
https://www.bloomberg.com/news/articles/2023-09-21/cisco-to-buy-splunk-for-157-a-share-in-28-billion-deal
https://aws.amazon.com/what-is/opensearch/

Case study of commercial data pipelines: Amazon Kinesis Video Streams

Amazon SageMaker is a cloud computing service that enables developers to create, train, and deploy ML models directly onto end-point devices.

Movie cameras, video-enabled IoT devices, surveillance cameras, live streaming devices, etc.



Input

Camera devices securely stream video to AWS using the Kinesis Video Streams SDK

Amazon Kinesis Video Streams

Ingest, durably store, encrypt, and index video streams for real-time and batch analysis

Amazon Rekognition Video

Amazon SageMaker

Apache MXNet and TensorFlow

HLS-based video playback

Custom video processing

Output

Real-time and batch-driven ML applications and video processing and playback services use the Kinesis Video Streams API to access and retrieve indexed video

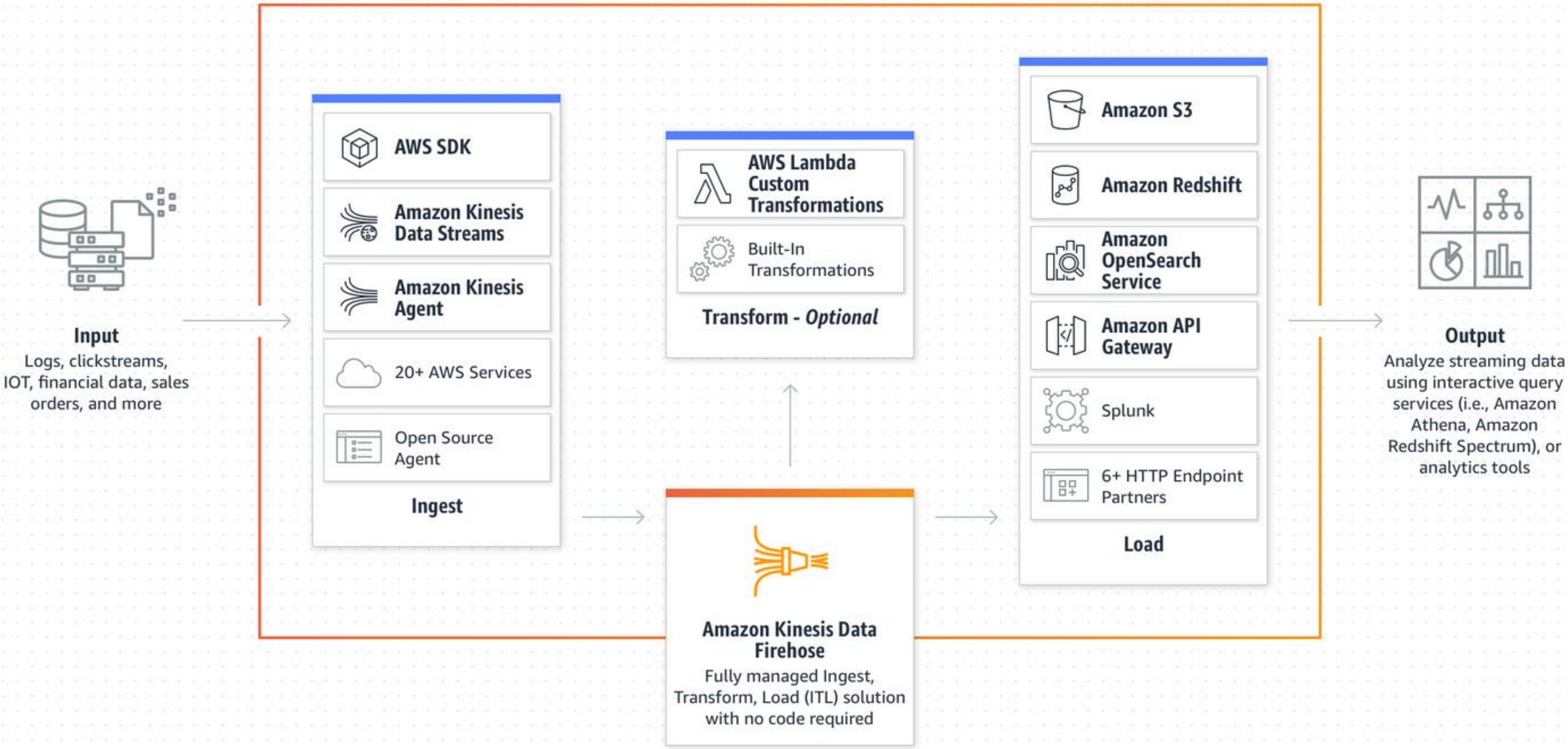
Amazon Rekognition Video is a cloud-based computer vision service.

TensorFlow and Apache MXNet are cloud-based deep learning libraries.

HLS = HTTP Live Streaming, an HTTP-based live streaming protocol.

https://aws.amazon.com/kinesis/

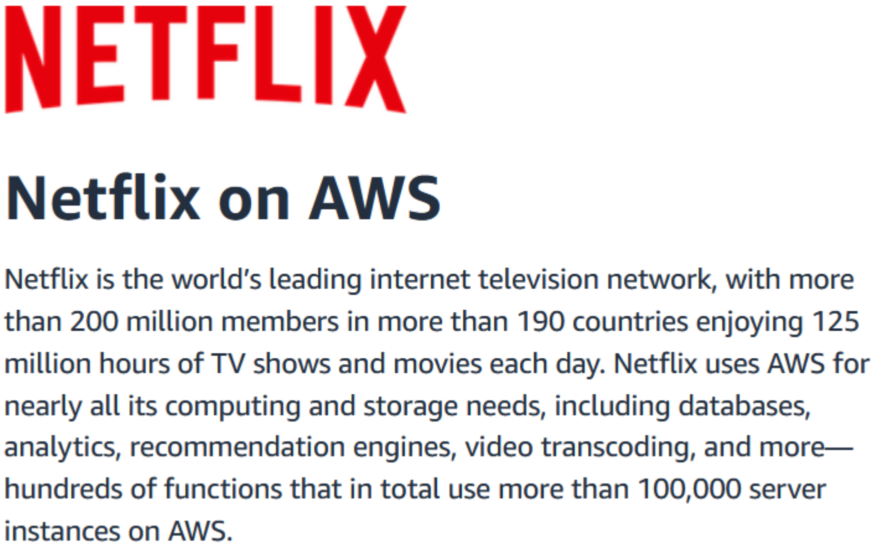
Case study of commercial data pipelines: A more complex AWS data pipeline



Input
Logs, clickstreams, IOT, financial data, sales orders, and more
AWS SDK
Amazon Kinesis Data Streams
Amazon Kinesis Agent
20+ AWS Services
Open Source Agent
Ingest
AWS Lambda Custom Transformations
Built-In Transformations
Transform - *Optional*
Amazon Kinesis Data Firehose
Fully managed Ingest, Transform, Load (ITL) solution with no code required
Amazon S3
Amazon Redshift
Amazon OpenSearch Service
Amazon API Gateway
Splunk
6+ HTTP Endpoint Partners
Load
Output
Analyze streaming data using interactive query services (i.e., Amazon Athena, Amazon Redshift Spectrum), or analytics tools

https://aws.amazon.com/kinesis/data-firehose/?nc=sn&loc=1

Who are using AWS streaming data pipelines?



NETFLIX

Netflix on AWS

Netflix is the world's leading internet television network, with more than 200 million members in more than 190 countries enjoying 125 million hours of TV shows and movies each day. Netflix uses AWS for nearly all its computing and storage needs, including databases, analytics, recommendation engines, video transcoding, and more—hundreds of functions that in total use more than 100,000 server instances on AWS.

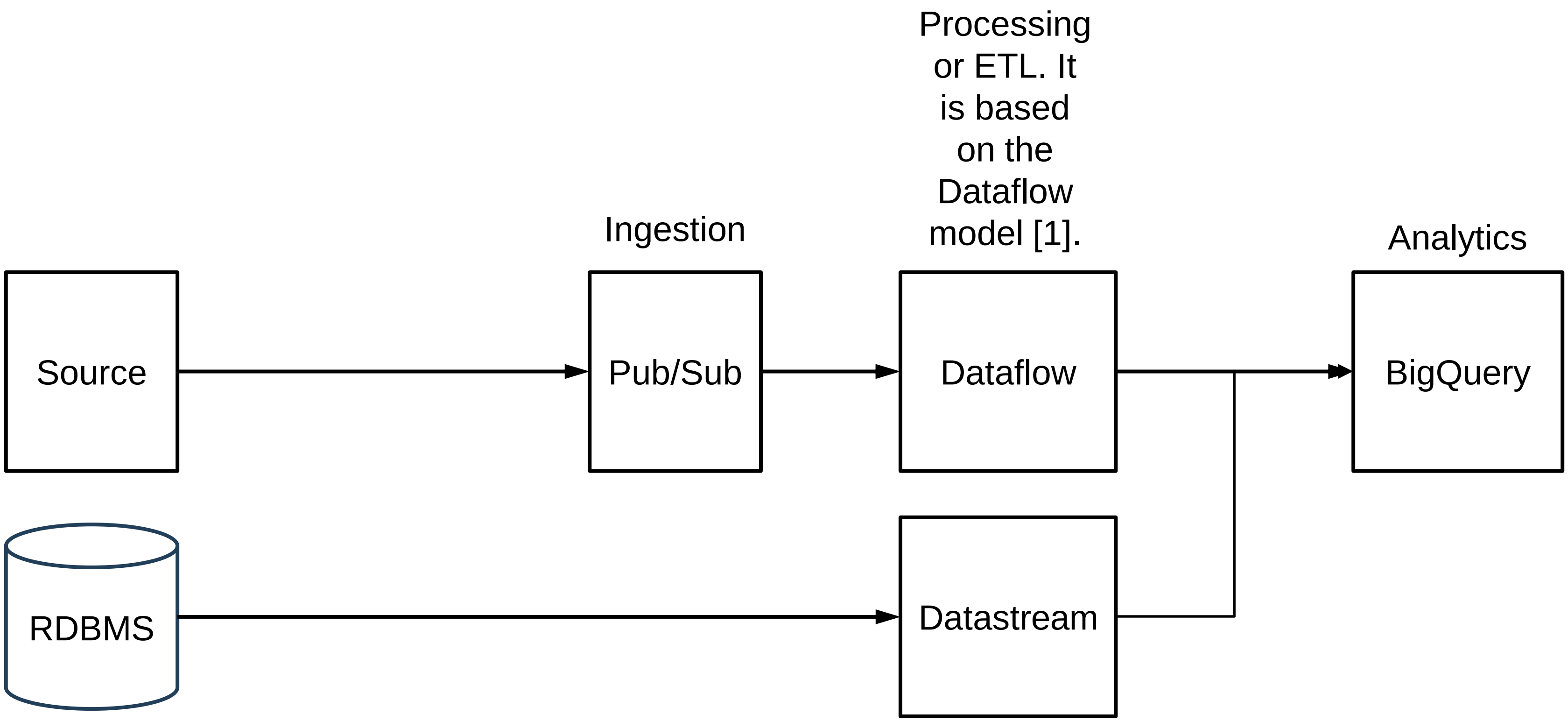
WYZE

Wyze Scales to Support Millions of Connected Home Devices Using AWS Services

2021

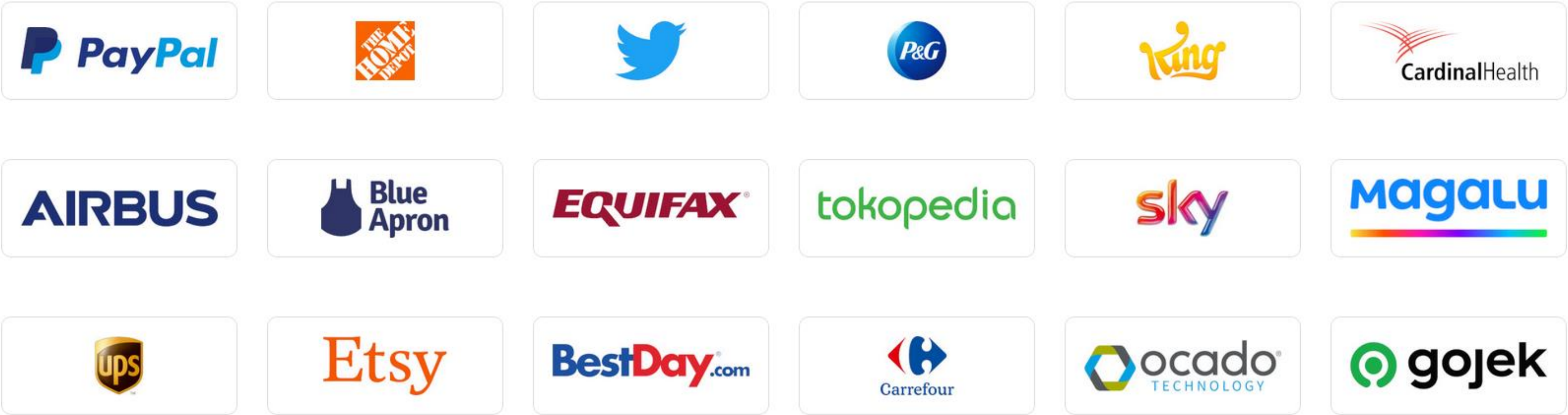
https://aws.amazon.com/solutions/case-studies/innovators/netflix/
https://aws.amazon.com/solutions/case-studies/wyze/

