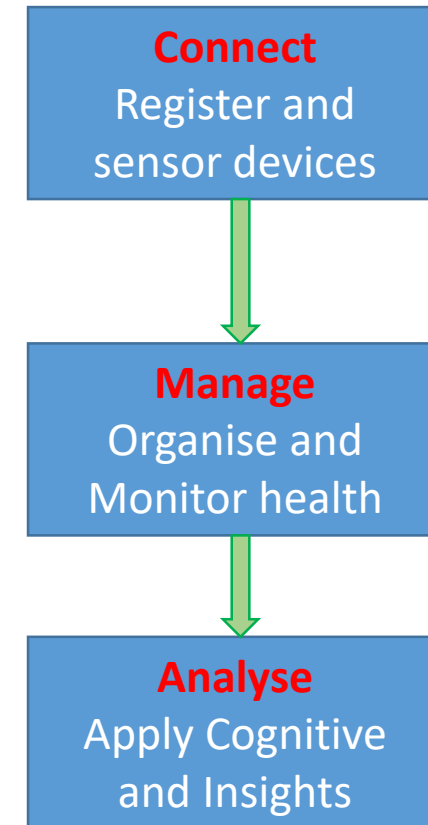


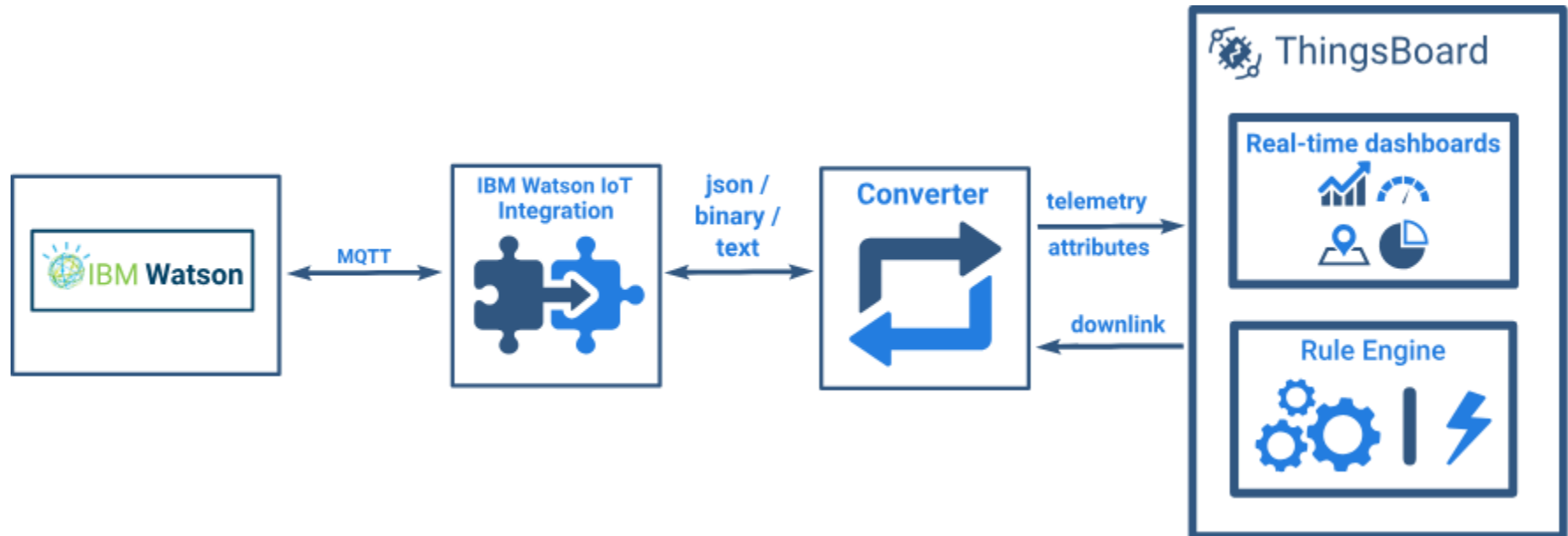
What is IBM Watson IOT?

- It is a super computer, also Based on cognitive computing-learn & then solve problems.
- Fully managed, cloud-hosted platform
- Connects devices, sensors and gateways
- Ingests and analyses massive data streams
- Tools for device management and security.



IOT platform Architecture

The Internet of Things (IoT) refers to a network of physical devices, vehicles, appliances, and other physical objects that are embedded with sensors, software, and network connectivity, allowing them to collect and share data.



