Module VI
Scalability of IoT Solutions

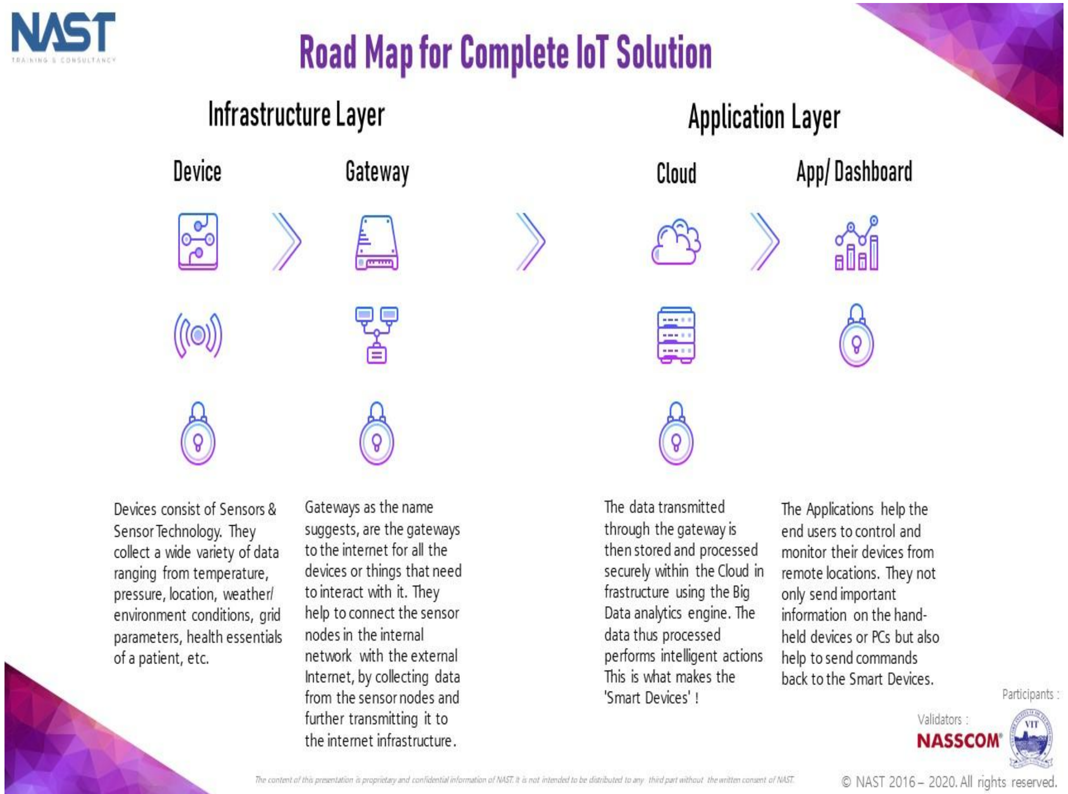
Roadmap for developing complete IoT solutions, Strategies for implementation, key Milestone, Scalability of IoT Solutions, Methods, platforms and tools. Web and Mobile Interfaces

By

Dr Shola Usharani

What is Roadmap in IoT technology

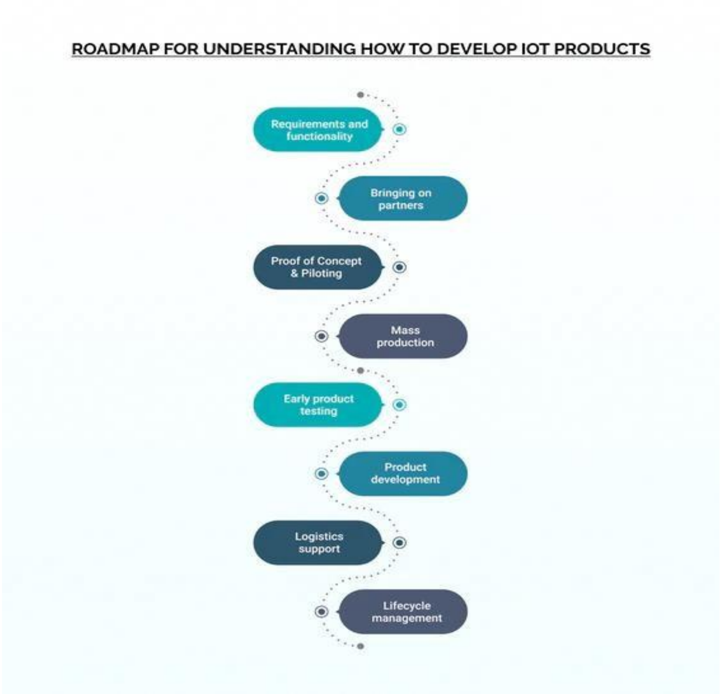
- It is one more general steps of administration and evaluation of tech-stacks in order to strategically define the technology initiative that assists in the process of constituting IoT products / ecosystems including research and development.
- Action plan which has to propose ingredients or inputs and behavior inside system



Need of roadmap in IoT technology

- To define **the time frame in** which the initiatives of technological processes will be implemented
- To answer the question of naming the business goal that will be achieved, i.e. how technology, applications, processes, etc. will help the growth of an organization or system that implements the technology roadmap
- To **have clearly documented developmental processes** that should in practical terms show the steps that need to be taken into account in the development process.
- **Declaring existing and predictions of future technological** solutions that can help achieve the goal of an efficient IoT product / ecosystem.
- To **know about the tools, external and internal,** that are used in some stages of development
- To answer the questions about how security processes in all processes will be implemented in these processes
- To know the questions related to tracking and update UX processes (menus, drag-and-drop, onboarding and branding …)

Steps in Roadmap for IoT Development



Requirements and functionality

- IoT projects are different as there are different project toolkits which are purely meant to solve certain problems with specific conditions.
- Write the following based on scenario
  - what functionality to add in that product and finding the right technology that suits them.
  - find out what your project needs or requirementsin terms of requirement, ecosystem, size, and format and also from a hardware and software point of view.

Bringing on partners

- Several different stakeholders need to engage to fully understand the principle and functionality of the future IoT ecosystem, e.g. Executives, Sales, Marketing, and Engineering must work together on the definition of tech-stack.
- Write the following based on scenario
  - Identify the project areas and putting it on **the project teams.**
  - stretching the internal team and outsourcing the common services like IT and Manufacturing.
  - Tracing those records
    - to check whether the project meets the requirements.
  - Understanding the project and teams
  - Clear communication process.