Case study of more commercial data pipelines: Google Cloud streaming data services



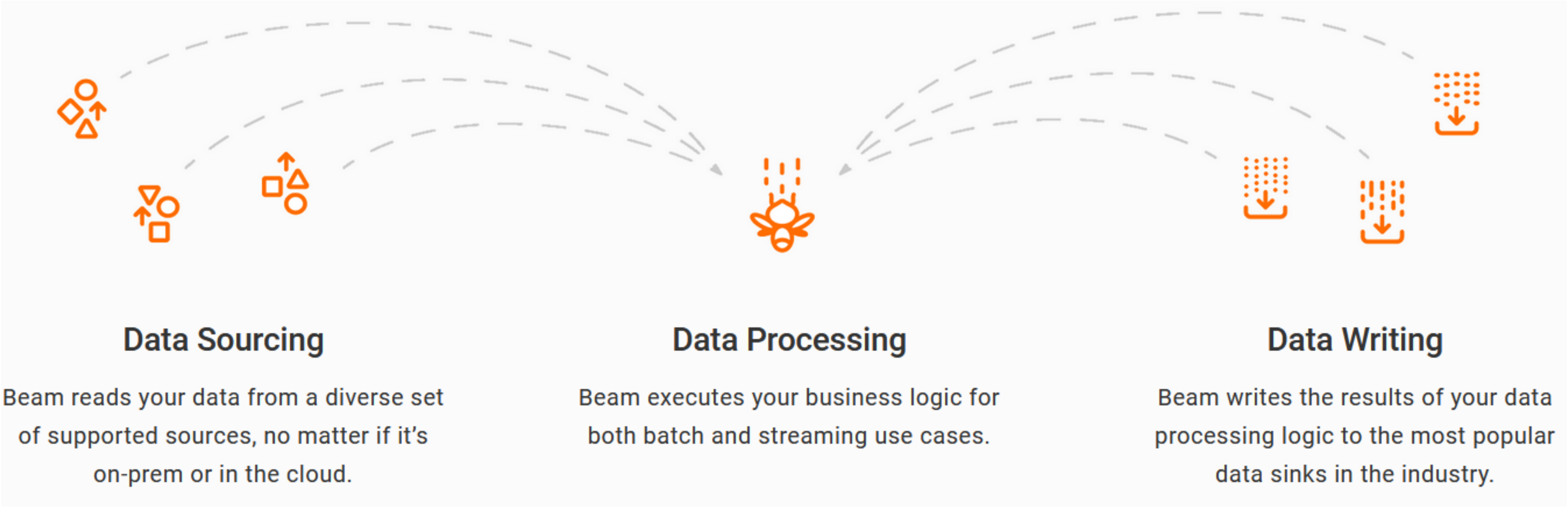
https://cloud.google.com/solutions/stream-analytics
[1] http://www.vldb.org/pvldb/vol8/p1792-Akidau.pdf

Who are using Google Cloud streaming data pipelines?



https://cloud.google.com/customers

Case study of more commercial data pipelines: Apache Beam



Data Sourcing

Beam reads your data from a diverse set of supported sources, no matter if it's on-prem or in the cloud.

Data Processing

Beam executes your business logic for both batch and streaming use cases.

Data Writing

Beam writes the results of your data processing logic to the most popular data sinks in the industry.

Apache Beam is an open-source implementation of the Dataflow model [1].

https://beam.apache.org/

[1] http://www.vldb.org/pvldb/vol8/p1792-Akidau.pdf

Case study of more commercial data pipelines: Apache Beam (contd.)

Portable

Execute pipelines on multiple execution environments (runners), providing flexibility and avoiding lock-in.

Write Once, Run Anywhere

Batch processing

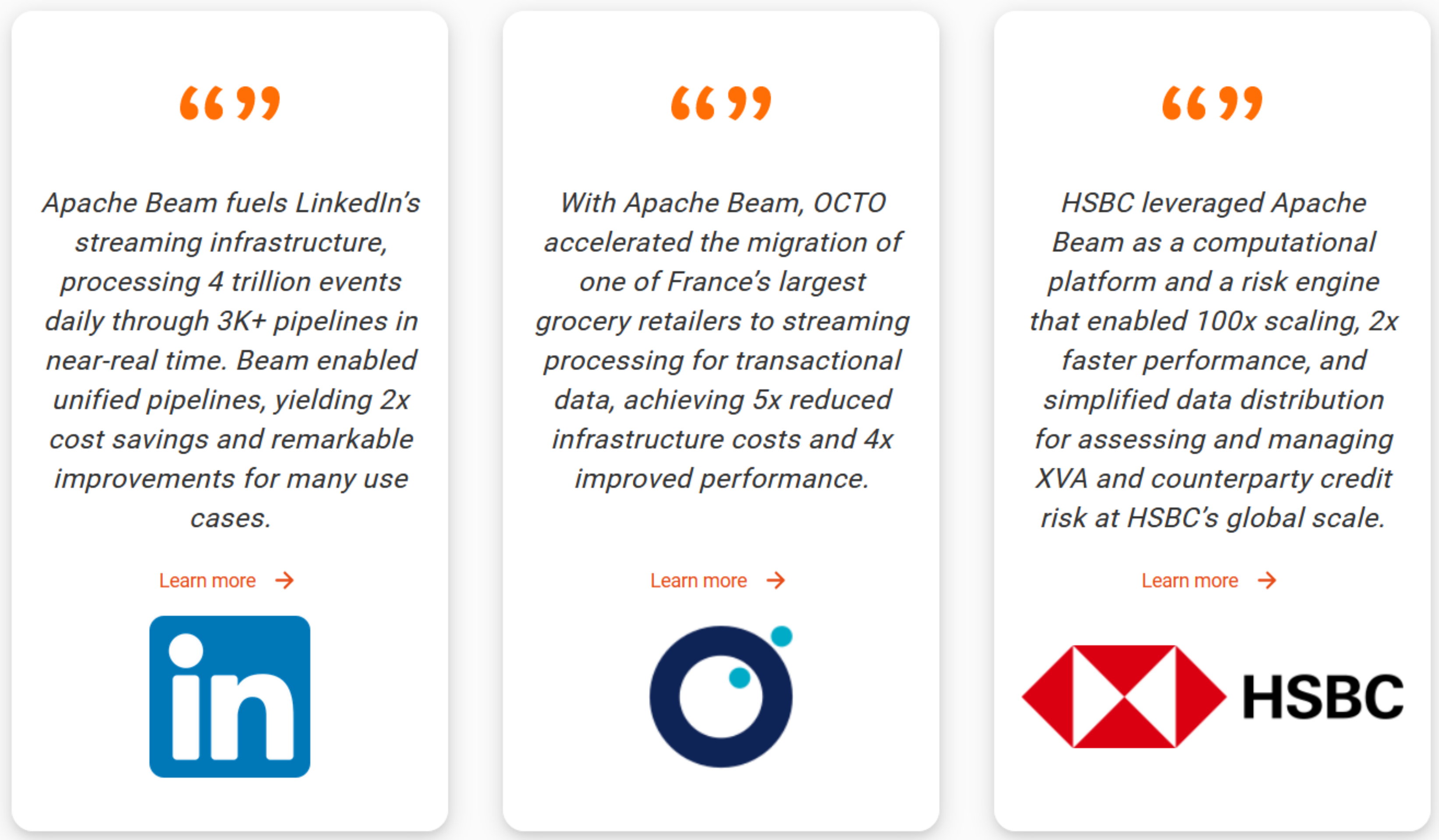
- Apache Flink
- Apache Spark
- Google Dataflow
- Apache Samza
- Apache Twister2
- Amazon Kinesis Data Analytics

Create Multi-language Pipelines

python™ Java GO TS Scala SQL

https://beam.apache.org/

Who are using Apache Beam data pipelines?



Apache Beam fuels LinkedIn's streaming infrastructure, processing 4 trillion events daily through 3K+ pipelines in near-real time. Beam enabled unified pipelines, yielding 2x cost savings and remarkable improvements for many use cases.

Learn more →

With Apache Beam, OCTO accelerated the migration of one of France's largest grocery retailers to streaming processing for transactional data, achieving 5x reduced infrastructure costs and 4x improved performance.

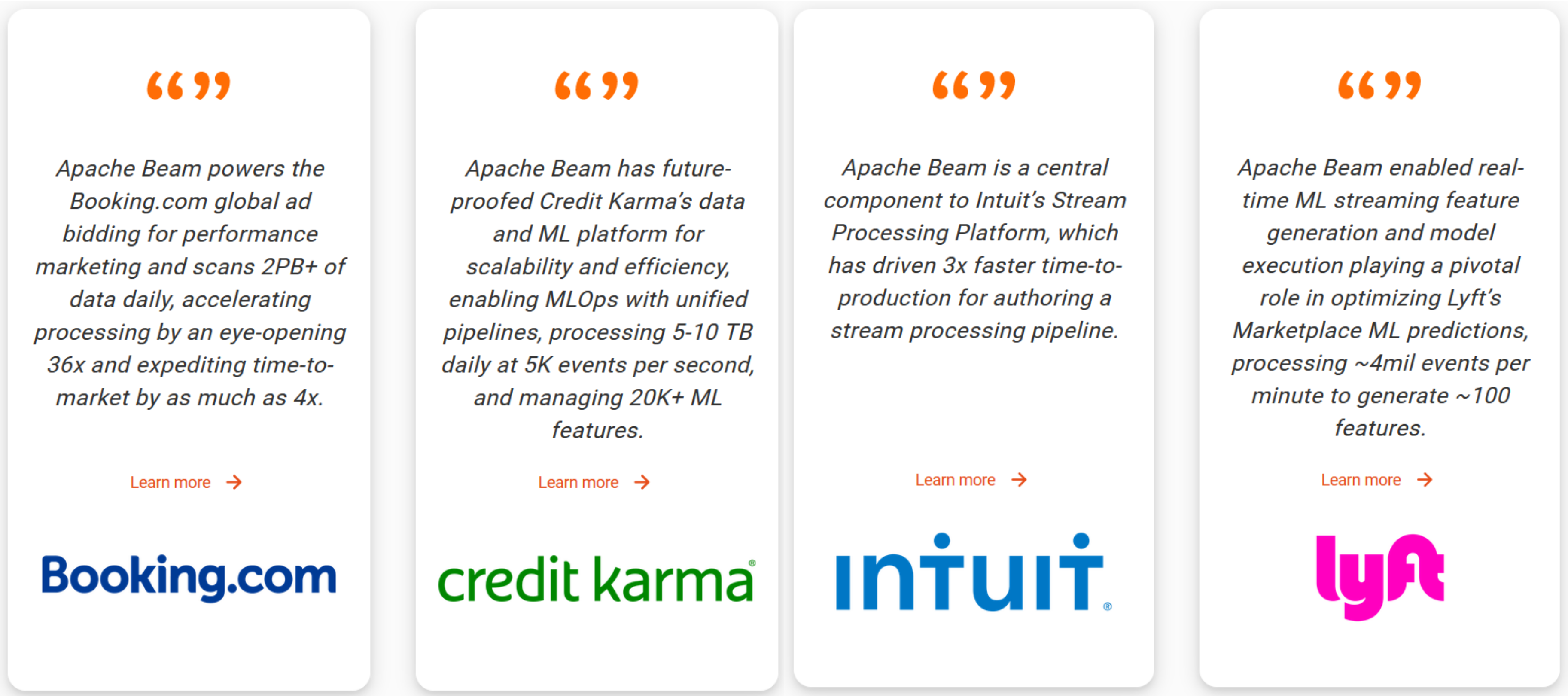
Learn more →

HSBC leveraged Apache Beam as a computational platform and a risk engine that enabled 100x scaling, 2x faster performance, and simplified data distribution for assessing and managing XVA and counterparty credit risk at HSBC's global scale.

Learn more →

https://beam.apache.org/

Who are using Apache Beam data pipelines? (contd.)



Apache Beam powers the Booking.com global ad bidding for performance marketing and scans 2PB+ of data daily, accelerating processing by an eye-opening 36x and expediting time-to-market by as much as 4x.

Apache Beam has future-proofed Credit Karma's data and ML platform for scalability and efficiency, enabling MLOps with unified pipelines, processing 5-10 TB daily at 5K events per second, and managing 20K+ ML features.

Apache Beam is a central component to Intuit's Stream Processing Platform, which has driven 3x faster time-to-production for authoring a stream processing pipeline.