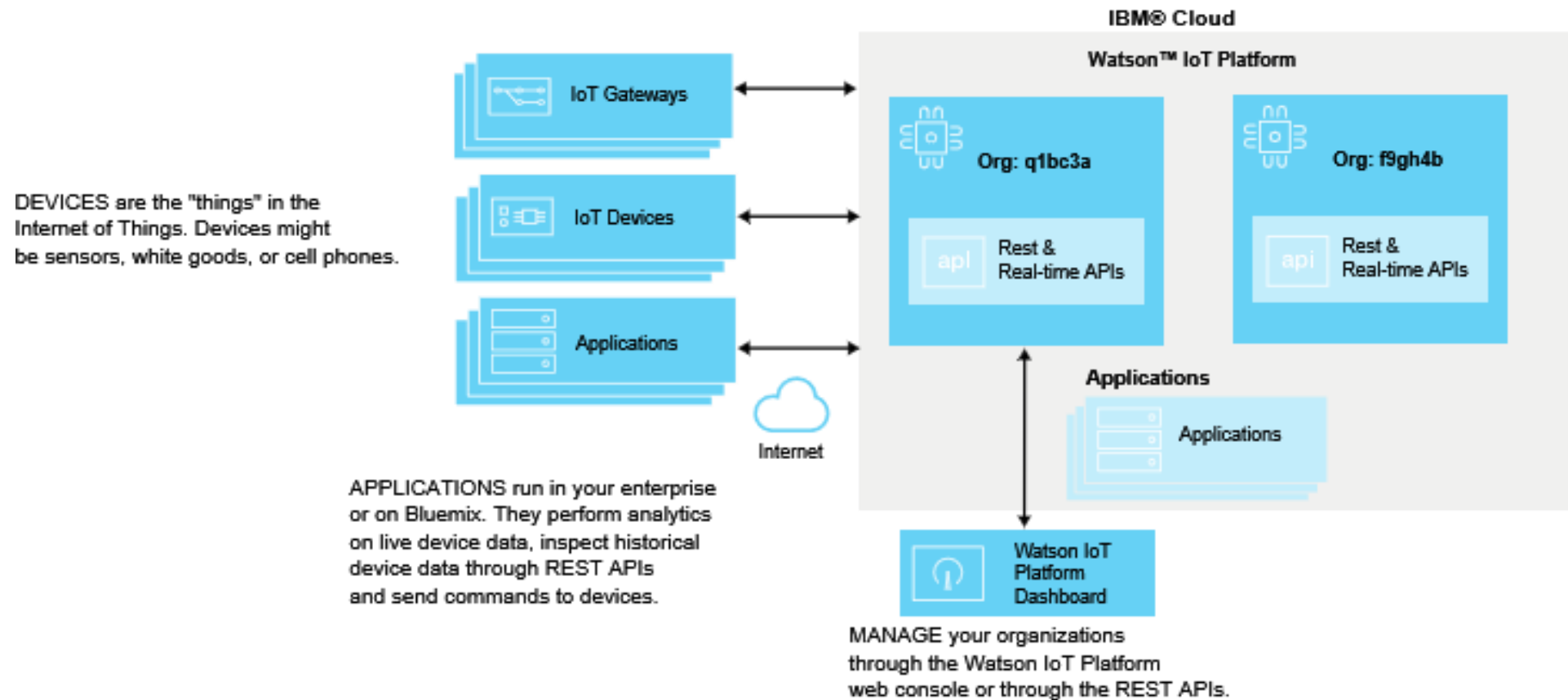
Chat Bot



Copyright © 2020 Infogain All rights reserved



Watson IoT Platform ORGANIZATIONS are how you can group common devices and applications. Devices and applications in the same organization work easily together.