Proof of Concept & Piloting

- Two important points to consider identifying existing solutions and assessing risk.
  - Knowing the account product requirements and the risk assessments, the design and product development plan will map out project roles, timelines, and tasks and all such things.
  - Need to write about exiting solutions & assessment of risks for the scenario.
- In this stage, you will also be engaging key stakeholders, CFO, CEO and CMO for getting their buy-in and input. This should cover the conceptualization of the product for giving your whole team a clear picture of how the end product will look like.

Mass production

- To monitor and track the whole production.
- This stage usually follows the testing, validation, and refinement of the product during earlier phases (like piloting or beta testing) and involves full-scale deployment to meet demand or fulfill project objectives.
- Key aspects of mass production
  - Large-Scale Manufacturing
  - Standardization
  - Supply Chain and Logistics
  - Cost Efficiency

Logistics support

- the transfer of production responsibility to manufacturing teams.
- Also, this covers quality control processes, documentation control and production planning.
- Key aspects
  - Supply Chain Management
  - Transportation & Distribution
  - Inventory Management
  - Resource Allocation
  - Maintenance and Support Services
  - Risk Management in delays & trasnportation

Product development

- The detail engineering of the product by creating the 3D design of your product, refining primary product features, validating the product features and at the last developing the prototype

Lifecycle management

- Analyze and monitor the product data generated through your IoT platform for enhancing the next iteration of your product.
  - Understanding metrics for the business outcomes.
- Handling of product updates is performed for avoiding any disruption in the usability

NAST

How to define a successful IoT implementation strategy

Part 1 out of 3

Clearly define your business goals

A clearly defined problem statement is the first thing you should have when you think of IoT implementation.

Identify the hardware and devices to be connected

You should diligently examine your existing inventory and identify the hardware and machinery th at you want to connect. You might have to purchase third-party adapters, sensors, actuators and other hardware depending on the IoT-readiness of your assets.

Analyze data points and store information

Attaching sensors with the hardware and machinery will result in multiple data points that will ultimately generate large amounts of data. You need to have the right strategy in place to store and analyze this data and ensure that it aligns with your expected outcomes.

NAST

How to define a successful IoT implementation strategy

Part 2 out of 3

Define connectivity options and protocols

Different types of connections to deal with:
Device to device (D2D), Device to gateway, Gateway to data systems, Between data systems

These types of connections are achieved with different types of IoT networks, such as:
PAN, LAN, MAN, WAN, NFC, Mesh networks, Ring networks, Tree networks

Ensure compliance, governance and security

Security is essential for IoT projects since the whole game revolves around the data generated. Lack of awareness can be a major obstacle to security and you're very likely to put your data at risk if comprehensive security measures are not taken.

Participants :
Validators : NASSCOM

The content of this presentation is proprietary and confidential information of NAST. It is not intended to be distributed to any third part without the written consent of NAST.

© NAST 2016 – 2020. All rights reserved.

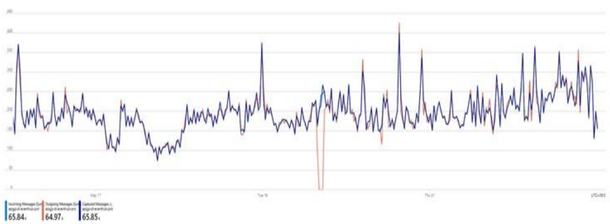
NAST

How to define a successful IoT implementation strategy

Part 3 out of 3

Make sense out of data

Gathering and storing is just a job half done. You need to analyze it in such a way that your whole purpose of implementing an IoT solution is served and you can make informed decisions. You need to define what KPIs you want to monitor and also decide machine learning (ML) algorithms to solve your business problems. Data will keep growing once you connect your assets, thus ML algorithms and analytics play a significant role in the success of your IoT project.



Real time Electricity Trade Data's from UK over a period of few days.

Participants :
Validators : NASSCOM

The content of this presentation is proprietary and confidential information of NAST. It is not intended to be distributed to any third part without the written consent of NAST.

© NAST 2016 – 2020. All rights reserved.

References

- https://www.novatec-gmbh.de/blog/iot-technology-roadmap/

IoT Scalability

- Number of devices connecting to the internet increasing exponentially.
- The data connected transferring through the web interface is increasing.
  - Media platforms, communication, playing game, online banking etc.,
- Ability of a device to adapt the changes of the environment that need to occur in future.
- Capability to handle the growing amount of work.
- Without this
  - Poor system performance
  - Reengineering of the whole system
- Device scalability
  - Horizontal scalability
  - Vertical scalability

Horizontal and vertical scalability

- Horizontal :
  - Expanding the network through number of hardware devices and software used in the network.
    - Adding more machines, nodes into the system.
    - Adding a distributed software application.
    - SOA system, web servers and more servers for load-balanced network.
- Vertical:
  - To increase the efficiency of existing hardware and software through adding more number of resources.
    - Increasing the processing power of as server to increase the speed.
    - Adding main memory, storage and network interfaces and processors.
    - All facilities the use of virtualization technology more effectively.

Features considered for scalability

- Business
  - Increasing large amount of data into the database and make it as scalable.
- Marketing
  - Suitable for all environments.
  - Should be easily understandable and meets the requirements of customer.
- Software
  - Should move from smaller to larger system with the full advantage of including performance metrics
- Hardware
  - Devices need to connect to their respective backend systems in a secure fashion as the information they relay is often private or sensitive, it is necessary to encrypt the data before it leaves the device
- Networks
  - A scalable network must have the ability to adapt when failures occur and it must also remain mostly operational until the issue is repaired

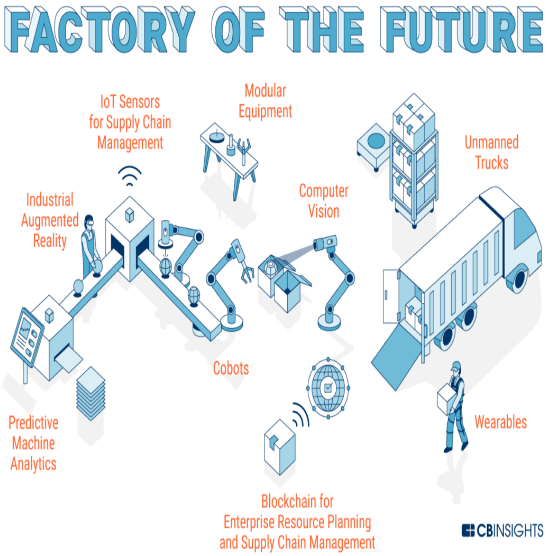
Techniques for Scalability

- ***Using automated bootstrapping***
- ***Controlling the IoT data pipeline***
- ***Applying the three-axis approach for scaling***
- ***Developing micro-services architecture***
- ***Adopting multiple data storage technologies***
- ***Create a system that can be easily expanded***

Using automated bootstrapping

- Mechanism that involves human interaction and facilitation starts becoming obsolete and impractical .
  - manual functions such as bootstrapping, software and security configuration, device registration and upgrade are no longer feasible.
- all these services must automate the aforementioned processes in order to save time and act more efficiently.