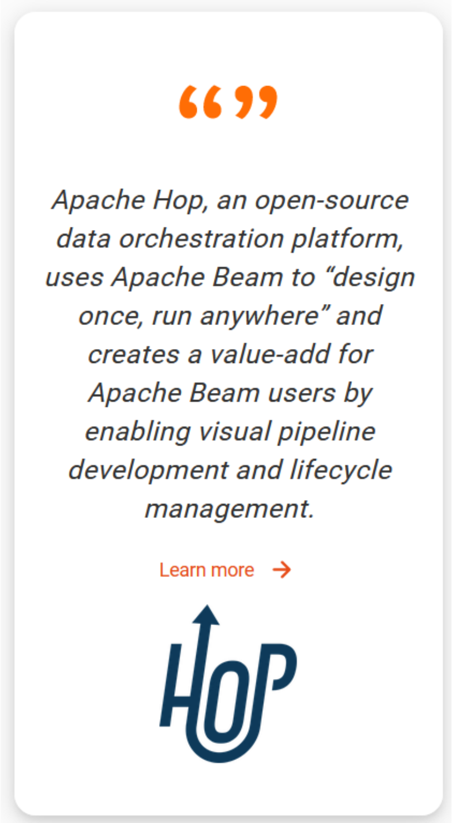
Apache Beam enabled real-time ML streaming feature generation and model execution playing a pivotal role in optimizing Lyft's Marketplace ML predictions, processing ~4mil events per minute to generate ~100 features.

https://beam.apache.org/

Who are using Apache Beam data pipelines? (contd.)

Extensible

Apache Beam is extensible, with projects such as TensorFlow Extended and Apache Hop built on top of Apache Beam.

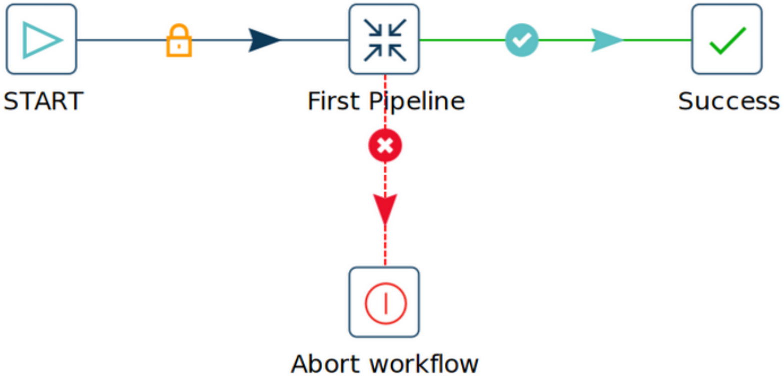


Apache Hop, an open-source data orchestration platform, uses Apache Beam to “design once, run anywhere” and creates a value-add for Apache Beam users by enabling visual pipeline development and lifecycle management.

Visual Development

Hop workflows and pipelines are developed visually through an intuitive drag and drop interface.

Visual development allows data developers and data engineers to keep focus on the business logic that needs to be implemented, on *what* needs to be done instead of *how* it needs to be done.

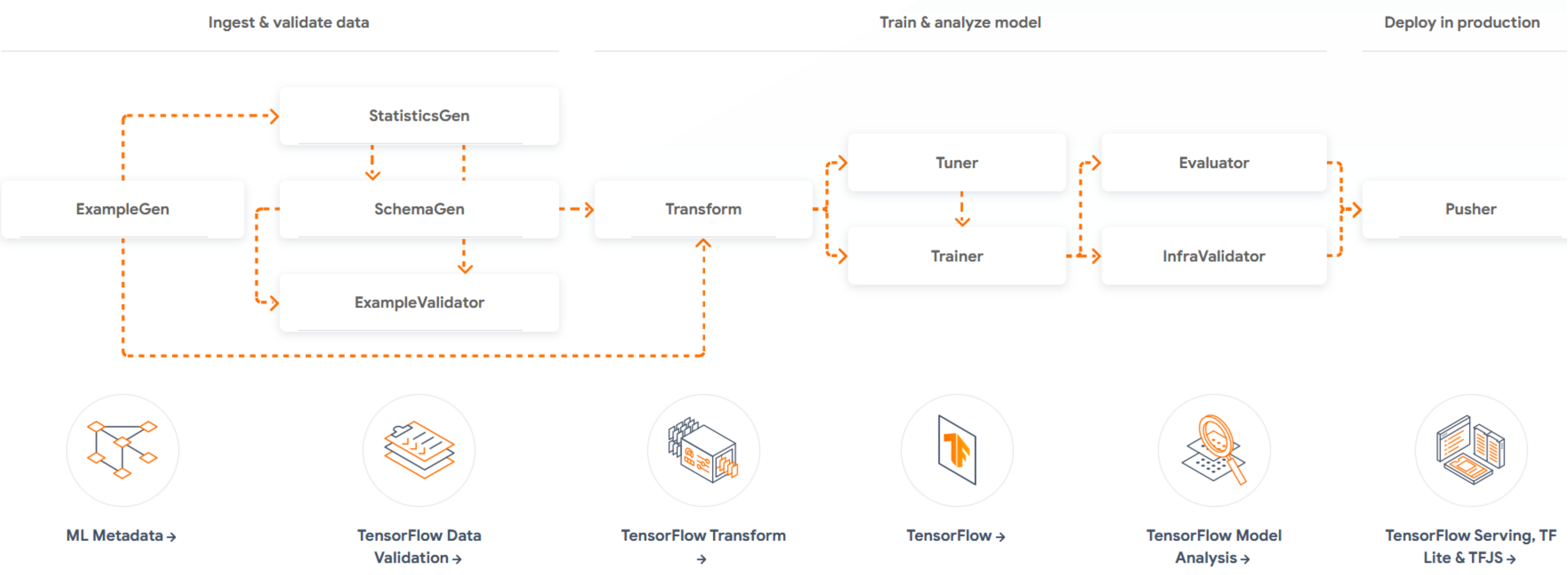


https://beam.apache.org/ , https://hop.apache.org/ , https://hop.apache.org/manual/latest/workflow/create-workflow.html

Who are using Apache Beam data pipelines? (contd.)

TensorFlow Extended (TFX)
TensorFlow is an ML library. On the other hand, a TFX pipeline is an end-to-end data pipeline for training and deploying models in high-performance production environments.

TFX also allows developers to use the TFX libraries of a particular **component** (say, Transform) **individually**. Thus, we can utilize TFX libraries to create a component into a different pipeline, such as the standard Apache Beam pipeline.



https://www.tensorflow.org/tfx

Who are using TensorFlow Extended (TFX)?

Spotify

Provides personalized recommendations to users using a pipeline bult with TFX and Kuberflow

AIRBUS

Provides malware protection for email attachments. The pipeline trains a distinct ML model for each file type (such as PDF or DOC).

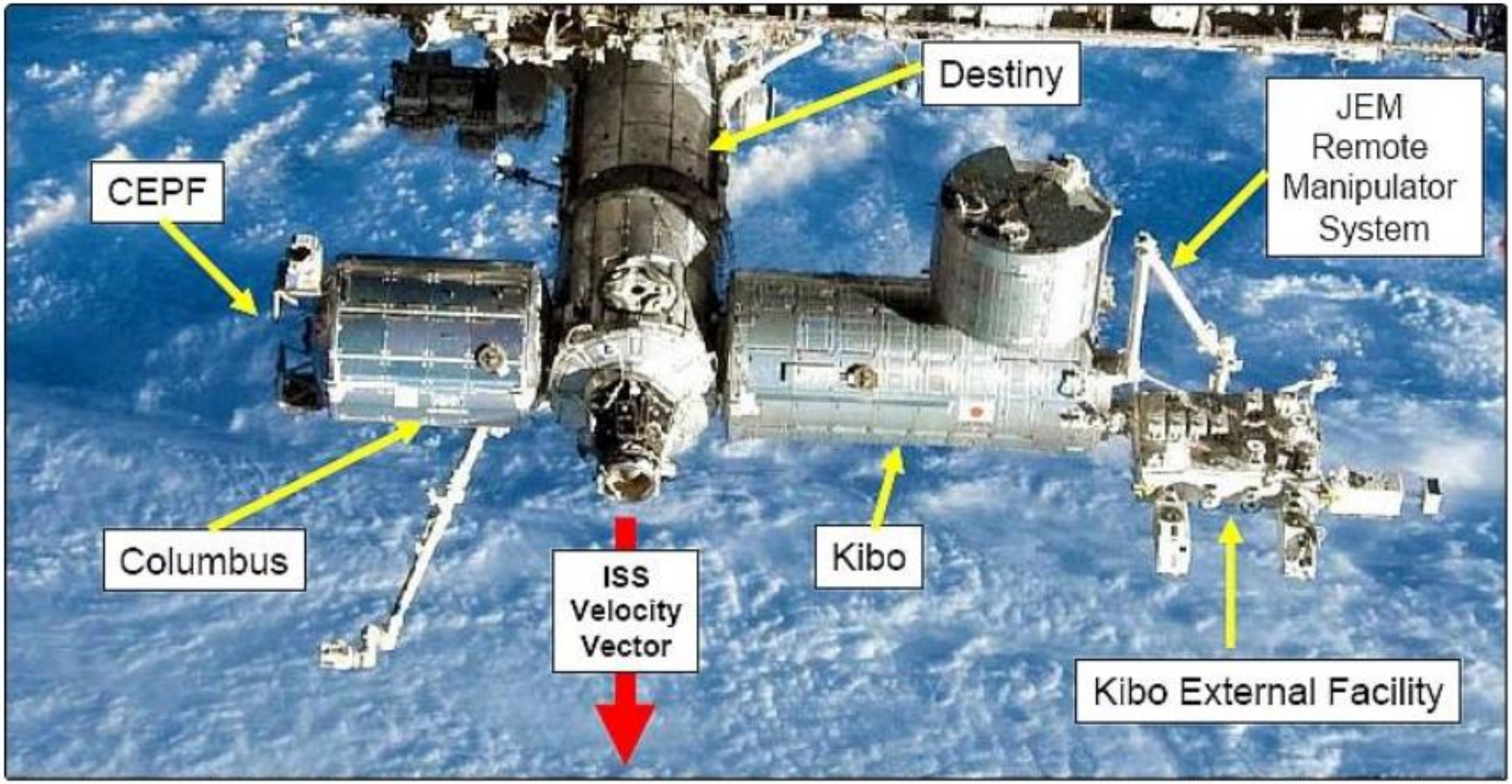
OpenX®

Provides targeted ads using a pipeline bult with TFX and Google Cloud Platform. The pipeline processes 1M+ requests per second and serves each request within 15 ms.

https://www.tensorflow.org/tfx
https://security.googleblog.com/2020/02/improving-malicious-document-detection.html

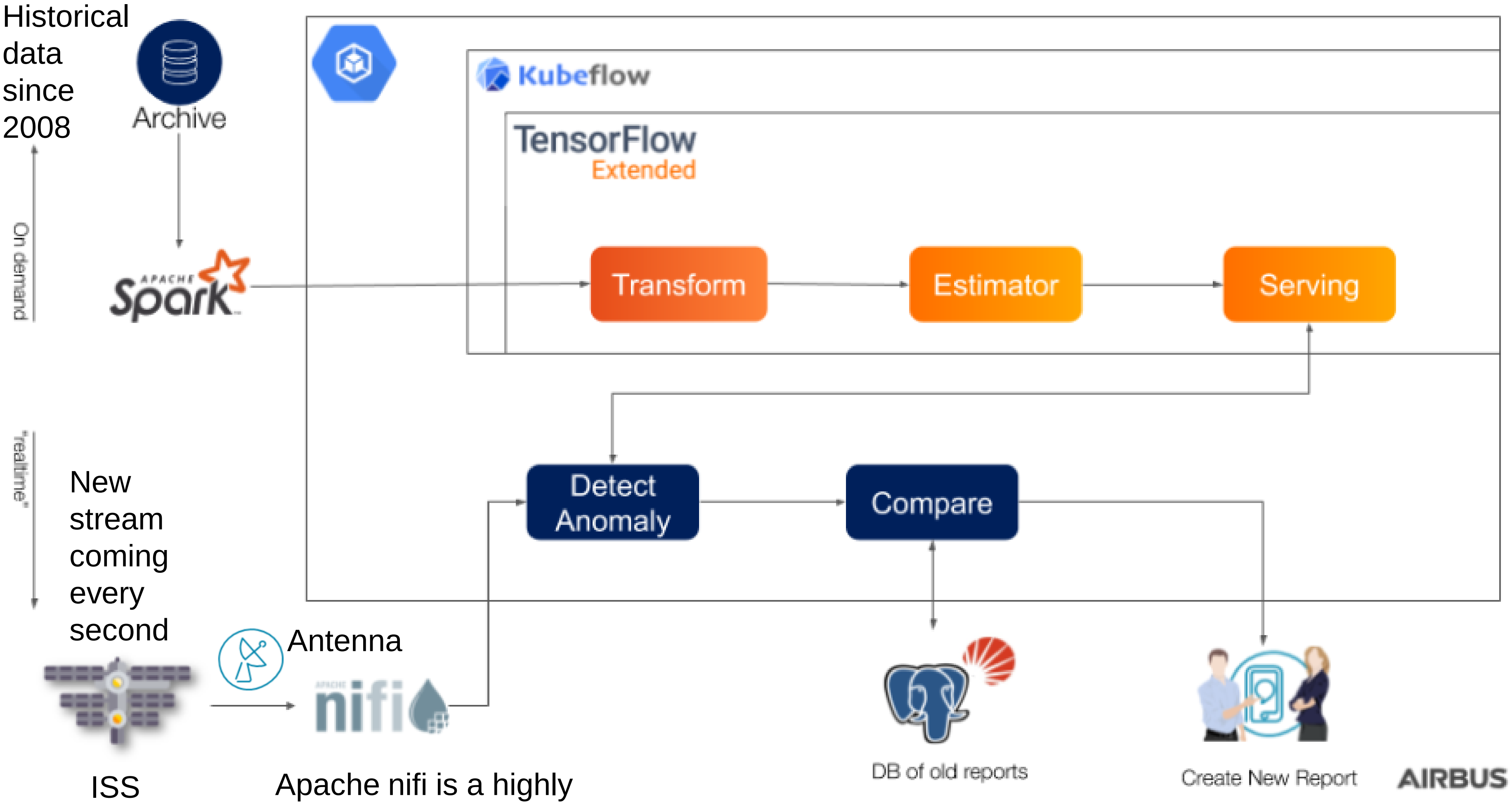
Who are using TensorFlow Extended (TFX)? (contd.)