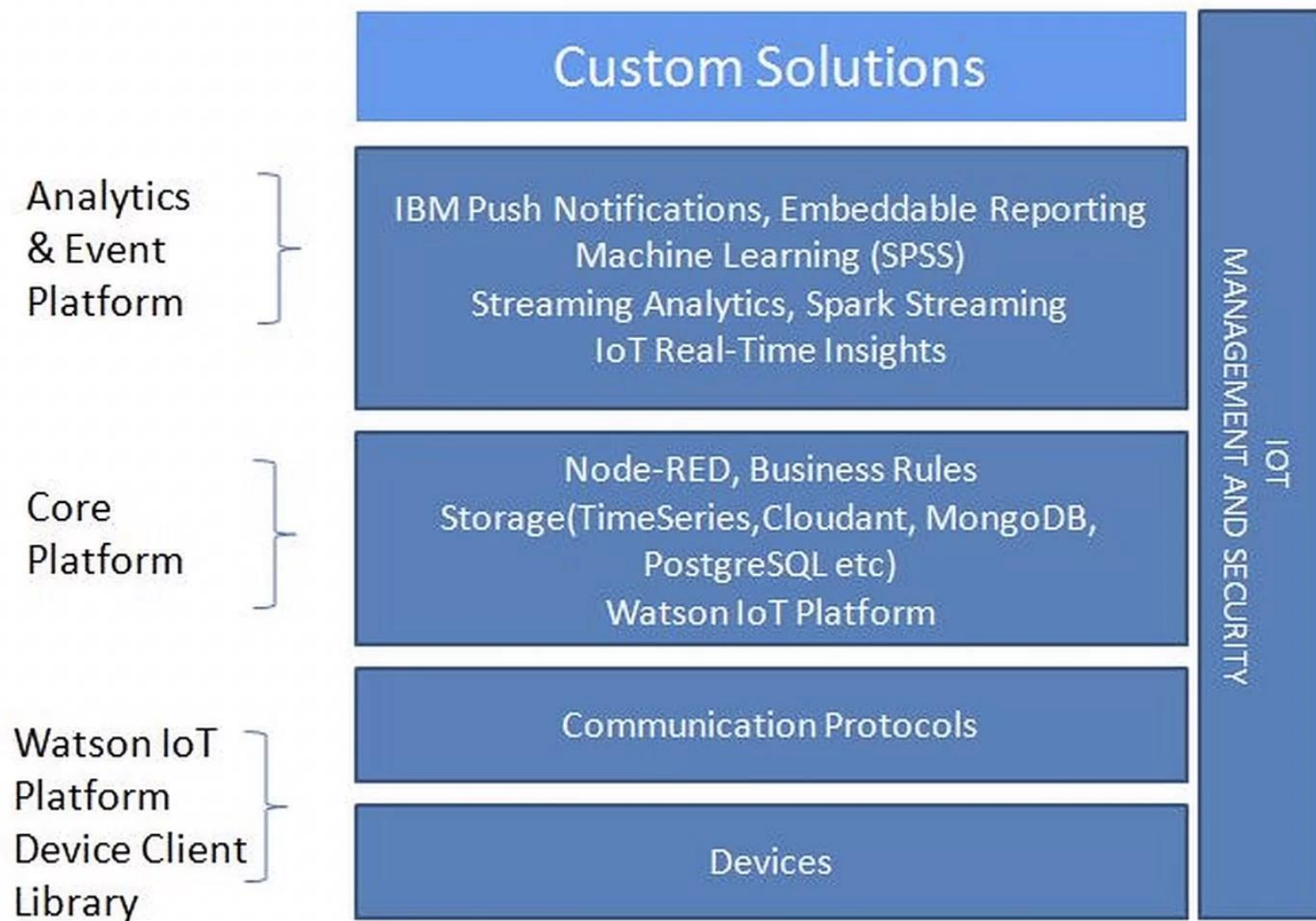


DEVICES are the "things" in the Internet of Things. Devices might be sensors, white goods, or cell phones.

APPLICATIONS run in your enterprise or on Bluemix. They perform analytics on live device data, inspect historical device data through REST APIs and send commands to devices.

MANAGE your organizations through the Watson IoT Platform web console or through the REST APIs.

IoT Stack Mapping for IBM IoT





IBM Watson IoT Platform

PROS

- ✓ Scalability
- ✓ Advanced Analytics
- ✓ Security
- ✓ Integration with other IBM Services
- ✓ Industry-Specific Solutions

CONS

- ✗ Complexity
- ✗ Cost
- ✗ Learning Curve
- ✗ Supplier Lock-In
- ✗ Limited Device Support

An **API endpoint** of entry is a specific URL that stands for the resource or functionality provided by an API ([Application Programming Interface](#)).

It acts as the interface filter where requests or information to/from a client are received and forwarded to the server that's hosting the API.

API endpoints consist of several key components that define how clients communicate with a server. These components help in structuring requests and generating appropriate responses.

API is a Big Service



Endpoint is like a doorway to a particular room

Endpoint Definition

Every API endpoint represents a specific resource or functionality provided by the API, allowing clients to interact with it.

- It defines what operation or data the endpoint provides.
- Developers can identify available endpoints through API documentation.

Specific URL



Specific function of API

HTTP Requests

Clients only interact with the API endpoints by using the HTTP ([Hypertext Transfer Protocol](#)) request-response model. The most common HTTP methods used are:

GET: Retrieves data from the server without modifying any existing resource. **E.g Adhar card retrives from e-sevai**

POST: Sends new data to the server to create a new resource. (**creating new adhar**)

PUT: Updates or replaces an existing resource with new data. (**updating adress**)

DELETE: Removes a specified resource from the server. (**deleting adhar**)

URL Structure

Each API endpoint is accessed through a unique URL that enables clients to communicate with the API.

Base URL: The base URL represents the main API server where requests are sent.

Path: The path specifies the exact endpoint that defines the requested resource or action.

Request Parameters

API endpoints often require additional data to process requests.

Query Parameters: Sent in the URL to filter or search data (e.g., /users?role=admin).

Path Parameters: Used in the URL to identify a specific resource (e.g., /users/101).

Request Body: Sent with POST or PUT requests to provide data for creating or updating