Replacing 500 manual employees with 5 automated robots → Industry 4.0



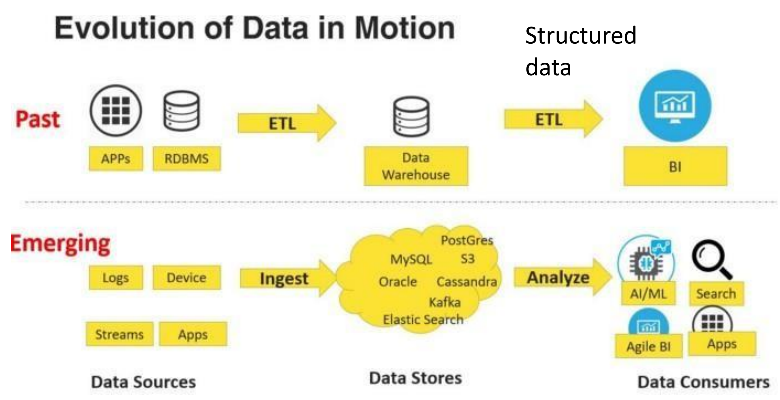
IoT Sensors for Supply Chain Management:
•Internet of Things (IoT) sensors are deployed across the factory to monitor inventory, track equipment, and ensure smooth supply chain operations in real-time.
Industrial Augmented Reality (AR):
•Augmented Reality devices assist workers and technicians by overlaying digital information on the physical environment, enhancing training, maintenance, and troubleshooting processes.
Predictive Machine Analytics:
•Predictive analytics algorithms analyze machine data to foresee potential issues, enabling preemptive maintenance and reducing downtime.
Cobots (Collaborative Robots):
•Cobots are robots designed to work alongside human workers safely, helping with repetitive tasks, precision assembly, and other manual processes.

Blockchain for Enterprise Resource Planning (ERP) and Supply Chain Management:
•Blockchain technology provides a secure and transparent way to track resources, manage transactions, and verify product provenance, enhancing traceability and security in the supply chain.
Modular Equipment:
•Modular machines can be reconfigured or replaced easily, allowing the factory to adapt quickly to changes in production demand or new product lines.
Computer Vision:
•Computer vision systems enable automated quality control and monitoring by detecting defects or anomalies in products and processes in real-time.
Unmanned Trucks:
•Autonomous vehicles transport materials and products around the factory floor or warehouse without human intervention, optimizing logistics.
Wearables:
•Wearable devices, such as smart glasses or wristbands, provide workers with real-time data and safety alerts, enhancing efficiency and safety on the floor.

Controlling the IoT data pipeline

- Data pipeline is used for data stream that is being transferred between various systems and “things”.
- control in the data pipeline as required by a particular service.
- IoT devices generate and transmit tremendous amount of data that needs to be processed and organized in a required format for it to be usable .
- **these data pipelines must be designed in such a way that they can handle sudden surge in the rate (seasonal based requests) of flow of data and performance issues**
- The capacities of these data streams must be adjusted based on valuable parameters such as the number of simultaneously connected devices or data streams.

Comparison of Old and new data processing models



Structured data

Un- Structured data like streams, images etc.

This image illustrates the evolution of data processing approaches from traditional (labeled as "Past") to modern (labeled as "Emerging") methods. Here’s a breakdown of each section:

Past

1.Data Sources: Previously, data mainly originated from applications (APPs) and relational databases (RDBMS).

2.ETL Process: Data was extracted, transformed, and loaded (ETL) into a Data Warehouse.

3.Data Consumption: The data in the warehouse was then further processed and transformed through another ETL phase before being used by Business Intelligence (BI) tools, which provided insights and reporting.

Emerging

1.Data Sources: Today, data sources are more diverse and include logs, devices, streams, and applications. This is due to the rise of IoT, web applications, and real-time monitoring.

2.Ingestion: Data is ingested into a broader variety of data storage solutions. Some common data stores are listed:

1. **Postgres, MySQL, Oracle**: Relational databases that support structured data.
2. **S3**: Cloud storage service by Amazon.
3. **Cassandra, Kafka, ElasticSearch**: NoSQL and streaming technologies that handle large volumes of unstructured or semi-structured data, often in real-time.

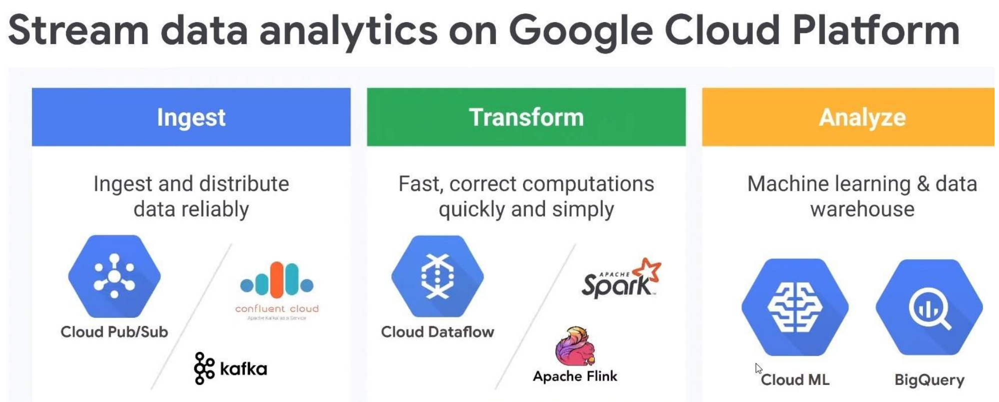
3.Analysis: Modern analytics and processing methods allow for faster and more complex analysis:

1. **AI/ML**: Artificial Intelligence and Machine Learning for predictive analytics and pattern recognition.
2. **Search**: Search capabilities, often powered by systems like Elasticsearch, allow users to find specific data quickly.
3. **Agile BI and Apps**: Agile Business Intelligence and various applications utilize real-time data for dynamic reporting, dashboards, and user applications.

Summary

The shift from traditional ETL processes and static data warehousing to a more agile, diverse, and real-time data infrastructure has enabled faster insights and more sophisticated analytics, empowering businesses to make better, data-driven decisions in real-time.