Airbus has built the Columbus (laboratory) module of the International Space Station (ISS) in 2008. To ensure the health of the crew as well as hundreds of systems onboard the Columbus module, measurements of about 17,000 telemetry parameters are beamed to earth in 1 second intervals.



https://blog.tensorflow.org/2020/04/how-airbus-detects-anomalies-iss-telemetry-data-tfx.html

Who are using TensorFlow Extended (TFX)? (contd.)



Historical data since 2008

New stream coming every second

Antenna

ISS

Apache nifi is a highly configurable data loader like Amazon Kinesis Firehose

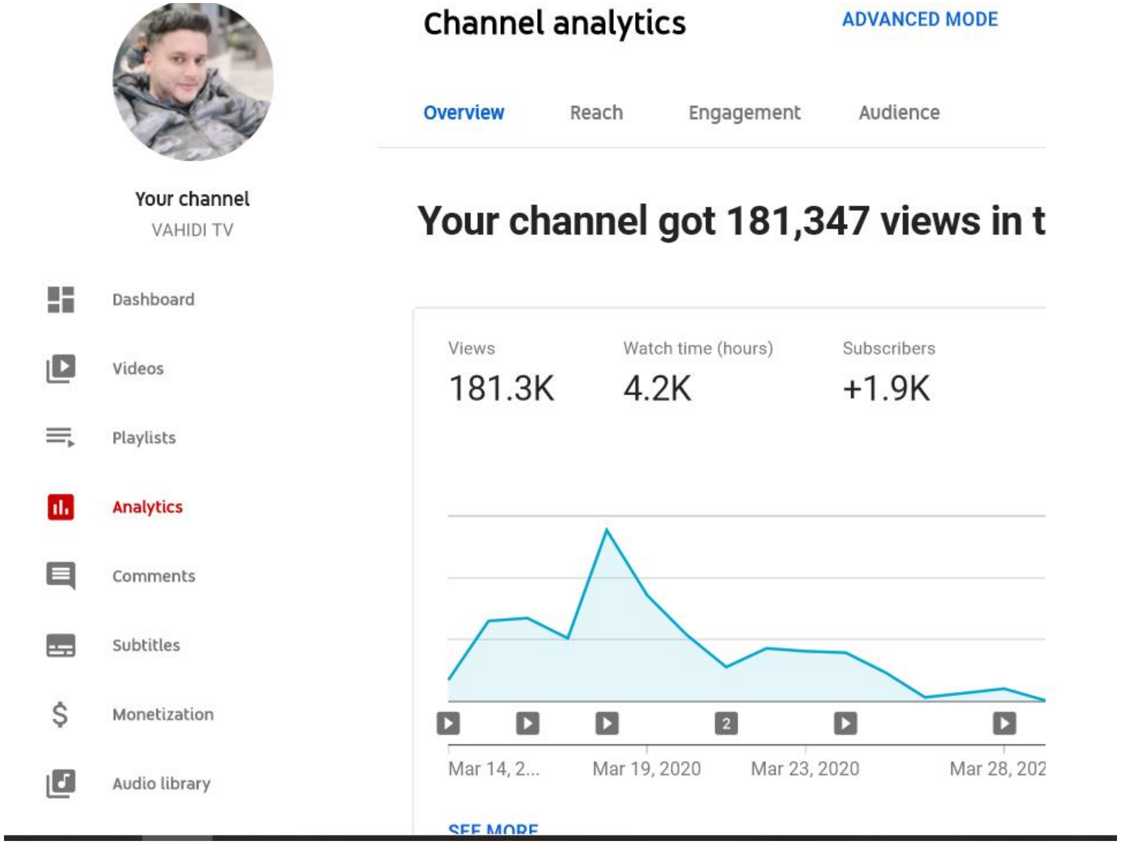
https://blog.tensorflow.org/2020/04/how-airbus-detects-anomalies-iss-telemetry-data-tfx.html

More commercial data pipelines

- Microsoft Azure Stream Analytics, etc.

Streaming data analytics

- YouTube channel analytics
- Instagram Insights



Your channel got 181,347 views in t

Views	Watch time (hours)	Subscribers
181.3K	4.2K	+1.9K

Image courtesy: https://b2shop.icu/ProductDetail.aspx?iid=187924013&pr=80.88

What is streaming data analytics?

Analyzing the streaming data to generate **business intelligence**

- Find patterns (such as trends)

- Spot exceptions (e.g., automatically detect fraudulent credit card transactions)

- Discover opportunities to improve the product/service
- Identify risks

https://www.youtube.com/watch?v=MbEfiX4sMXc

What is streaming data analytics?

The traditional tools and techniques used for analyzing **historical transactional data** are not always applicable on the **streaming data**. For streaming data analytics, we need new tools and techniques.

For example, in stock markets, we can not wait until the end of the day to run analytics. Instead, we need a **dynamic dashboard** providing us with a **minute-by-minute analysis**.

https://www.youtube.com/watch?v=MbEfiX4sMXc

In-memory analytics

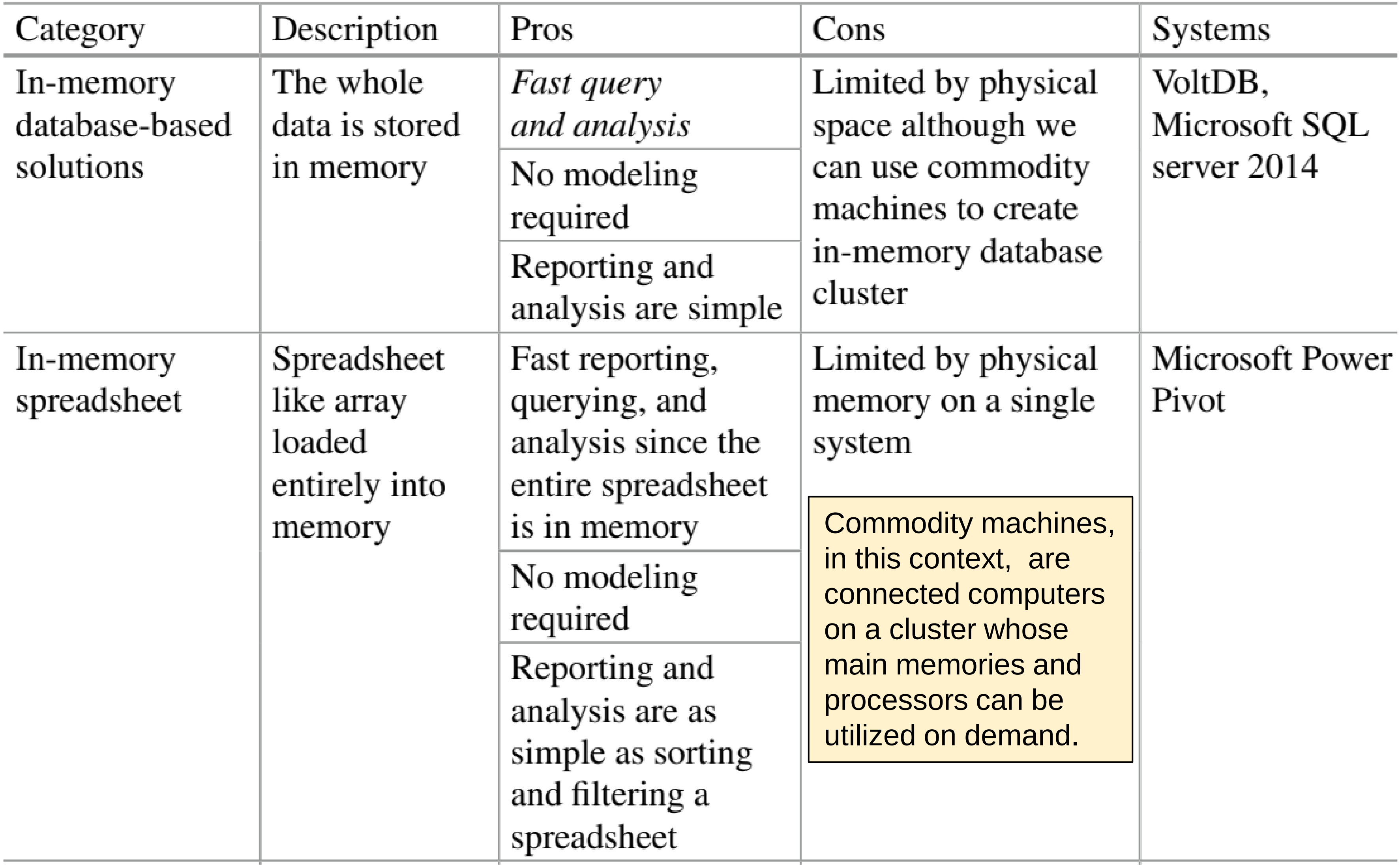
We need to produce **real-time analysis***.

Hence, it is preferred that data is **stored and analyzed in memory.**

Databases providing in-memory storage and analytics are known as **in-memory databases**. They usually utilize disks only as archives or persistent storages.

* real-time analysis = Near real-time analysis

Examples of in-memory databases

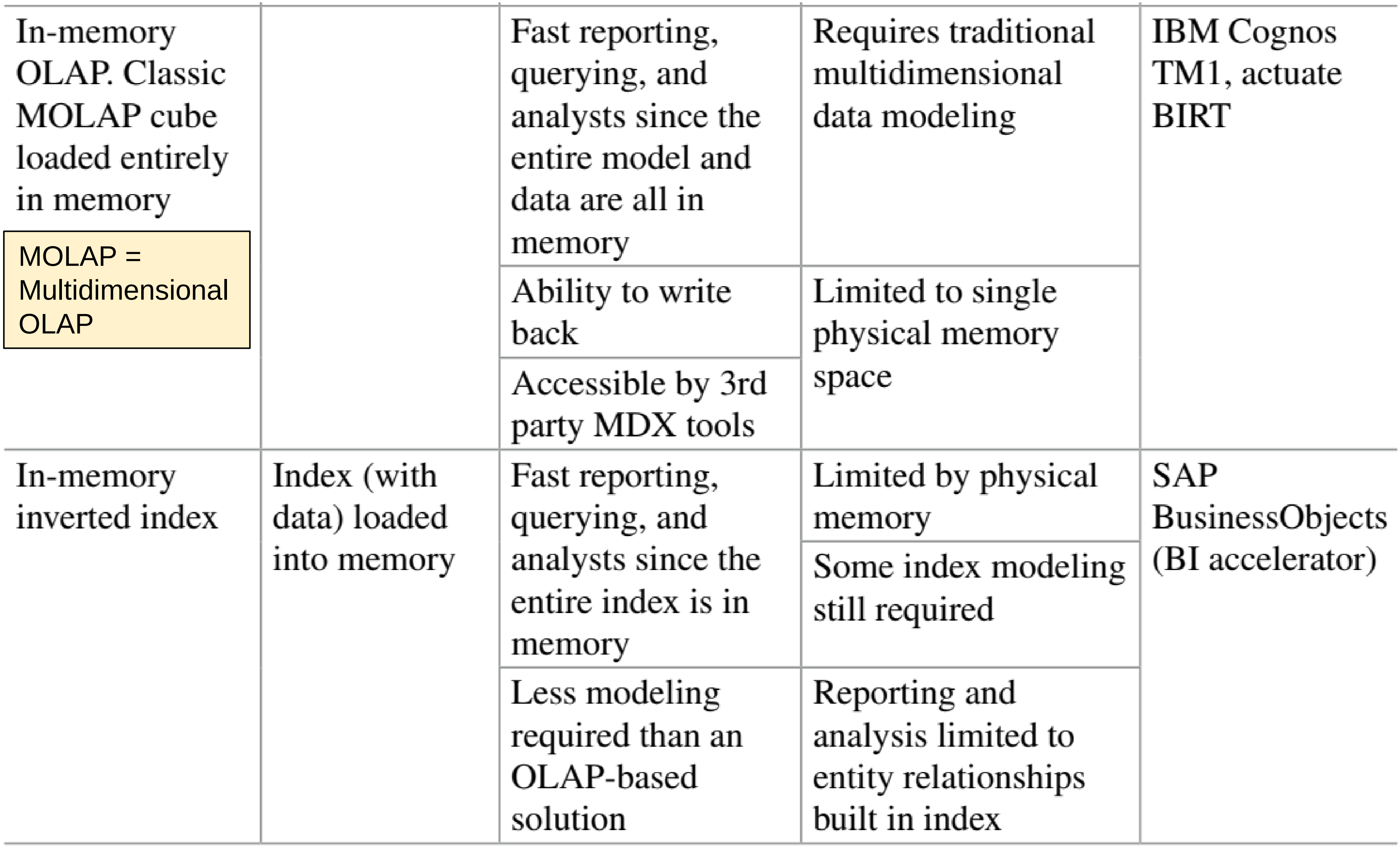


Category	Description	Pros	Cons	Systems
In-memory database-based solutions	The whole data is stored in memory	*Fast query and analysis* No modeling required Reporting and analysis are simple	Limited by physical space although we can use commodity machines to create in-memory database cluster	VoltDB, Microsoft SQL server 2014
In-memory spreadsheet	Spreadsheet like array loaded entirely into memory	Fast reporting, querying, and analysis since the entire spreadsheet is in memory No modeling required Reporting and analysis are as simple as sorting and filtering a spreadsheet	Limited by physical memory on a single system	Microsoft Power Pivot

Commodity machines, in this context, are connected computers on a cluster whose main memories and processors can be utilized on demand.