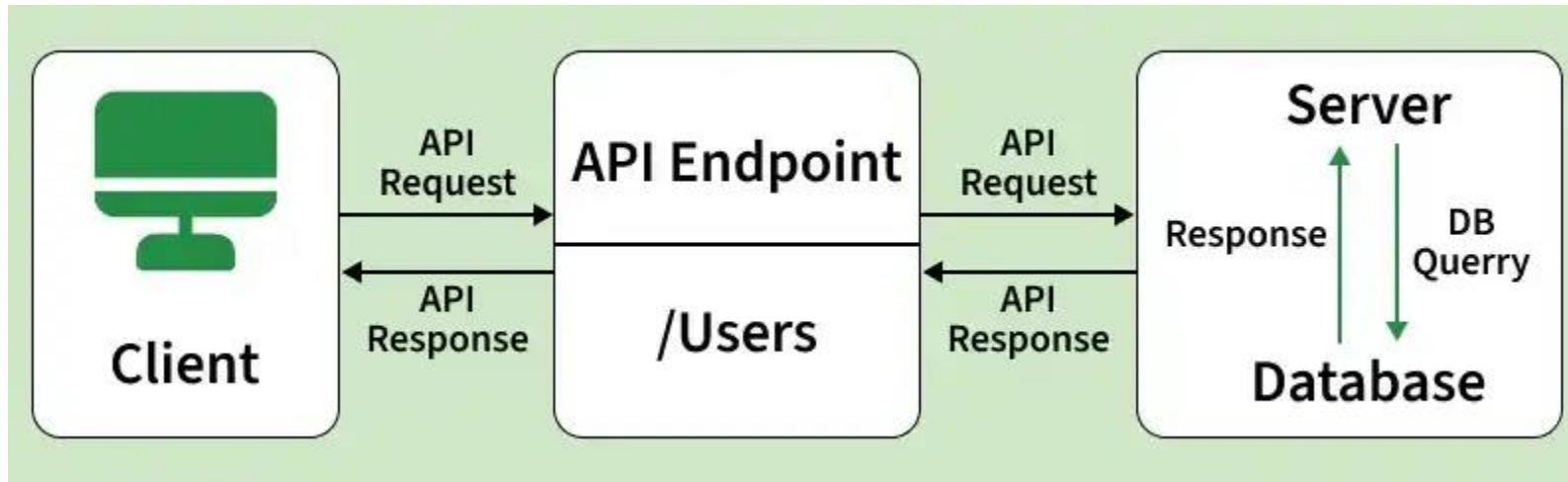
After receiving a request, the server processes it and sends back a response to the client.

Status Code: Indicates whether the request was successful or encountered an issue (e.g., 200 OK, 404 Not Found).

Response Data: Contains the requested information or confirmation of the action performed.

Error Message: Provides details when the request cannot be completed successfully.

Working of API Endpoints



API endpoints act as communication points between a client and a server.

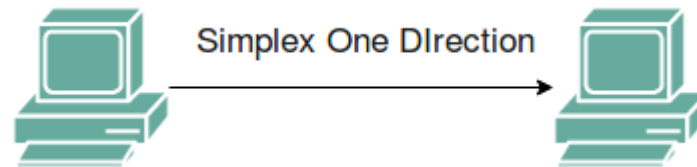
- **Client Sends Request:** The client (such as a browser or mobile app) sends an HTTP request to a specific API endpoint using methods like GET, POST, PUT, or DELETE.
- **Endpoint Receives Request:** The API endpoint identifies the requested resource (e.g., /users) and forwards the request to the appropriate server logic.
- **Server Processes Request:** The server performs the required operation such as fetching, creating, updating, or deleting data.
- **Database Interaction:** If required, the server communicates with the database to retrieve or modify data.
- **Response is Generated:**
- **Response Sent Back:**

Device creation and data transmission

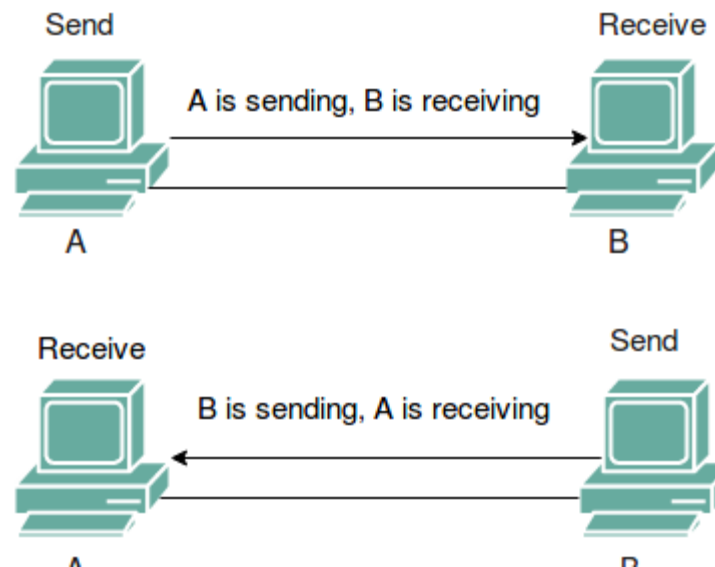
Describe how data (digital information) is exchanged between two devices over a communication channel. They specify whether information moves in a single direction or in both directions.

Directional Modes: Simplex vs. Half-Duplex vs. Full-Duplex

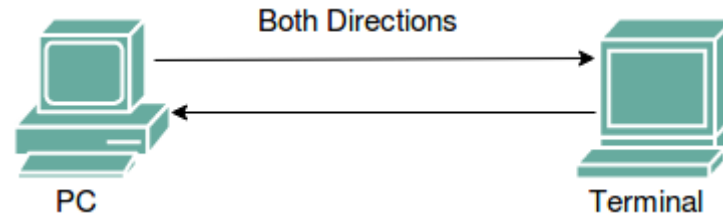
Simplex: Data flows in one direction only, making it ideal for broadcasting applications.



Half-duplex: Data flows in both directions, but only one direction at a time, suitable for error detection and controlled communication.



Full-duplex: Data flows simultaneously in both directions, perfect for real-time interactions.



Synchronization Methods: Synchronous vs. Asynchronous

Synchronous: Utilizes clock signals for precise timing, beneficial for high-speed data transfers.