IoT data streaming



This figure illustrates a stream data analytics workflow on the **Google Cloud Platform (GCP)**. It shows how data is processed in real-time through three main stages: **Ingest**, **Transform**, and **Analyze**. Each stage leverages specific Google Cloud and open-source tools to enable fast and reliable data streaming, transformation, and analysis. Here's an explanation of each stage:

1. Ingest

•**Purpose**: The Ingest phase is responsible for collecting and reliably distributing real-time data from various sources into the analytics pipeline.

•**Tools**:

- **Cloud Pub/Sub**: A messaging service that allows data to be ingested in real-time and distributed to multiple systems, enabling seamless data integration and scalability.
- **Confluent Cloud** (with Apache Kafka): This managed service version of Apache Kafka provides scalable event streaming, making it easier to ingest large volumes of real-time data into the pipeline.

2. Transform

•**Purpose**: The Transform phase processes the ingested data to apply computations, aggregations, and transformations, ensuring the data is prepared and ready for analysis.

•**Tools**:

- **Cloud Dataflow**: A fully managed streaming analytics service that enables fast, correct computations on large datasets. It is often used for ETL (Extract, Transform, Load) processes and is based on the Apache Beam programming model.
- **Apache Spark** and **Apache Flink**: These are popular open-source frameworks for processing large data sets in a distributed manner. They support batch and stream processing, which is essential for transforming data in real time.

3. Analyze

•**Purpose**: The Analyze phase involves deriving insights from the transformed data using machine learning models and data analysis tools.

•**Tools**:

- **Cloud ML (Machine Learning)**: Provides tools for building, training, and deploying machine learning models, enabling advanced analytics and predictive modeling.
- **BigQuery**: Google Cloud's fully managed, serverless data warehouse that allows fast SQL queries on massive datasets, making it ideal for data analytics.

Summary

This workflow on Google Cloud Platform enables a seamless, scalable, and efficient pipeline for real-time data analytics:

•**Ingest**: Collects and distributes data quickly and reliably.

•**Transform**: Processes data in real time, preparing it for analysis.

•**Analyze**: Uses machine learning and data warehousing tools to generate insights and predictions.

By leveraging these tools, organizations can achieve fast, reliable, and actionable analytics from real-time data, allowing them to make data-driven decisions in a timely manner.

Applying the three-axis approach for scaling

- scaling in various directions
  - X axis: is associated with **utilizing more resources** to distribute the incoming demands amongst various servers, in such a way that all servers are capable of handling requests
  - Y axis: means **distributing the tasks** at hand based on processes
  - Z-axis means **allocating the responsibilities** on the basis of the **incoming request or response data**.

X-Axis: Scale by Cloning Services

•**Description**: This approach involves creating multiple copies of the same service (also known as horizontal scaling). Each instance of the service operates independently and can handle requests, allowing the workload to be distributed across multiple instances.

Y-Axis: Scale by Using Microservices (Functional Decomposition)

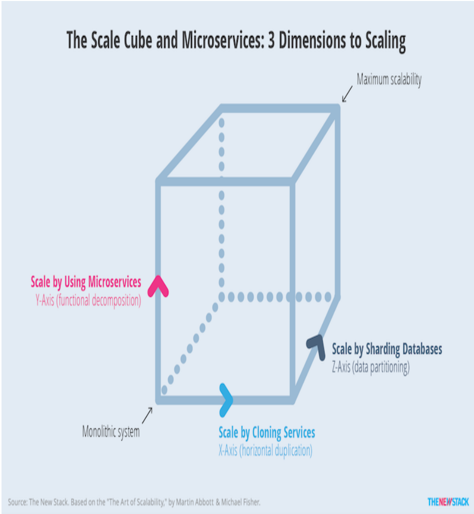
•**Description**: Instead of cloning the entire application, the Y-axis splits it into smaller, independent services based on functionality. This is the essence of the microservices approach, where each service is responsible for a specific function or business capability

Z-Axis: Scale by Sharding Databases (Data Partitioning)

•**Description**: Z-axis scaling involves partitioning or "sharding" data across multiple databases or storage locations. Each shard contains a subset of the data, based on specific criteria, like user ID ranges or geographical regions.

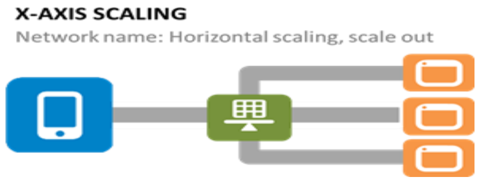
Applying the three-axis approach for scaling

- Organizational construction was used as an analogy for systems to help better reinforce the approach of each of the three axes of scale. The cube is constructed such that the initial point (x = 0, y = 0, z = 0) is a monolithic system or organization (single person) performing all tasks with no bias based on the task, customer, or requestor.

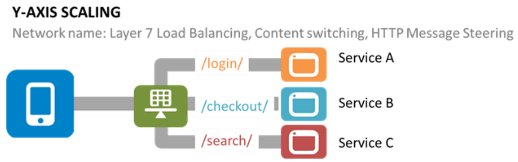


Explanation: The X-axis represents horizontal scaling, where the entire application or service is cloned across multiple instances. This is a straightforward approach where identical copies of the application or service are created.
Illustration: The figure shows a network that directs traffic to multiple identical service instances, all handling the same tasks.
Explanation: The Y-axis focuses on functional decomposition, breaking down the application into multiple services, each responsible for a specific function. This approach aligns with microservices architecture, where each service handles distinct tasks.**Illustration**: The figure shows different services, such as /login, /checkout, and /search, each performing a unique function. Traffic is routed to the appropriate service based on the request type.
Explanation: The Z-axis involves sharding or partitioning data based on specific criteria. This approach divides data into subsets, each managed by different instances or databases, and routes requests based on data criteria.**Illustration**: The figure shows partitions based on product names (e.g., Products A-G, Products H-O, Products P-Z). Traffic is directed to the partition that holds the relevant data.