Table 8.1, Raj and Raman

Examples of in-memory databases (contd.)



In-memory OLAP. Classic MOLAP cube loaded entirely in memory		Fast reporting, querying, and analysts since the entire model and data are all in memory	Requires traditional multidimensional data modeling	IBM Cognos TM1, actuate BIRT
		Ability to write back	Limited to single physical memory space	
		Accessible by 3rd party MDX tools		
In-memory inverted index	Index (with data) loaded into memory	Fast reporting, querying, and analysts since the entire index is in memory	Limited by physical memory Some index modeling still required	SAP BusinessObjects (BI accelerator)
		Less modeling required than an OLAP-based solution	Reporting and analysis limited to entity relationships built in index	

MOLAP = Multidimensional OLAP

Table 8.1, Raj and Raman

SAP HANA

“SAP HANA is a modern, **in-memory database** and platform that is deployable **on premise** or **in the cloud**.

The SAP HANA platform is a flexible data source-agnostic in-memory data platform that allows you to analyze large volumes of data **in real time**. Using the database services of the SAP HANA platform, you can store and access data in-memory and column-based. SAP HANA allows online transaction processing (**OLTP**) and online analytical processing (**OLAP**) on one system, **without the need for redundant data storage or aggregates**. Using the application services of the SAP HANA platform, you can develop applications, run your custom applications built on SAP HANA, and manage their lifecycles.”

https://help.sap.com/docs/SAP_HANA_PLATFORM/eb3777d5495d46c5b2fa773206bbfb46/d3b1adcdbb571014a93eff11ad9a1d89.html

SAP HANA appliance

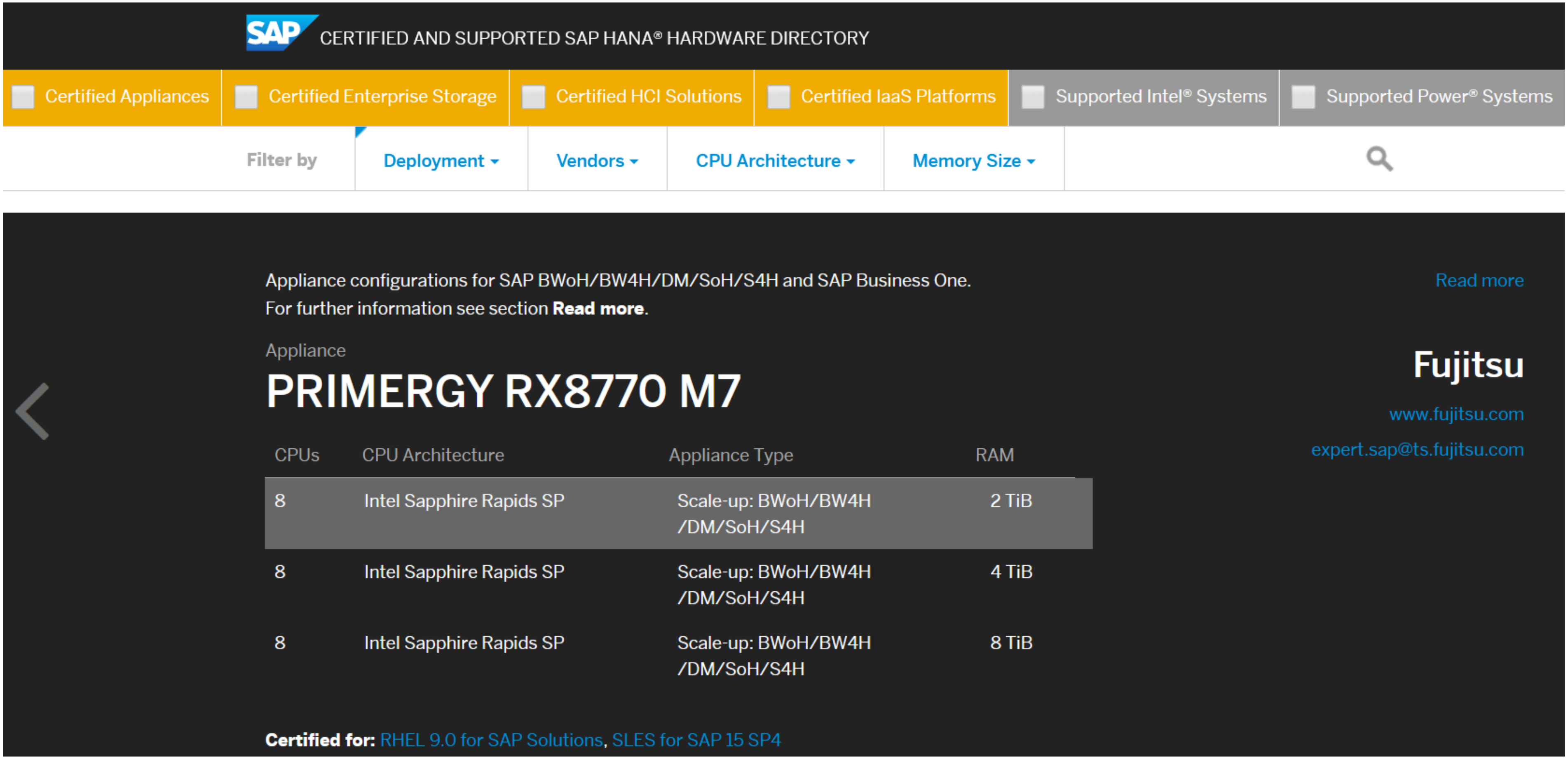
SAP HANA comes as an **appliance** combining software components from SAP **optimized on proven hardware** provided by **SAP's hardware partners (e.g., HP, Dell)**.

"This approach offers you well-defined hardware designed for the performance needs of an in-memory **solution out of the box**. The appliance delivery is the first choice if you are looking for a **preconfigured hardware** set-up and a **preinstalled software** package for a fast implementation done by your chosen hardware partner and fully supported by both, the partner and SAP.

You can decide to implement SAP HANA using the appliance delivery model, meaning preconfigured software and hardware bundled by an SAP hardware partner, or you can opt for the SAP HANA tailored data center integration approach, which allows you more flexibility when **integrating your SAP HANA system with your existing storage solution**. For more information see SAP HANA Tailored Data Center Integration."

https://help.sap.com/docs/SAP_HANA_PLATFORM/eb3777d5495d46c5b2fa773206bbfb46/e2ebe79cf9a248ee9dda83cef123863f.html

Example of a SAP HANA certified appliance



SAP
CERTIFIED AND SUPPORTED SAP HANA® HARDWARE DIRECTORY
□ Certified Appliances
□ Certified Enterprise Storage
□ Certified HCI Solutions
□ Certified IaaS Platforms
□ Supported Intel® Systems
□ Supported Power® Systems
Filter by
Deployment ▾
Vendors ▾
CPU Architecture ▾
Memory Size ▾
Read more

Appliance configurations for SAP BWoH/BW4H/DM/SoH/S4H and SAP Business One.
For further information see section **Read more**.

Appliance

PRIMERGY RX8770 M7

Fujitsu

www.fujitsu.com

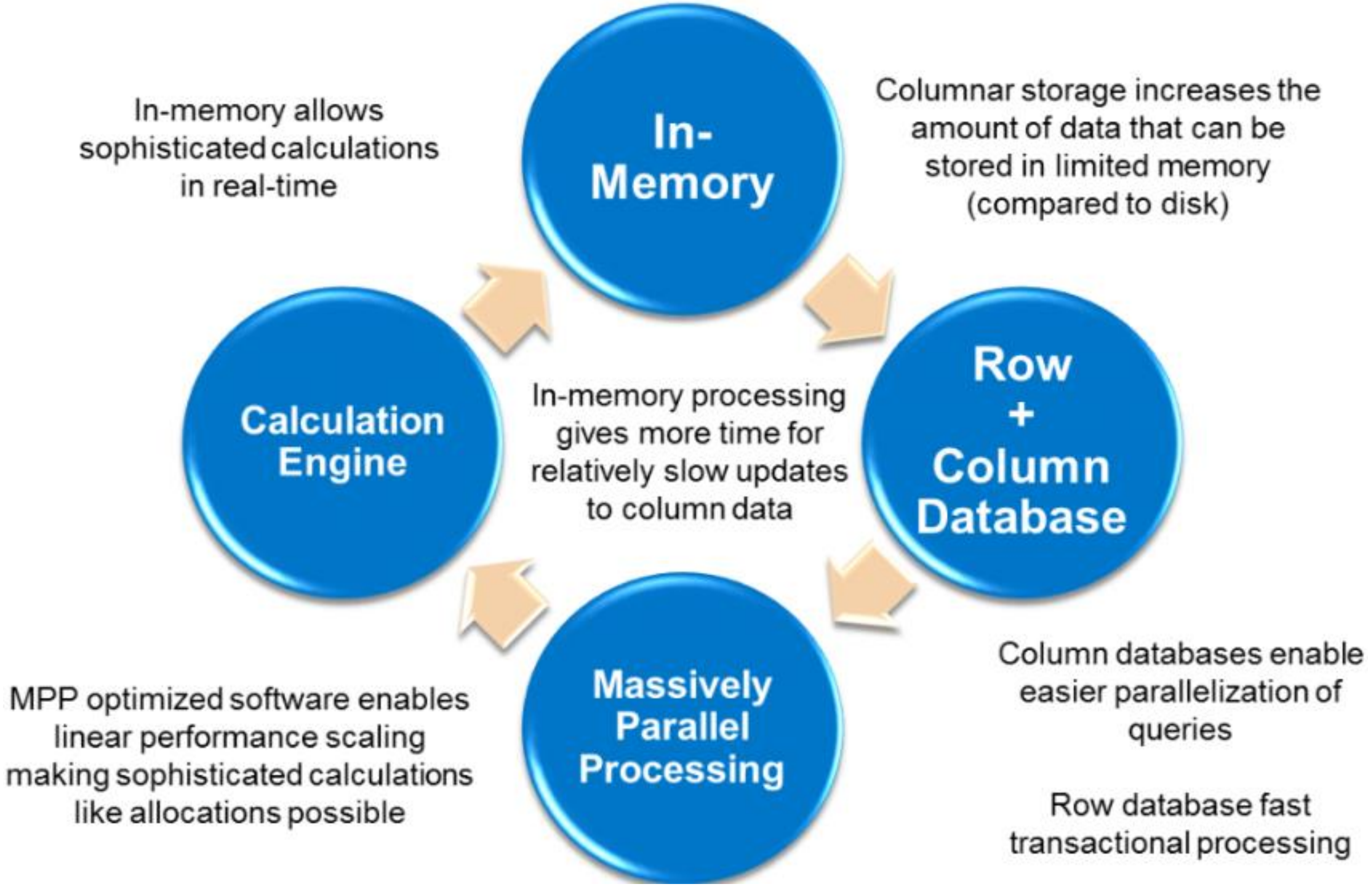
expert.sap@ts.fujitsu.com

CPUs	CPU Architecture	Appliance Type	RAM
8	Intel Sapphire Rapids SP	Scale-up: BWoH/BW4H /DM/SoH/S4H	2 TiB
8	Intel Sapphire Rapids SP	Scale-up: BWoH/BW4H /DM/SoH/S4H	4 TiB
8	Intel Sapphire Rapids SP	Scale-up: BWoH/BW4H /DM/SoH/S4H	8 TiB