Asynchronous: Employs start/stop bits, adaptable for varying hardware capabilities.

Bit Transfer Techniques: Serial vs. Parallel

Serial: Sends data bits sequentially, optimal for long-distance transfers.

Parallel: Transmits multiple bits simultaneously, ideal for short-distance, high-volume data movement.

Data transmission methods and protocols:

There are different communication methods and protocols used for data transmission, including:

1.Serial Transmission: In serial transmission, data is sent sequentially, bit by bit, over a single channel. Each bit is sent one after the other along the same path. This method is commonly used in communication over long distances and is known for its simplicity and reliability.

2.Parallel Transmission: Parallel transmission involves sending multiple bits simultaneously over multiple channels. Each bit travels along its own channel, allowing for faster data transfer rates compared to serial transmission. However, parallel transmission can be more susceptible to signal interference and crosstalk, especially over long distances.

3.Packet Switching: Packet switching is a method where data is divided into smaller packets before transmission. Each packet contains a portion of the original data along with routing information. These packets are then sent independently over the network and reassembled at the destination. Packet switching is the basis for communication on the internet and is known for its efficiency and ability to handle variable network conditions.

4.Circuit Switching: Circuit switching establishes a dedicated communication path between the sender and receiver before data transmission begins. This path remains active for the duration of the communication session. Circuit switching is commonly used in traditional telephone networks and provides a consistent connection but may be less efficient for bursty data transmission.

5.Wireless Transmission: Wireless transmission utilizes electromagnetic waves, such as radio waves or microwaves, to transmit data through the air. Wireless technologies include Wi-Fi, Bluetooth, cellular networks, and satellite communication. Wireless transmission offers the flexibility of mobility and eliminates the need for physical cables but may be susceptible to interference and signal attenuation.

1. Multiplexing: Multiplexing is a method of combining multiple data streams into a single transmission channel. This allows for the efficient use of network resources by transmitting multiple signals simultaneously. Common multiplexing techniques include time-division multiplexing (TDM), frequency-division multiplexing (FDM), and code-division multiplexing (CDM).

2. [HTTP protocol](#) defines how data is formatted and transmitted between a client and a server over a network, typically the Internet. It specifies the rules for requesting and responding to resources, such as web pages, images, or files, allowing for efficient and standardized communication.

The most common standards of data formatting are:

Data Representation: Data transmission format begins with representing information in a digital form, typically using binary digits (0s and 1s). These binary digits represent the smallest units of data and are grouped together to encode meaningful information.

Data Units: Data transmission involves organizing data into units for efficient transmission and processing. Common data units include:

Bit - The smallest unit of data, representing a single binary digit.

Byte - A group of 8 bits, commonly used as the basic unit of data storage and transmission.

Packet - A structured unit of data containing a header (metadata) and payload (actual data). Packets are used for efficient transmission across networks and may be further encapsulated in higher-level protocols.

Data Encoding: Data encoding transforms raw data into a format suitable for transmission over communication channels. Encoding schemes may include:

ASCII - American Standard Code for Information Interchange (ASCII) is a character encoding standard that represents text characters using 7 or 8 bits.

Unicode - Unicode is a character encoding standard that supports a wider range of characters and symbols from various writing systems worldwide.

Binary Encoding - Binary encoding represents data directly as sequences of binary digits, suitable for transmission over digital communication channels.

Data Formatting: Data transmission format includes organizing data into structured formats for communication. Common formats include:

Text - Textual data is represented as sequences of characters using character encoding schemes such as ASCII or Unicode.

Binary - Binary data consists of sequences of binary digits representing numerical values, encoded data, or machine instructions.

Structured Data - Structured data formats, such as JSON (JavaScript Object Notation) and XML (eXtensible Markup Language), organize data into hierarchical structures for easy parsing and interpretation.

Node-RED is a stream-based development tool for visual programming, primarily focused on wiring the Internet of Things (IoT). This programming tool is developed to wire together hardware devices, APIs, and online services in innovative ways.

- It is an open-source, flow-based, low-code programming tool that uses a visual, browser-based editor for wiring together hardware devices, APIs, and online services
- Developed by IBM Emerging Technology, Node-RED is an open-source programming tool.
- It operates by passing messages between nodes. The messages in Node-RED are simple JavaScript objects that can have any set of properties.
- Node-RED simplifies application development through a drag-and-drop interface, making it accessible to both experienced programmers and non-developers.

Nodes: These are pre-built functional blocks (e.g., input, output, processing, network nodes like MQTT or HTTP) that perform specific tasks. A vast library of over 5,000 community-contributed nodes is available.

Flows: Applications are built by connecting nodes with wires in a workspace (organized into tabs) to define the path and logic of data or events. These flows are stored in JSON format, which can be easily shared and imported.

Messages: Data is passed between nodes in the form of simple JavaScript objects called messages, which typically contain a payload property.