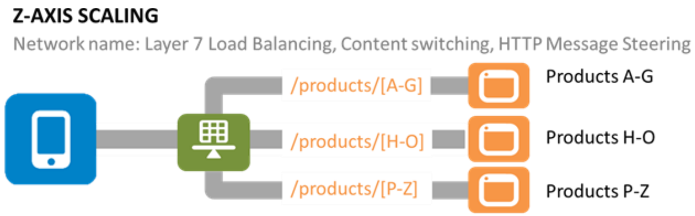
X-Axis Cloning Service



Y-Axis functional decomposition

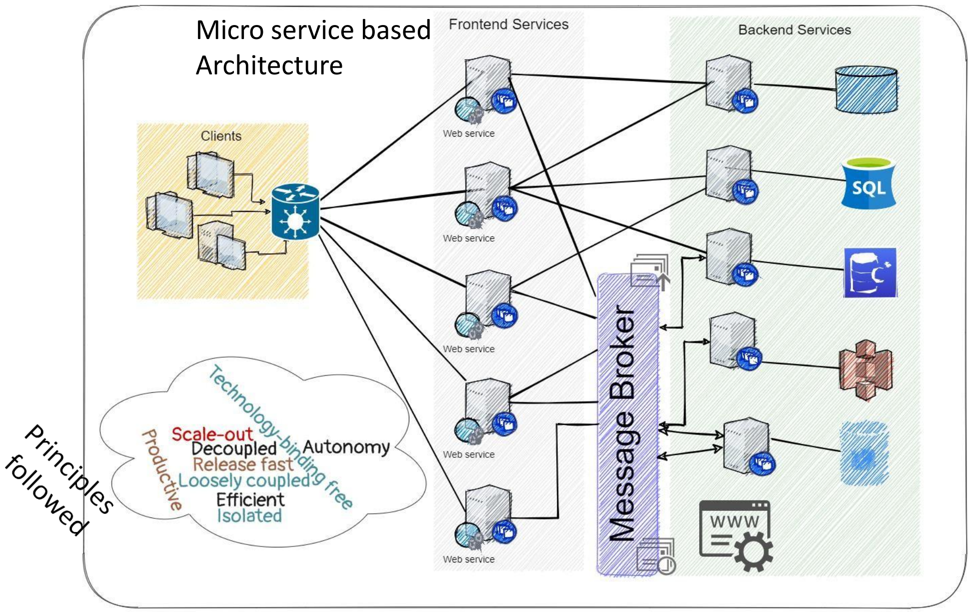


Z-Axis sharding and partition based on requests



Developing micro services architecture

- applications are comprised of individualistic **micro-processes** disseminating
- It is useful to divide each application into **several independent instances** which are often called functional units each of which performs a **separate function.**
- Each of these functional units should work **independently and executed**



Microservices-based architecture is a design approach that divides an application into small, independent services, each with its own responsibilities and functionality. This architecture is particularly suited to scaling because it allows each service to be independently scaled, deployed, and managed. Here's how microservices help with scaling and the key elements of implementing a scalable microservices-based architecture:

1. Independent Services (Microservices)

•Each service in a microservices architecture performs a specific business function (e.g., user authentication, payment processing, product catalog).

•Services are loosely coupled and communicate with each other through APIs, often using lightweight protocols like REST or gRPC.

•By isolating services, teams can scale each service independently based on load or demand, rather than scaling the entire application.

Adopting multiple data storage technologies

- Different techniques for their storage rather than using one common technique for all.
- These applications are required to be built on the best suited technology components that are available and hence each of the **microservices** that we are using should use that component that is suited for its needs.
- Build a common API to support all those services

References

- Anisha Gupta1., Rivana Christie2, Prof. R. Manjula3, « Scalability in Internet of Things: Features, Techniques and Research Challenges », International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 7 (2017), pp. 1617-1627.
- https://www.intuz.com/iot-product-development-guide#h_70012386567341592392849374.
- (93) Understanding the basics of any IoT data pipeline — from ingress capture to processing to generation of value (Part 1) | by Ravi Kumar. | Products, Platforms, Business & Innovation in Industry 4.0/IIoT | Medium
- https://medium.com/@helmi.confo/orthogonality-in-scalability-2e88aa92f03a
- https://www.cbinsights.com/research/future-factory-manufacturing-tech-trends/

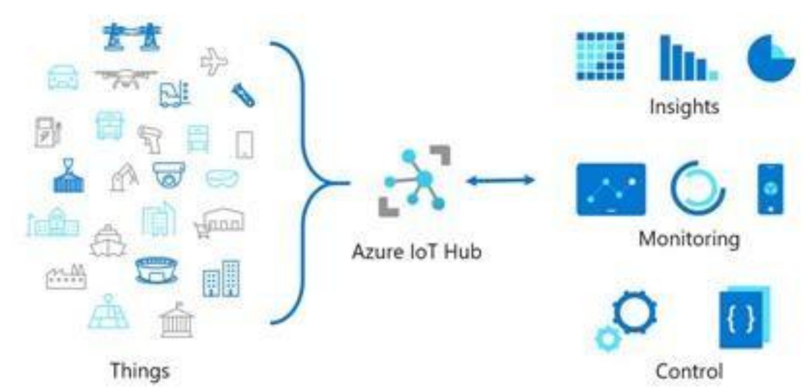
IoT Scalability in Azure (Methods)
&
Platforms , Tools, Web and Mobile Interfaces

Some Azure IoT Services and Methods

Microsoft Azure

- Azure-IoT Hub
- Azure-Event Hub
- Azure Stream Analytics
- Azure- data lake
- Azure-Synapse (Data Ware House)
- Azure-Data Factory

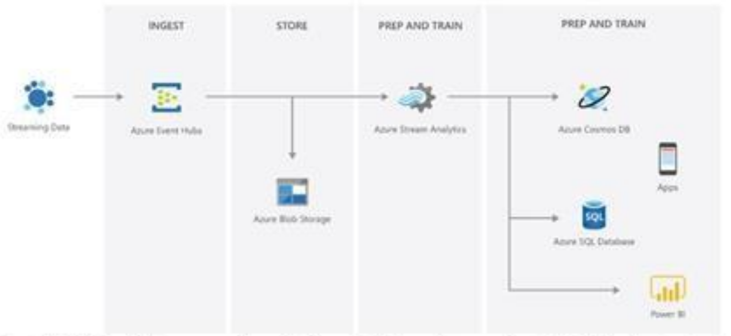
Azure-IoT Hub



IoT Hub is a managed service, hosted in the cloud, that acts as a central message hub for bi-directional communication between your IoT solutions and the devices it manages.