Certified for: RHEL 9.0 for SAP Solutions, SLES for SAP 15 SP4

https://www.sap.com/dmc/exp/2014-09-02-hana-hardware/enEN/#/solutions?filters=v:deCertified&id=s:2749

Example of a SAP HANA certified appliance



In-Memory

Row + Column Database

Massively Parallel Processing

Calculation Engine

In-memory allows sophisticated calculations in real-time

Columnar storage increases the amount of data that can be stored in limited memory (compared to disk)

In-memory processing gives more time for relatively slow updates to column data

Column databases enable easier parallelization of queries

Row database fast transactional processing

MPP optimized software enables linear performance scaling making sophisticated calculations like allocations possible

https://bi-insider.com/portfolio-item/sap-hana-platform-technical-overview/

SAP HANA features

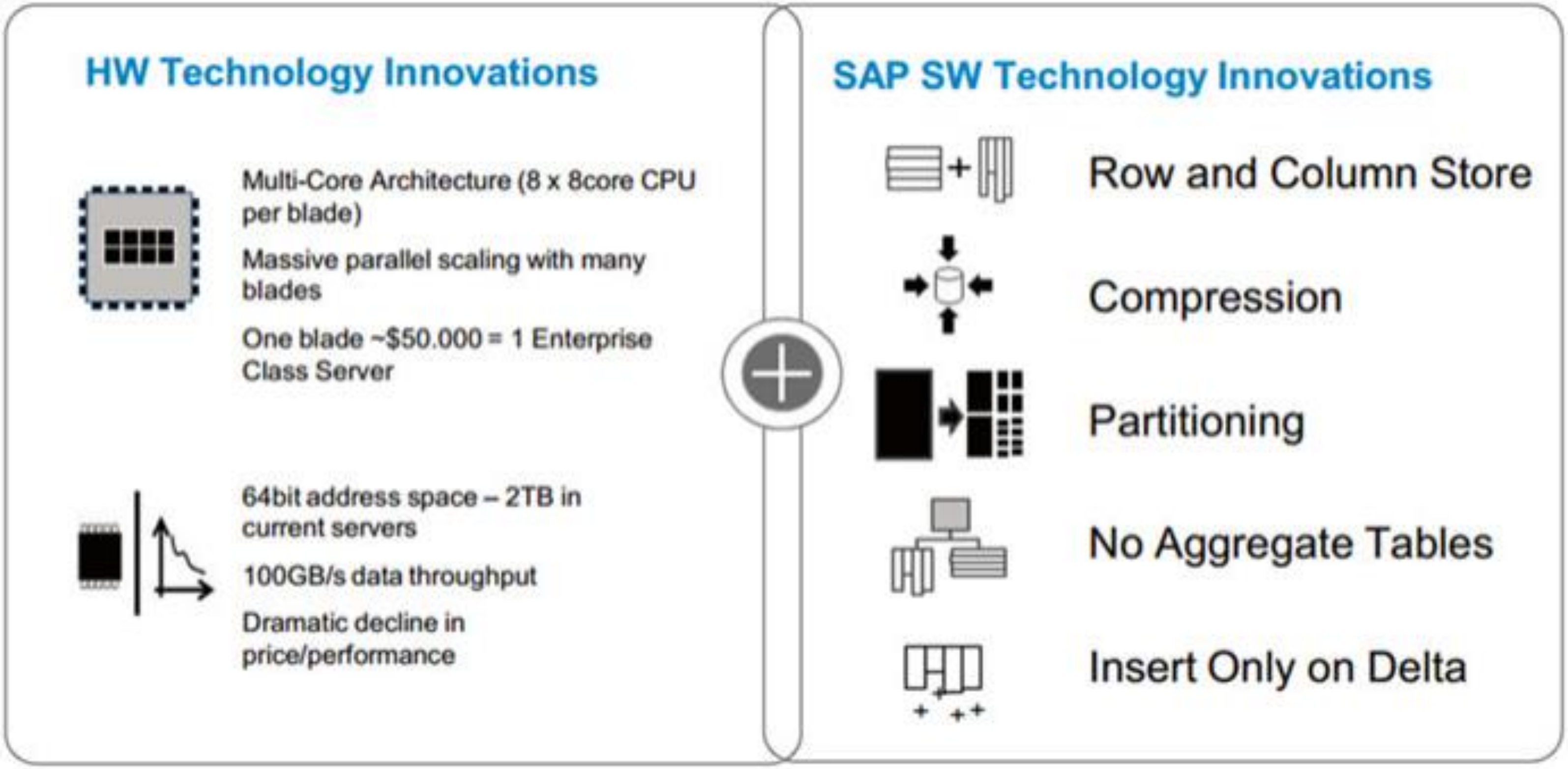

HW Technology Innovations
Multi-Core Architecture (8 x 8core CPU per blade)
Massive parallel scaling with many blades
One blade ~$50.000 = 1 Enterprise Class Server
64bit address space – 2TB in current servers
100GB/s data throughput
Dramatic decline in price/performance
SAP SW Technology Innovations
Row and Column Store
Compression
Partitioning
No Aggregate Tables
Insert Only on Delta

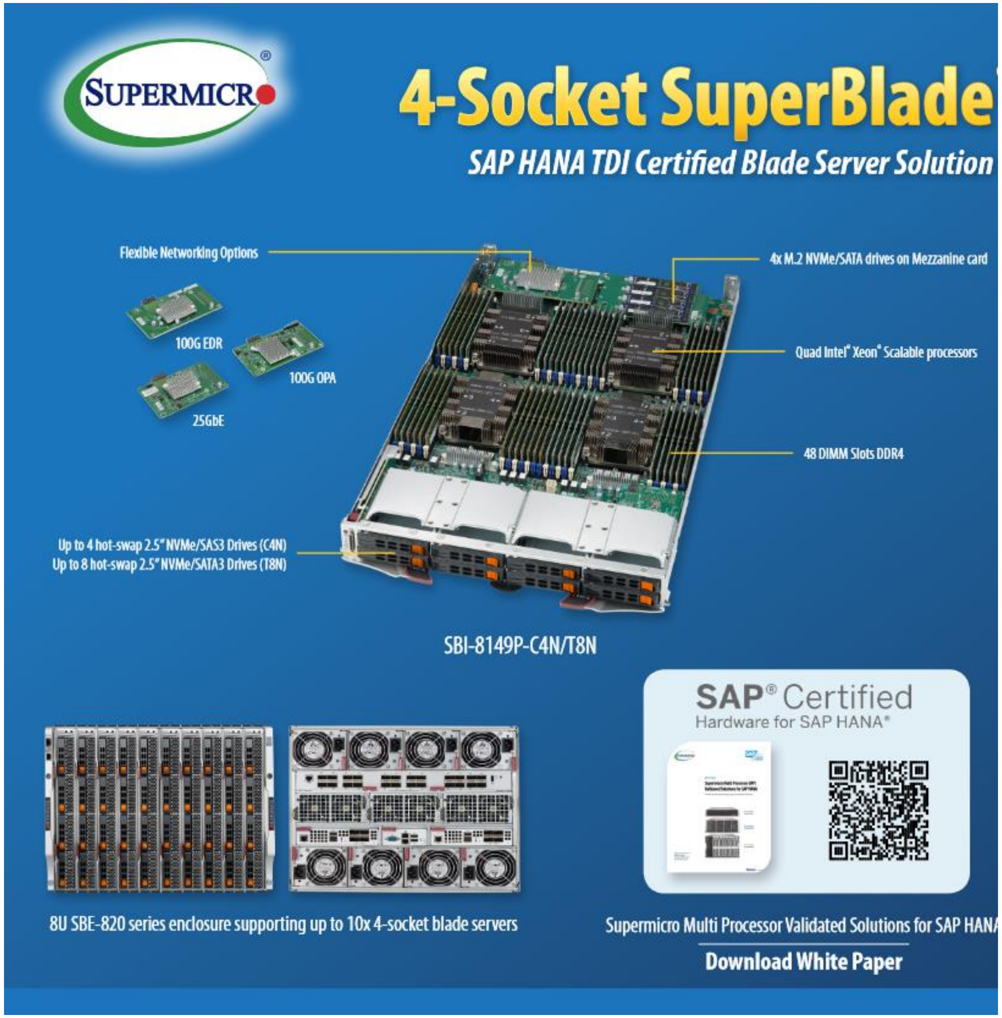
Fig. 8.6 SAP HANA features

What is a blade (computing server)?

Server Type	Definition	Use Case
Rack Server	An encased server used to **stack** and install several servers in a large closet.	Midsize to large businesses with on-premises server closets where space is limited and computing power is necessary for high-end applications.
Blade Server	An encased server used to insert **small servers** into a blade bay where several can be stacked horizontally in a rack.	Midsize to large businesses with on-premises server closets where space is limited, but several servers are necessary to handle high-end applications.
Tower Server	A **stand-alone** computer that looks like a standard desktop but has additional server resources installed in the machine.	Small businesses or home networks that need a server to store files or manage network resources, but only one server is needed and scalability isn't a concern.

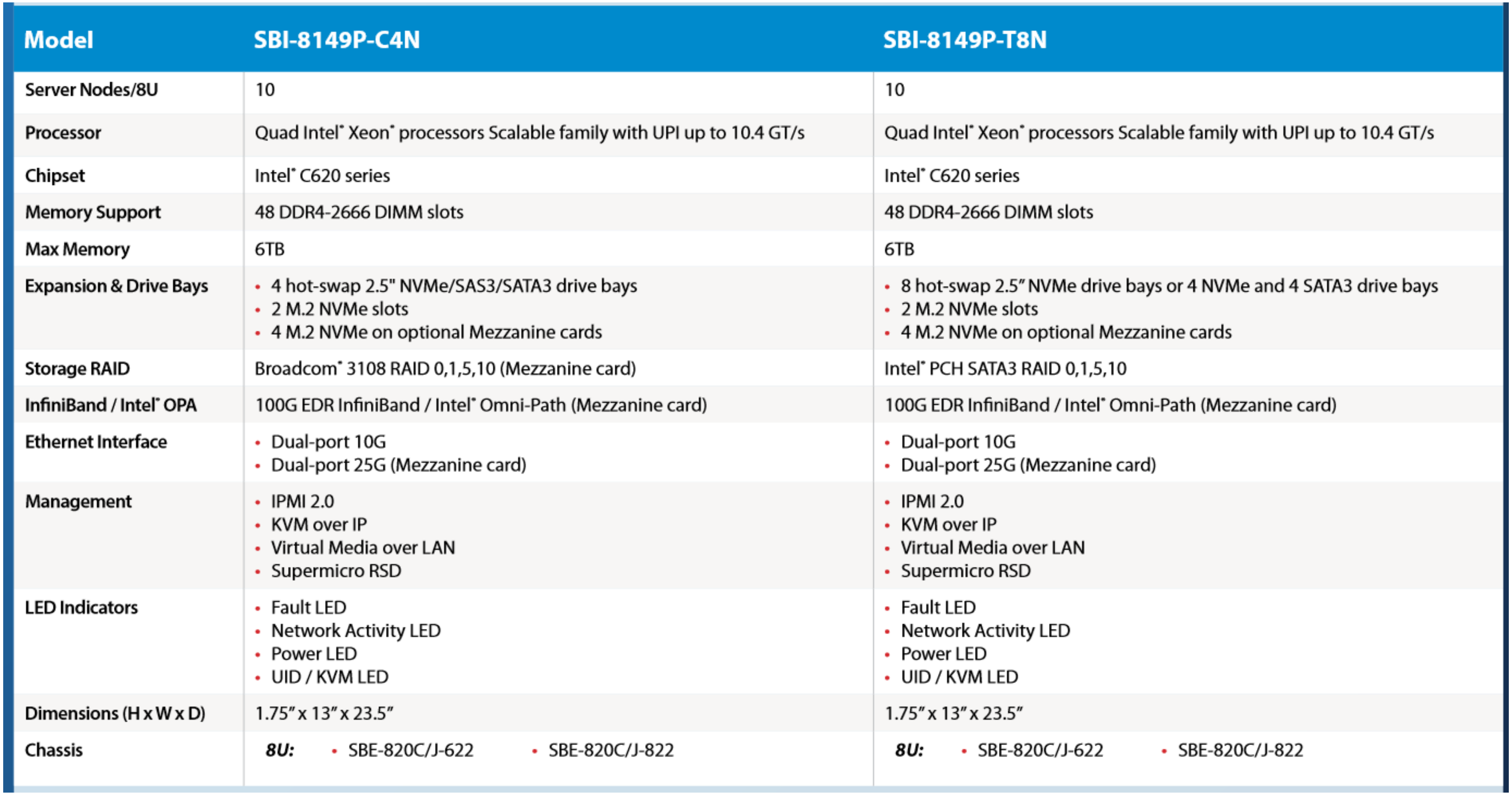
https://blog.purestorage.com/purely-informational/blade-server-vs-rack-server-vs-tower-server/ , https://www.router-switch.com/faq/tower-server-vs-rack-server.html , https://tekbloq.com/2016/05/26/whats-the-difference-rack-blade-or-tower-server/ , https://www.lenovo.com/in/en/data-center/servers/towers/ThinkSystem-ST250-Server/p/77XX7TRST25

Example of a SAP HANA certified blade server


SUPERMICRO
4-Socket SuperBlade
SAP HANA TDI Certified Blade Server Solution
Flexible Networking Options
100G EDR
100G OPA
25GbE
4x M.2 NVMe/SATA drives on Mezzanine card
Quad Intel® Xeon® Scalable processors
48 DIMM Slots DDR4
Up to 4 hot-swap 2.5" NVMe/SAS3 Drives (C4N)
Up to 8 hot-swap 2.5" NVMe/SATA3 Drives (T8N)
SBI-8149P-C4N/T8N
SAP® Certified
Hardware for SAP HANA®
8U SBE-820 series enclosure supporting up to 10x 4-socket blade servers
Supermicro Multi Processor Validated Solutions for SAP HANA
Download White Paper

https://www.supermicro.com/datasheet/datasheet_8U-Blade_SAP.pdf

Example of a SAP HANA certified blade server (contd.)