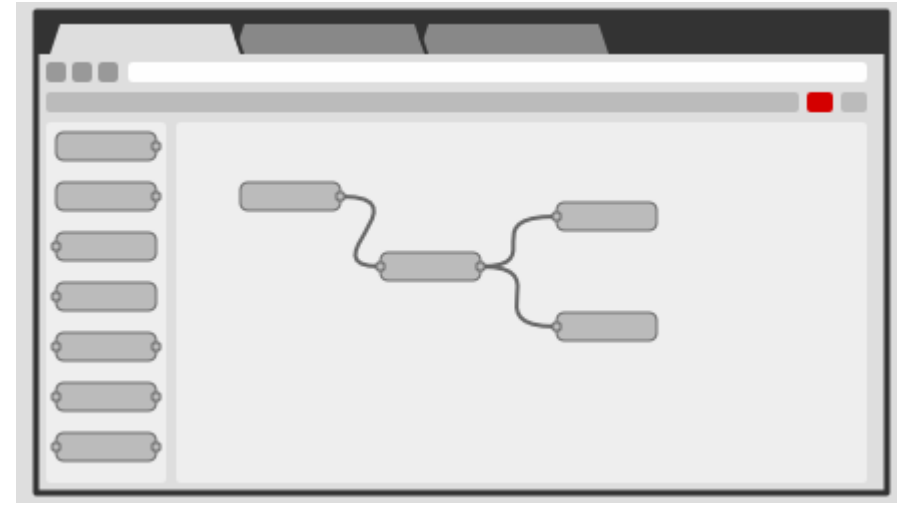
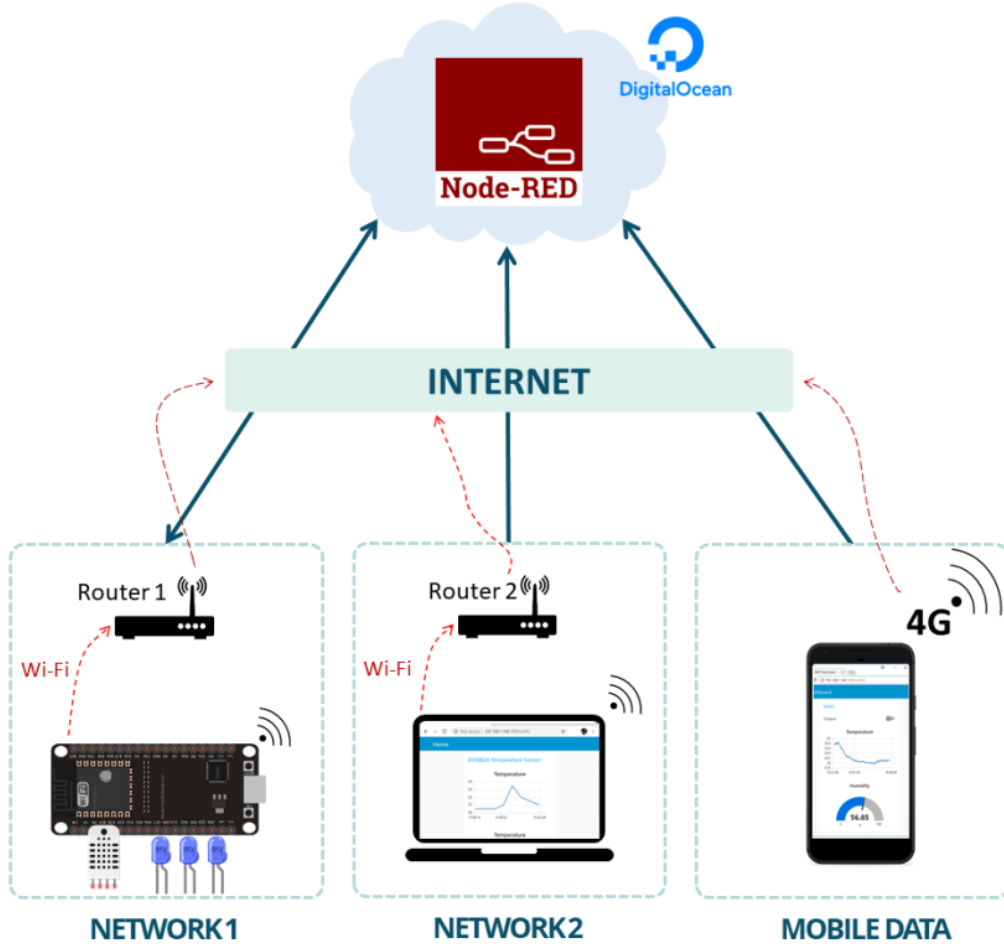
Editor: The entire environment runs in a web browser, providing a palette of nodes on the left, a workspace in the center, and a sidebar for debugging, help, and configuration on the right.