- **This diagram illustrates the structure and function of Azure IoT Hub in connecting "Things" (IoT devices) to provide insights, monitoring, and control. Here's a breakdown of the diagram:**
- **Things (Left Side): This section represents various IoT devices or "Things," such as drones, vehicles, sensors, industrial equipment, and other connected devices. These are physical objects that collect data or perform actions in the real world.**
- **Azure IoT Hub (Center): This hub acts as the central cloud-based service that connects and manages IoT devices. It facilitates secure, bidirectional communication between IoT devices and the cloud.Insights, Monitoring, and Control**
- **(Right Side):Insights: The data from the devices can be analyzed to provide insights, such as visualizations or analytics, to help understand patterns and trends.**
- **Monitoring: Continuous monitoring of device health and performance allows users to track and manage device statuses, detect issues, and maintain operational reliability.**
- **Control: Users can send commands back to the IoT devices, allowing remote management and control, such as adjusting settings or triggering specific actions.**
- **In essence, Azure IoT Hub enables IoT devices to communicate with cloud services, where data is processed and managed, allowing users to gain insights, monitor performance, and control devices remotely.**

Azure-Event Hub



Event Hubs is a fully managed, real-time data ingestion service that's simple, trusted, and scalable. Stream millions of events per second from any source to build dynamic data pipelines and immediately respond to business challenges.

Note - " Additionally, IoT Hub uses Event Hubs for its telemetry flow path, so your IoT solution also benefits from the tremendous power of Event Hubs."

This diagram explains the workflow of Azure Event Hub, a service for ingesting and processing large volumes of streaming data in real-time. Here's a step-by-step explanation of each part of the workflow:

1.Streaming Data (Left): This represents incoming data from various sources, such as applications, devices, or other data-generating systems.

2.Ingest (Azure Event Hub):

1. The data first enters **Azure Event Hub**, a big data streaming platform and event ingestion service. Event Hub allows the capture of millions of events per second, enabling real-time analytics and insights.

3.Store (Azure Blob Storage):

1. The data ingested by Event Hub can be stored in **Azure Blob Storage** for long-term storage or batch processing. Blob Storage provides scalable storage for large amounts of unstructured data, such as logs and raw events.

4.Prep and Train (Azure Stream Analytics):

1. Data can then be sent to **Azure Stream Analytics**, which processes the data in real-time to generate insights. This step typically involves preparing and transforming the data to make it suitable for further analysis or machine learning tasks.

5.Prep and Train (Azure Cosmos DB, Azure SQL Database, Power BI):

1. Processed data can be stored in **Azure Cosmos DB** for NoSQL, globally distributed storage or in **Azure SQL Database** for structured, relational data storage.
2. **Power BI** can be used to visualize the data for reporting and analysis.
3. Additionally, **Apps** can use the processed data to drive actions, enhance user experiences, or enable interactive dashboards.

In summary, Azure Event Hub is used to ingest streaming data, which can be stored, processed, and analyzed through various Azure services, allowing for real-time insights and decision-making.

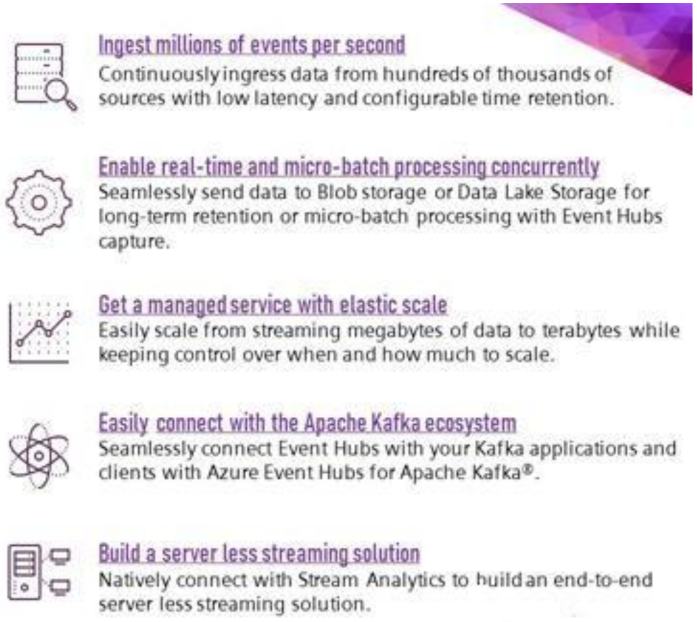
What it is?

- Event Hubs provides a distributed stream processing platform with low latency and seamless integration, with data and analytics services inside and outside Azure to build your complete big data pipeline.

Various Scenarios can be processed by Event Hubs

- Anomaly detection (fraud/outliers)
- Application logging
- Analytics pipelines, such as clickstreams
- Live dashboarding
- Archiving data
- Transaction processing
- User telemetry processing
- Device telemetry streaming

Contd.,



Difference between IoT Hub and Event Hub

- Azure IoT Hub and Azure Event Hubs are cloud services that can ingest large amounts of data and process or store that data for business insights.
- IoT Hub was developed to address the unique requirements of connecting IoT devices to the Azure cloud while Event Hubs was designed for big data streaming. Microsoft recommends using Azure IoT Hub to connect IoT devices to Azure.
- Azure Event Hubs ,It is designed for high throughput data streaming scenarios where customers may send billions of requests per day.
- Event Hubs uses integrated big data and analytics services of Azure including **Databricks, Stream Analytics, ADLS, and HDInsight**. With features like Event Hubs Capture and Auto-Inflate, this service is designed to support your big data apps and solutions.

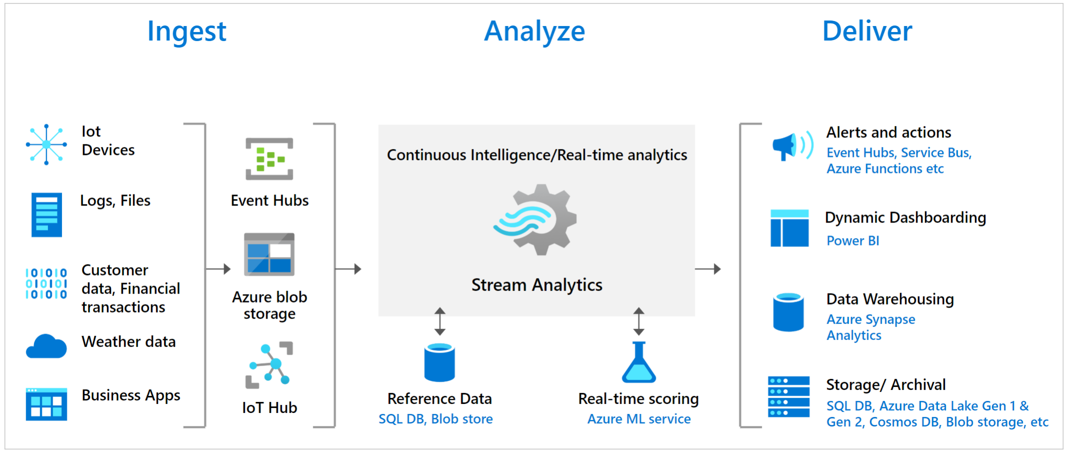
Azure Stream Analytics

- A real-time analytics and complex event-processing engine that is designed to analyze and process **high volumes of fast streaming data from multiple sources simultaneously**.
- Patterns and relationships can be identified in information extracted from a number of input sources including devices, sensors, clickstreams, social media feeds, and applications.
- These patterns can be used to trigger actions and initiate workflows such as creating alerts, feeding information to a reporting tool, or storing transformed data for later use
- Stream Analytics is available on Azure IoT Edge runtime, enabling to process data on IoT devices.