Model	SBI-8149P-C4N	SBI-8149P-T8N
Server Nodes/8U	10	10
Processor	Quad Intel® Xeon® processors Scalable family with UPI up to 10.4 GT/s	Quad Intel® Xeon® processors Scalable family with UPI up to 10.4 GT/s
Chipset	Intel® C620 series	Intel® C620 series
Memory Support	48 DDR4-2666 DIMM slots	48 DDR4-2666 DIMM slots
Max Memory	6TB	6TB
Expansion & Drive Bays	• 4 hot-swap 2.5" NVMe/SAS3/SATA3 drive bays • 2 M.2 NVMe slots • 4 M.2 NVMe on optional Mezzanine cards	• 8 hot-swap 2.5" NVMe drive bays or 4 NVMe and 4 SATA3 drive bays • 2 M.2 NVMe slots • 4 M.2 NVMe on optional Mezzanine cards
Storage RAID	Broadcom® 3108 RAID 0,1,5,10 (Mezzanine card)	Intel® PCH SATA3 RAID 0,1,5,10
InfiniBand / Intel® OPA	100G EDR InfiniBand / Intel® Omni-Path (Mezzanine card)	100G EDR InfiniBand / Intel® Omni-Path (Mezzanine card)
Ethernet Interface	• Dual-port 10G • Dual-port 25G (Mezzanine card)	• Dual-port 10G • Dual-port 25G (Mezzanine card)
Management	• IPMI 2.0 • KVM over IP • Virtual Media over LAN • Supermicro RSD	• IPMI 2.0 • KVM over IP • Virtual Media over LAN • Supermicro RSD
LED Indicators	• Fault LED • Network Activity LED • Power LED • UID / KVM LED	• Fault LED • Network Activity LED • Power LED • UID / KVM LED
Dimensions (H x W x D)	1.75" x 13" x 23.5"	1.75" x 13" x 23.5"
Chassis	*8U:* • SBE-820C/J-622 • SBE-820C/J-822	*8U:* • SBE-820C/J-622 • SBE-820C/J-822

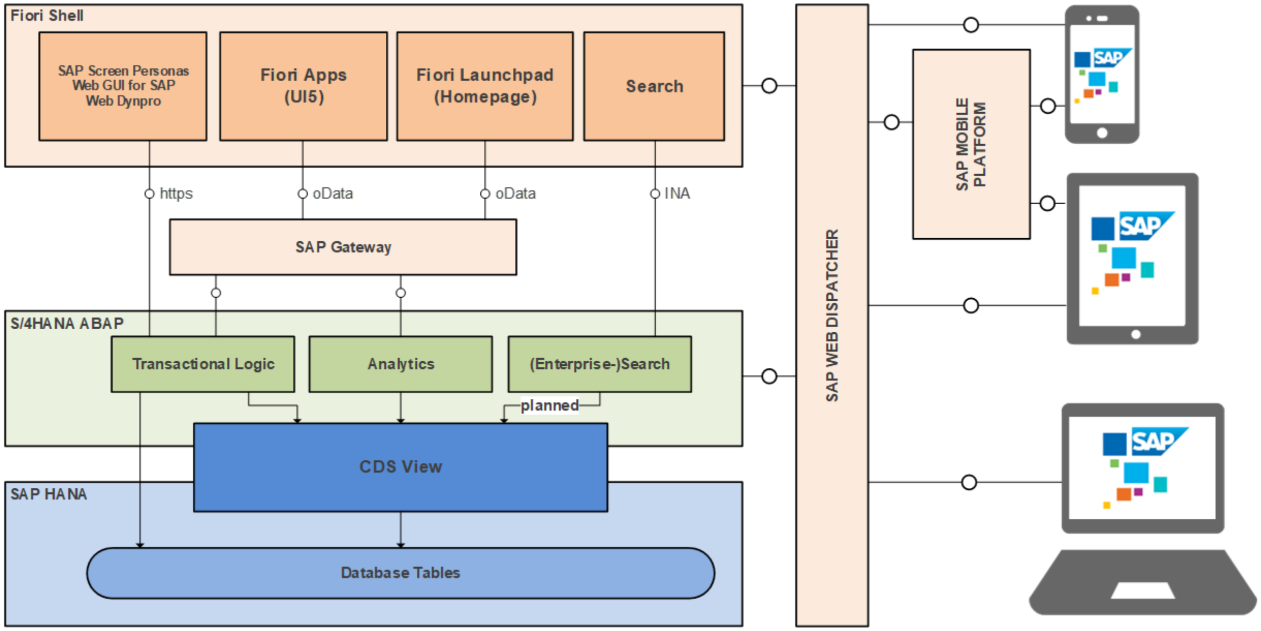
https://www.supermicro.com/datasheet/datasheet_8U-Blade_SAP.pdf

SAP S/4HANA is an ERP built on top of SAP HANA

Launchpad / Fiori
(SAPGUI for Windows still available for compatibility reasons on-premise)

Application engines

Data structures
(Compatibility provided through Core Data Services)



CDS = Core data services

https://erpiseasy.com/2021/03/20/what-is-sap-s4hana/

Kognitio architecture

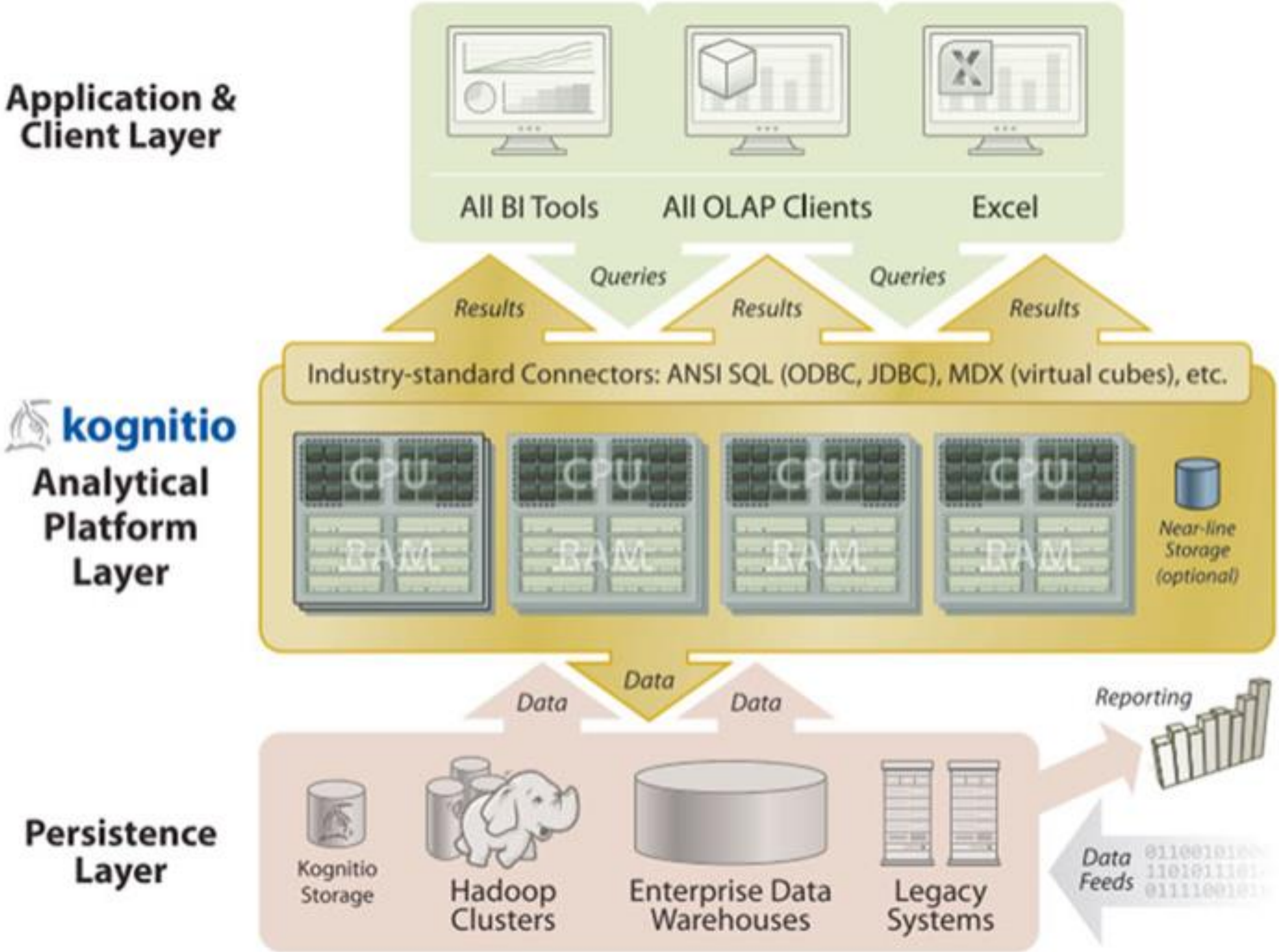


Fig. 8.7 Kognitio analytical platform architectural view

Predictive analytics

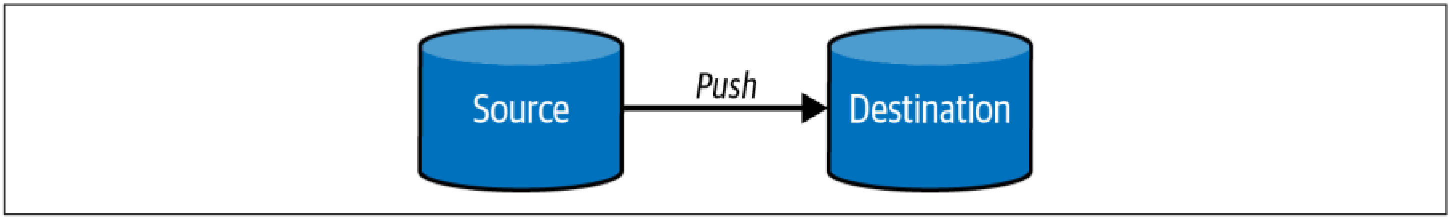
- Definition
- Examples
  - Deciding whether to give loan to a customer
  - Determining the amount of a car insurance premium
- Bias, discrimination, feedback loops (false stereotyping)
- Responsibility and accountability
- ‘Systems thinking’

Section ‘Predictive Analytics’, Chapter 12, Kleppmann

Techniques involved in streaming data pipelines: Events

- Event = Record (and timestamp)
- The producer/publisher/sender sends a **message** containing an **event** to the consumer/subscriber/recipient
- In other words, the consumer **ingests** messages containing new events from the producer

How are messages ingested?