Types of Nodes in Node-RED

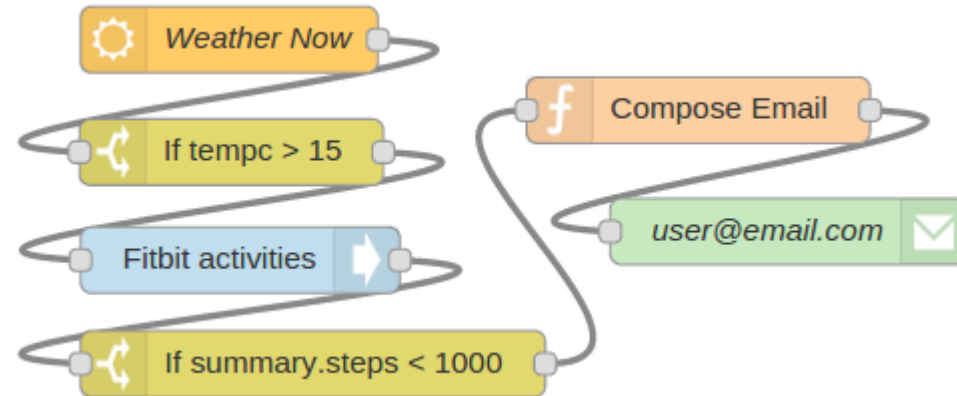
- **Inject Node:** This node allows [manual triggering of flows and facilitates injecting events at scheduled intervals](#).
- **Debug Node:** [Displays the content of a message](#), either the payload or the entire object, which is [useful for debugging](#).
- **Template Node:** Modifies the output based on a Mustache (logic-less) template.

How different components interconnected

CLOUD NODE-RED DASHBOARD



Data flow and Logic



Installing NODE-RED

Installing with npm (Node package manager)

```
sudo npm install -g --unsafe-perm node-red
```

To upgrade to the latest version:

```
sudo npm install -g --unsafe-perm node-red
```

Installing with docker

```
docker run -it -p 1880:1880 --name mynodered nodered/node-red
```

Installing with snap

```
sudo snap install node-red
```

Once Node-RED is installed and set up, it can be run in three ways:

- Locally
- On a device
- In a cloud environment

Cross-Platform Compatibility:

Node-RED can run on various operating systems, including **Raspbian**, **Ubuntu**, Windows, **Docker**, Debian, and **macOS**.

Cloud Platforms:

IBM Cloud

SenseTecnica FRED

Amazon Web Services

Microsoft Azure

Pre-installed Device:

Raspberry Pi

BeagleBone Black

Interacting with Arduino

Android

- Data-centric world, the capacity to analyze and interpret data is indispensable for businesses, researchers, and decision-makers.
- Data analytics involves scrutinizing raw data to discern patterns, derive conclusions, and bolster decision-making.
- **Compounded Annual Growth Rate (CAGR) of 13.5%** from 2021 through 2030.

What is Data Analytics?

Data analytics involves analyzing raw data to discover patterns, make informed conclusions, and aid decision-making. This process includes transforming data into valuable insights that improve processes, elevate customer experiences, and foster innovation.

The [process of data analytics](#) typically involves various essential steps:

Data Collection: Collecting data from diverse sources like databases, sensors, and online platforms.

Data Cleaning: Ensuring that the data is precise, thorough, and free from errors or discrepancies.

Data Analysis: Applying statistical and computational techniques to identify patterns and trends.

Data Visualization: Creating visual representations of the data to make it easier to understand and interpret.

Data Interpretation: Synthesizing insights and leveraging analyzed data to arrive at well-founded decisions.

Several tools and technologies are pivotal in data analytics, each offering unique functionalities and advantages.

•**Excel:**

•**SQL:** SQL, also known as Structured Query Language, plays a crucial role in the management and querying of relational databases.

•**R:** R is a programming language and environment crafted specifically for statistical computing and visualizing data. It's favored for its extensive libraries and robust data analysis capabilities.

•**Python:** Python stands out as a versatile programming language renowned for its robust libraries like Pandas, NumPy, and Matplotlib, which empower data analysts and scientists with readability and flexibility, making it a top choice in their toolkit.

•**Tableau:** Tableau stands out as a premier data visualization tool, empowering users to craft interactive dashboards that can be easily shared. It simplifies the process of visualizing complex data and sharing insights with stakeholders.

Applications of Data Analytics

In business and marketing, data analytics helps organizations understand customer behavior, optimize marketing strategies, and improve sales performance.

Customer Segmentation: Identifying different customer groups based on purchasing behavior and preferences.

Marketing Campaign Analysis: Assessing the impact of marketing campaigns and strategies.

Sales Forecasting: Predicting future sales trends to make informed business decisions.

In the **finance sector**, data analytics plays a critical role in managing risks, detecting fraud, and making investment decisions.

Risk Management: Assessing and mitigating financial risks by analyzing market trends and economic indicators.

Fraud Detection: Identifying fraudulent activities through anomaly detection and pattern recognition.

In **Education**, data analytics helps institutions track student performance, personalize learning experiences, and enhance curriculum development.

Student Performance Tracking: Monitoring student progress to identify areas of improvement and provide targeted support.

Curriculum Improvement: Analyzing educational data to improve and enhance curriculum design.³

In **Healthcare**, data analytics is used to improve patient outcomes, enhance operational efficiency, and advance medical research.

- **Patient Data Analysis:** Analyzing patient records to identify trends and improve treatment plans.
- **Predictive Modeling for Diseases:** Using historical data to predict disease outbreaks and patient health risks.