Examples on Azure Stream Analytics

- Analyze real-time telemetry streams from IoT devices.
- Web logs/clickstream analytics.
- Geospatial analytics for fleet management and driverless vehicles.
- Remote monitoring and predictive maintenance of high value assets.
- Real-time analytics on Point of Sale data for inventory control and anomaly detection.

Azure-stream Analytics



Ingest

The *Ingest* stage is responsible for collecting and storing data from various sources.

Data Sources:

•IoT Devices: Data from connected devices, such as sensors in an Internet of Things (IoT) environment.

•Logs, Files: System logs or files generated by applications.

•Customer Data, Financial Transactions: Data related to customers or transactional data, often from business processes.

•Weather Data: Data from weather monitoring systems.

•Business Apps: Data generated by business applications or software.

Ingestion Services:

Event Hubs: Azure Event Hubs captures and streams large amounts of event data from sources like applications, sensors, or devices.Azure Blob Storage: A scalable storage solution for large volumes of unstructured data, like text and binary data.IoT Hub: Connects, monitors, and manages IoT devices, serving as a central hub for IoT data.

Analyze

real-time analytics is performed on the ingested data. This is where Azure Stream Analytics processes and extracts insights from streaming data.

Stream Analytics: This service performs real-time analytics and processing on streaming data, providing continuous intelligence as data flows in.Additional Components:Reference Data: Data that provides context to the streaming data, which can be stored in SQL databases or Blob Storage.Real-time Scoring: Integration with Azure Machine Learning (Azure ML) allows real-time scoring, where ML models can make predictions based on incoming data in real-time.

The *Deliver* stage is where the results of the analysis are shared with other systems, visualized, or stored for future use.

•Alerts and Actions: Based on analysis, alerts or automated actions can be triggered using services like Event Hubs, Service Bus, or Azure Functions to notify users or initiate workflows.

•Dynamic Dashboarding: Power BI is used to create dynamic dashboards that visualize data in real-time, providing decision-makers with actionable insights.

•Data Warehousing: Azure Synapse Analytics is used for large-scale data warehousing, enabling deeper analysis and historical storage of processed data.

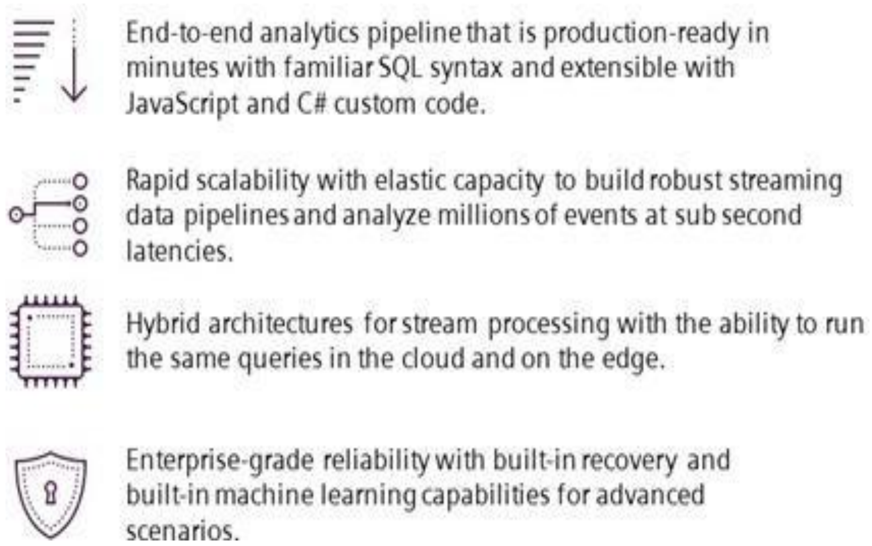
•Storage/Archival: Processed data can be archived for long-term storage in various storage solutions, such as:

- **SQL Databases**: For relational data.
- **Azure Data Lake Gen 1 & Gen 2**: Big data storage solutions.
- **Cosmos DB**: A globally distributed, multi-model database.
- **Blob Storage**: For storing large volumes of unstructured data.

Working principle

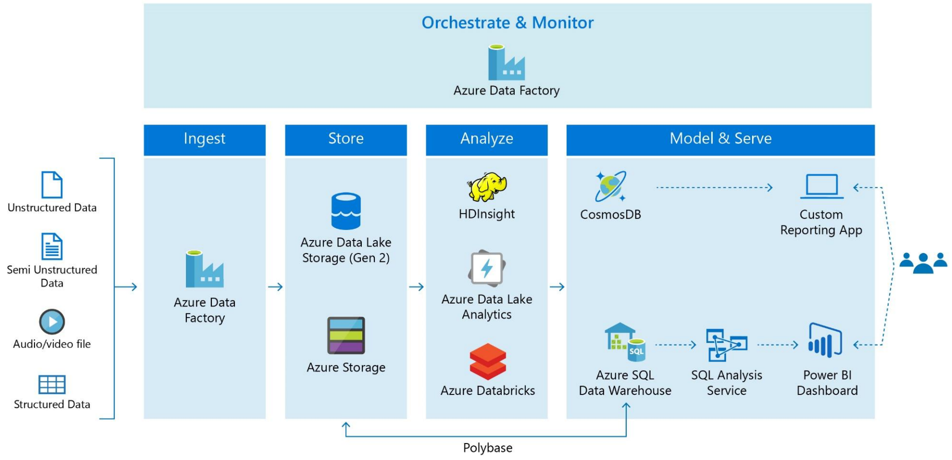
- Azure Stream Analytics job consists of an **input, query, and an output**.
- Stream Analytics ingests data from Azure Event Hubs (including Azure Event Hubs from Apache Kafka), Azure IoT Hub, or Azure Blob Storage.
- **Send data to services such as Azure Functions, Service Bus Topics or Queues to trigger communications or custom workflows downstream**.
- Send data to a Power BI dashboard for real-time dashboarding.
- Store data in other Azure storage services (e.g. Azure Data Lake, Azure Synapse Analytics, etc.) to train a machine learning model based on historical data or perform batch analytics.