Push message notifications

Figure 7-7. Pushing data from source to destination

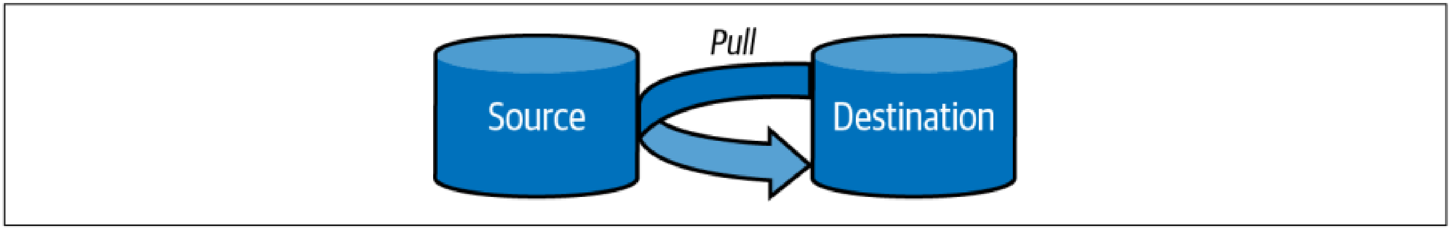


Figure 7-8. A destination pulling data from a source

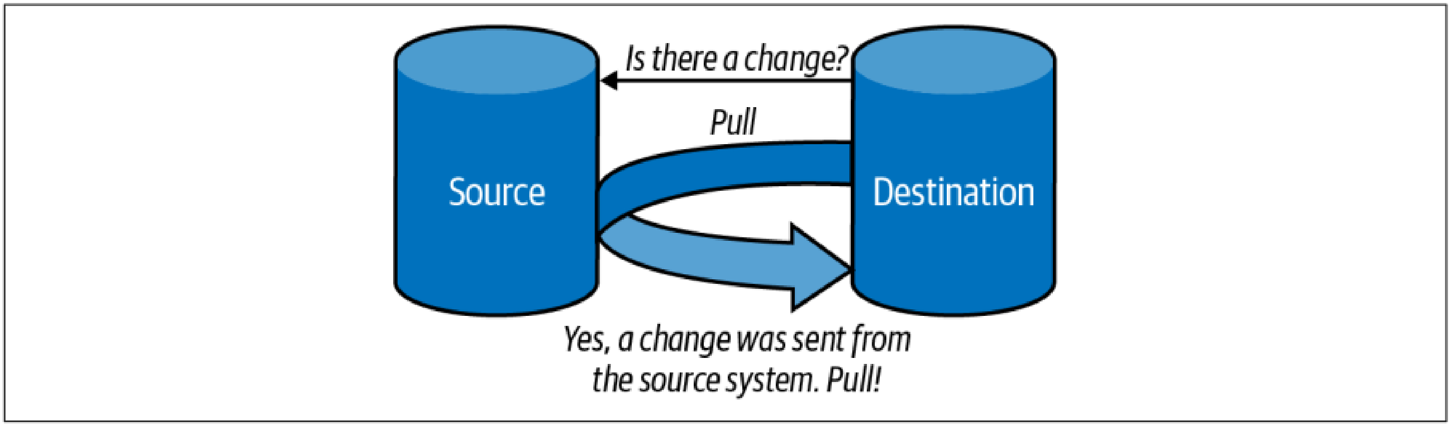


Figure 7-9. Polling for changes in a source system

What is in a message?

“Every message has a schema.”

Example:
Someone posting a tweet is an event.
The tweet is packaged as a message.
The message has the schema {sender_id, text, timestamp}.
The message is sent out by the source application (i.e. the X.com client on the user’s smartphone).
The message is ingested by a consumer application (e.g., Google Pub/Sub).

Message brokers

A message broker is a middleware that translates a message from the data producer’s messaging protocol to the data consumer’s messaging protocol.

E.g., Apache ActiveMQ

Sometimes the messaging system is brokerless. In that case, the consumers ingest data directly from the producers without the involvement of a message broker.

Change Data Capture (CDC)

- CDC is a class of software ‘design patterns’. If we want to design a software that **detects changes/deltas** in the data producer and moves the delta to the data consumer, CDC provides a template of what components should our software have and how these components should interact.

- A **software design pattern** is a design template that presents what components a software should have and how these components should interact.

Webhooks

- Webhooks are a web development method that uses **reverse APIs**.

- An **API** is a software provided by a data provider. Data consumers make requests to the data provider using the API and get the requested data in response. E.g., for our ML projects, we download datasets using APIs of different web platforms.

- A **reverse API** is a software provided by a data consumer. Data producers make requests to the data consumer using the reverse API and the consumer ingests the requested data in response. E.g., GitHub Desktop is a reverse API using which we request GitHub.com to ingest our code delta.

Event logs

- Data producer maintains an ordered log of events. The events are listed in the order they were sent.

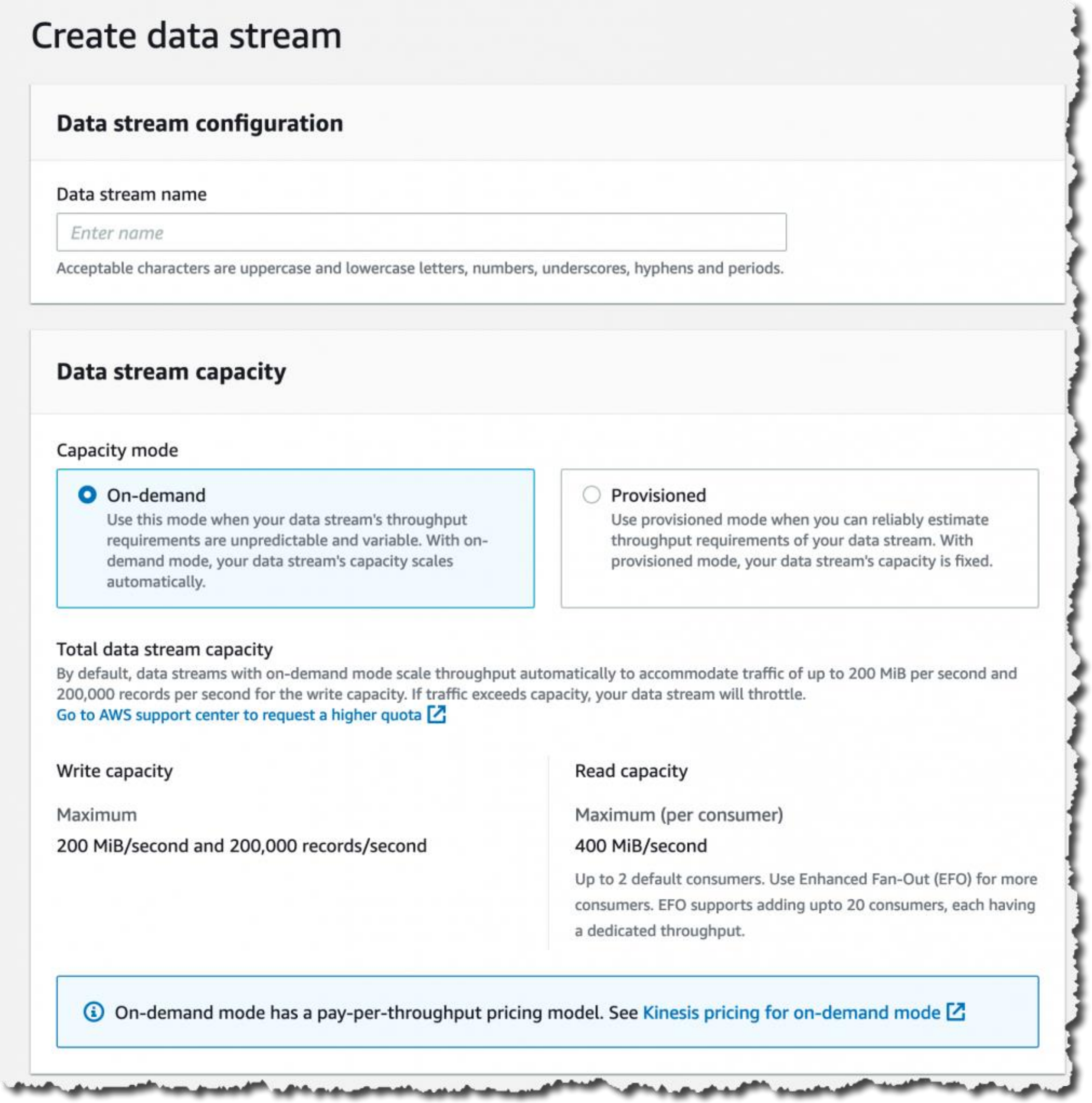
- Data consumer maintains an ordered log of events. The events are listed in the order they were consumed.

- There are mechanisms to synchronize the event logs of the producer and the consumer. Suppose, the producer log says events {e1, e2, e3} were sent but the consumer log says only events {e1, e3} were received. In that case, the producer may resend event ‘e2’.

Topics or streams

- Related events can be grouped into topics or streams
- Multiple producers can send messages to the same topic
- Multiple consumers can receive messages from the same topic

How to create a data stream: Example from Amazon Kinesis Data Streams



Create data stream

Data stream configuration

Data stream name

Enter name

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

Data stream capacity

Capacity mode

● On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

○ Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

Total data stream capacity
By default, data streams with on-demand mode scale throughput automatically to accommodate traffic of up to 200 MiB per second and 200,000 records per second for the write capacity. If traffic exceeds capacity, your data stream will throttle.
Go to AWS support center to request a higher quota

Write capacity

Maximum

200 MiB/second and 200,000 records/second

Read capacity

Maximum (per consumer)

400 MiB/second

Up to 2 default consumers. Use Enhanced Fan-Out (EFO) for more consumers. EFO supports adding upto 20 consumers, each having a dedicated throughput.

ⓘ On-demand mode has a pay-per-throughput pricing model. See Kinesis pricing for on-demand mode

The infrastructure is on the cloud (AWS). Hence, we can **scale to virtually unlimited volume of data stream**.
Moreover, **we pay for what we use**.

That means we do not have to manage the servers that our streaming based app is using. For this reason, such pay-as-you-go cloud services are also known as **serverless** services.

https://www.infoq.com/news/2021/12/kinesis-data-streams-ondemand/

Relationship between a data stream and its event logs

The event logs are maintained as long as the corresponding (data) stream is not terminated.

The event logs enable

- events to be queried over various ranges

- events to be aggregated

- events to be combined with the events of the other streams. This phenomenon is known as ‘stream joins’.

Stream joins

Multiple streams belonging to the same ‘session’ can be joined. E.g., five streams have been joined in the following Chandrayaan-3 live streaming session.

https://www.indiatv.in/india/national/chandrayaan-3-pm-narendra-modi-to-visit-bengaluru-on-august-26-to-congratulate-isro-team-2023-08-25-983763

Stream joins (contd.)

A stream can also be joined with a database relation, such as a table stored inside a RDBMS.

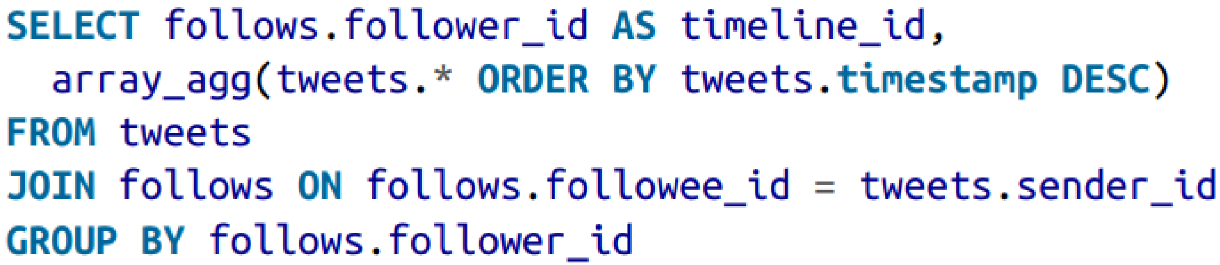
Example:

When a user logs onto his/her X (Twitter) account, a **personalized timeline** is shown. The timeline consists of the tweets by the people he/she follows ordered from the latest tweet to the oldest.

Stream joins (contd.)

It is achieved by joining the ‘tweet’ stream with the ‘follows’ table.

Here, each event in the ‘tweet’ stream is a tweet with the schema {sender_id, text, timestamp}. The ‘follows’ table has the schema {follower_id, followee_id}.



```
SELECT follows.follower_id AS timeline_id,
  array_agg(tweets.* ORDER BY tweets.timestamp DESC)
FROM tweets
JOIN follows ON follows.followee_id = tweets.sender_id
GROUP BY follows.follower_id
```

Query courtesy: Page 475, Kleppmann

What happens if the producers send messages faster than the consumers can process them?

- **Drop** messages, e.g., UDP (second-by-second sensor readings and stock market feeds)

- **Block** the producers from sending new messages. It is known as applying ‘backpressure’ or ‘flow control’, e.g., TCP

- **Buffer** messages in a queue (Mutliple factors (such as the buffer size) and multiple use cases (such as what happens if the buffer is full) to keep in mind. Thoughtful design is necessary.)

Schema registry

In practice, a data producer can change the schema of the messages on the fly.

E.g., the producer can introduce new attributes, remove existing attributes, update the data type of an attribute.

A schema registry is a version control software that maintains the version history of the schema. It helps the data consumer to understand the schema of the incoming messages and extract information accordingly.

E.g., Confluent Schema Registry (https://github.com/confluentinc/schema-registry) is a scheme registry software that helps Apache Kafka (a stream processor) to adapt to the schema changes in the incoming messages.

Schema registry (contd.)

Currently, many top data engineers are involved in developing schema registry software that can automatically

- detect the schema changes and
- help the consumer adjust its data pipeline on the fly

with the help of AI.

References

- Section 8.3 ‘In-Memory Analytics’, P. RAJ, A. RAMAN, D. NAGARAJ, S. DUGGIRALA (2015), High-Performance Big-Data Analytics: Computing Systems and Approaches, Springer, 1st Edition.

References (contd.)

- Chapter 11 ‘Stream Processing’, M. KLEPPMANN (2017), Designing Data-Intensive Applications The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, O’Reilly.
  - Pages 440-443 (Transmitting Event Streams)

References (contd.)

- Chapter 7 ‘Ingestion’, J. Reis, M. Housley (2022), Fundamentals of Data Engineering, O'Reilly Media, Inc.,ISBN: 9781098108304
  - Pages 233-239
  - 242-244
  - 255-256 (Stream Joins)
  - 259-260 (Webhooks)

Thank you