Characteristics



Azure Data Lake offers all the capabilities and services to help developers, Data Scientists, and Analysts to store data of any size, shape, and speed. It helps to perform all kinds of processing and analytics across platforms using various languages. It makes it very easy and speeds the process of storing and ingesting data using batch, streaming, and interacting analytics.

Azure-Data Lake



Azure Data Factory: This is a data integration service that orchestrates the flow of data through each stage of the pipeline.

Ingest

The *Ingest* stage is responsible for bringing data into the pipeline. This stage handles various data types, including:

•Unstructured Data: Such as text documents or images.

•Semi-Unstructured Data: Like JSON or XML files that have some structure but are not strictly organized.

•Audio/Video Files: Multimedia content.

•Structured Data: Data that is highly organized, such as relational tables.

In this stage:

•Azure Data Factory ingests and prepares this data, facilitating the movement of data from its source to storage.

Store

Azure Data Lake Storage (Gen 2): A scalable storage service for big data analytics, suitable for unstructured and semi-structured data.

Azure Storage: General-purpose storage that can store a variety of data types, including blobs, files, queues, and tables.

Analyse

HDInsight: An Azure service that allows for **big data processing using popular frameworks like Hadoop, Spark, Hive, and more**.

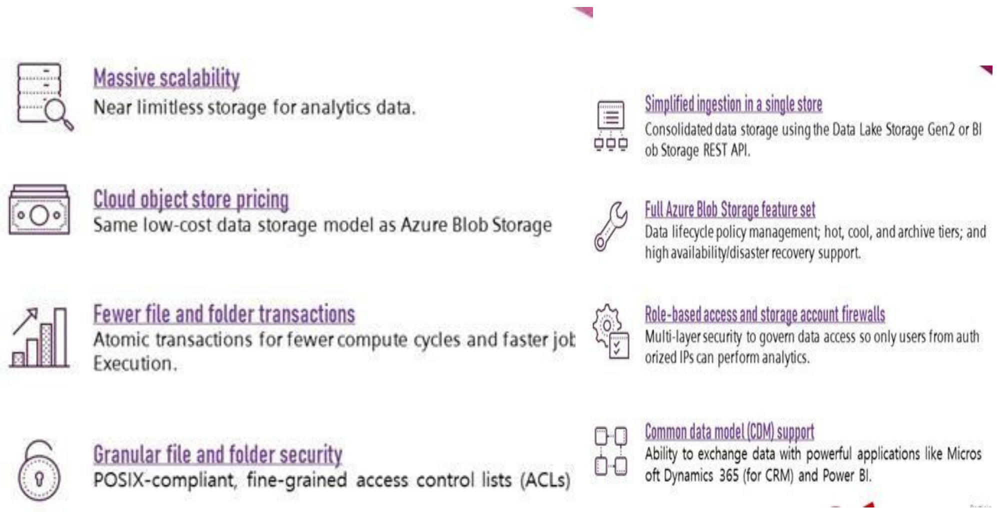
Azure Data Lake Analytics: An **on-demand analytics job service** that allows for large-scale data transformation and analysis.

Azure Databricks: A collaborative **data processing platform that supports machine learning, data engineering, and big data analytics**. It’s optimized for Apache Spark and provides tools for data scientists and engineers.

Model & Serve

This stage prepares the analyzed data for end-user consumption, delivering insights through reports, dashboards, and applications.
•CosmosDB: A globally distributed, multi-model database that serves data to applications in real-time.
•Azure SQL Data Warehouse: A fully managed, scalable data warehousing solution, used for storing and querying structured data.
•SQL Analysis Service: Provides analytical data processing, enabling complex data models that support interactive and real-time querying.
•Power BI Dashboard: Power BI provides visualization and reporting tools that allow end-users to view data insights and generate interactive reports.
The outputs from these services can be delivered to:
•Custom Reporting Apps: Applications that provide tailored reporting and analytics.
•Power BI Dashboards: Visualization dashboards for business intelligence.
•Polybase is a feature that enables the SQL Data Warehouse to query and import data directly from external storage, like Azure Data Lake, making it easier to integrate and query across different data sources.

Contd.,



Azure-Synapse(Data ware house)

- **Azure Synapse** is an integrated analytics service that accelerates time to insight across data warehouses and big data systems.
- Azure Synapse brings together the best of **SQL** technologies used in enterprise data warehousing,
  - **Spark (**environment with the global scale and availability of Azure platform**)** technologies used for big data, and **Pipelines** for data integration and ETL/ELT.
  - **Synapse Studio** provides a unified experience for **management, monitoring, coding, and security. Synapse has deep integration** with other Azure services such as **PowerBI**, **CosmosDB**, and **AzureML**.

Features

- Industry-leading SQL
- Industry-standard Apache Spark
- Interop of SQL and Apache Spark on your Data Lake
- Built in data integration via pipelines
- Unified management, monitoring and security

Azure Data Factory

- For example, imagine a **gaming company that collects petabytes of game logs** that are produced by games in the cloud. The company wants to **analyze these logs to gain insights into customer preferences, demographics, and usage behavior**. It also wants to identify **up-sell and cross-sell opportunities, develop compelling new features,** drive business growth, and provide a better experience to its customers.
  - To analyze these logs, the company needs to use reference data such as **customer information, game information, and marketing campaign information that is in an on-premises data store**. The company wants to utilize this data from the on-premises data store, combining it with additional log data that it has in a cloud data store.
  - To extract insights, it hopes to process the joined **data by using a Spark cluster in the cloud (Azure HDInsight), and publish the transformed data into a cloud data warehouse such as Azure Synapse Analytics (formerly SQL Data Warehouse) to easily build a report on top of it**. They want to automate this workflow, and monitor and manage it on a daily schedule. They also want to execute it when files land in a blob store container.
- Azure Data Factory is the platform that solves such data scenarios. **It is the *cloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale***. Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that can ingest data from disparate data stores. You can build complex ETL processes that transform data visually with data flows or by using compute services such as Azure HDInsight Hadoop, Azure Databricks, and Azure SQL Database.
- Azure Data Factory is a managed cloud service that's built for these complex hybrid **extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects.**

References

- https://azure.microsoft.com/en-in/services/iot-hub/#security.
- https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction
- https://www.xplenty.com/blog/data-ingestion-vs-etl/

https://medium.com/@pallav.aggarwal/web-based-user-interface-for-your-iot-product-f1ef99630989

WEB AND MOBILE INTERFACES