Hadoop

Google file system (GFS)

Map Reduce

Yahoo ----- Nutch

Doug cutting developed **Hadoop**



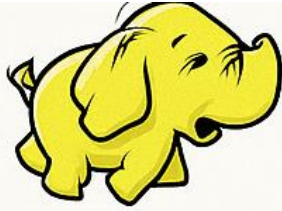
Technology

- HDFS- Hadoop distributed file system
- YARN- Yet another Resource negotiator
- Map Reduce: Data processing algorithm
- PIG and HIVE
- APPACHE AMBARI { Administrator } it administrate for complete system

E.g if data processing algorithm done in Java and python, SQL you can done in PIG and HIVE

- Hadoop is an open-source software framework that is used for storing and processing large amounts of data in a distributed computing environment.
- Designed to handle big data and is based on the MapReduce programming model, which allows for the parallel processing of large datasets.
- Designed to process large volumes of data (Big Data) across many machines without relying on a single machine.
- Its based on **Java programming** with some native code in **C** and **shell scripts**.
- Hadoop works by connecting many inexpensive computers (called nodes) in a cluster.

Hadoop



Hadoop Core Components

HDFS

YARN

Hadoop
MapReduce

Hadoop Ecosystem

Apache
ZooKeeper

Apache
Pig

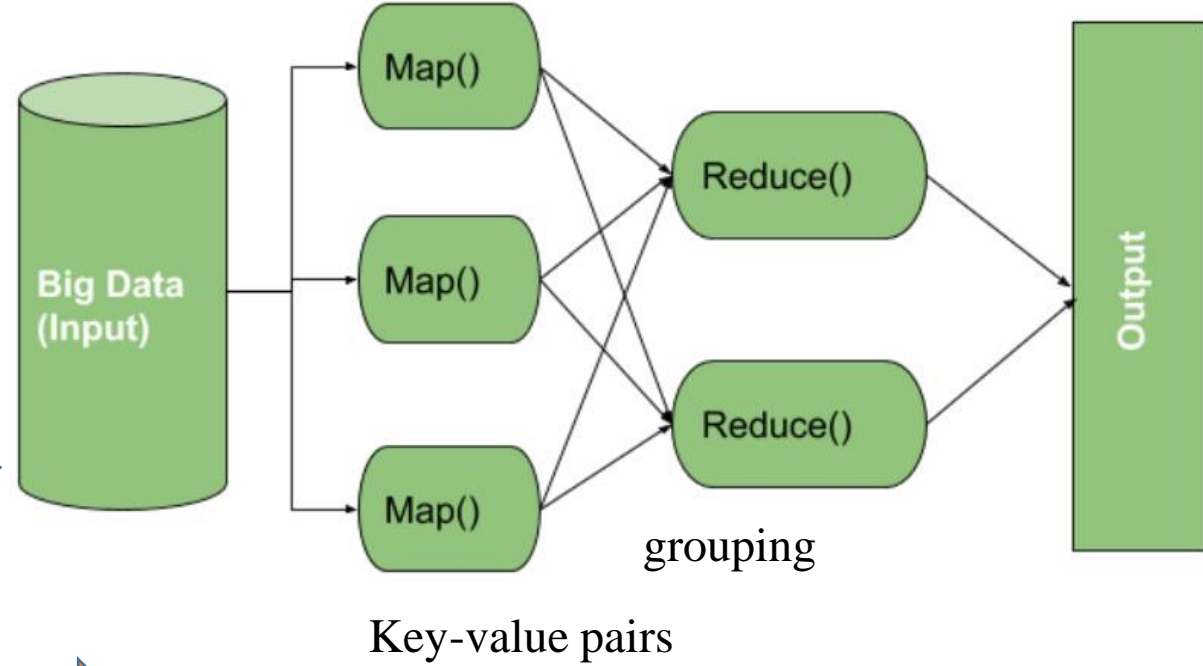
Apache
Sqoop

Apache
HBase

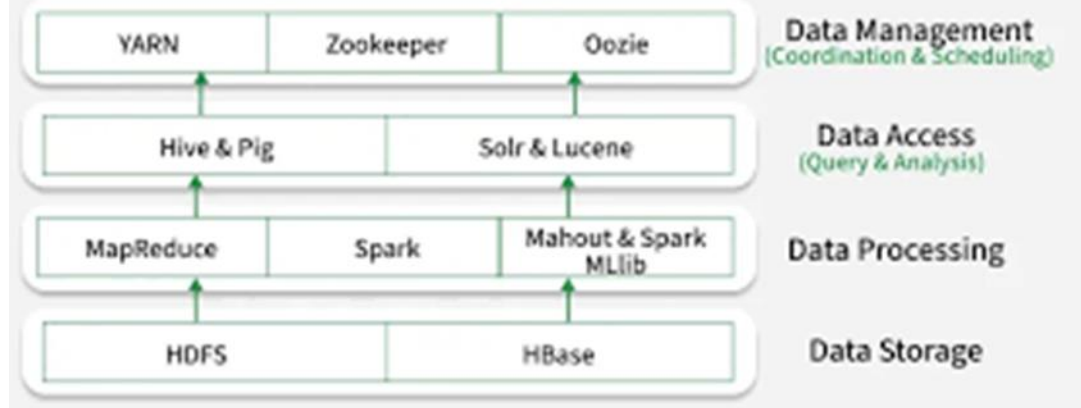
Apache
Flume

Apache
Spark

MapReduce Programming Model



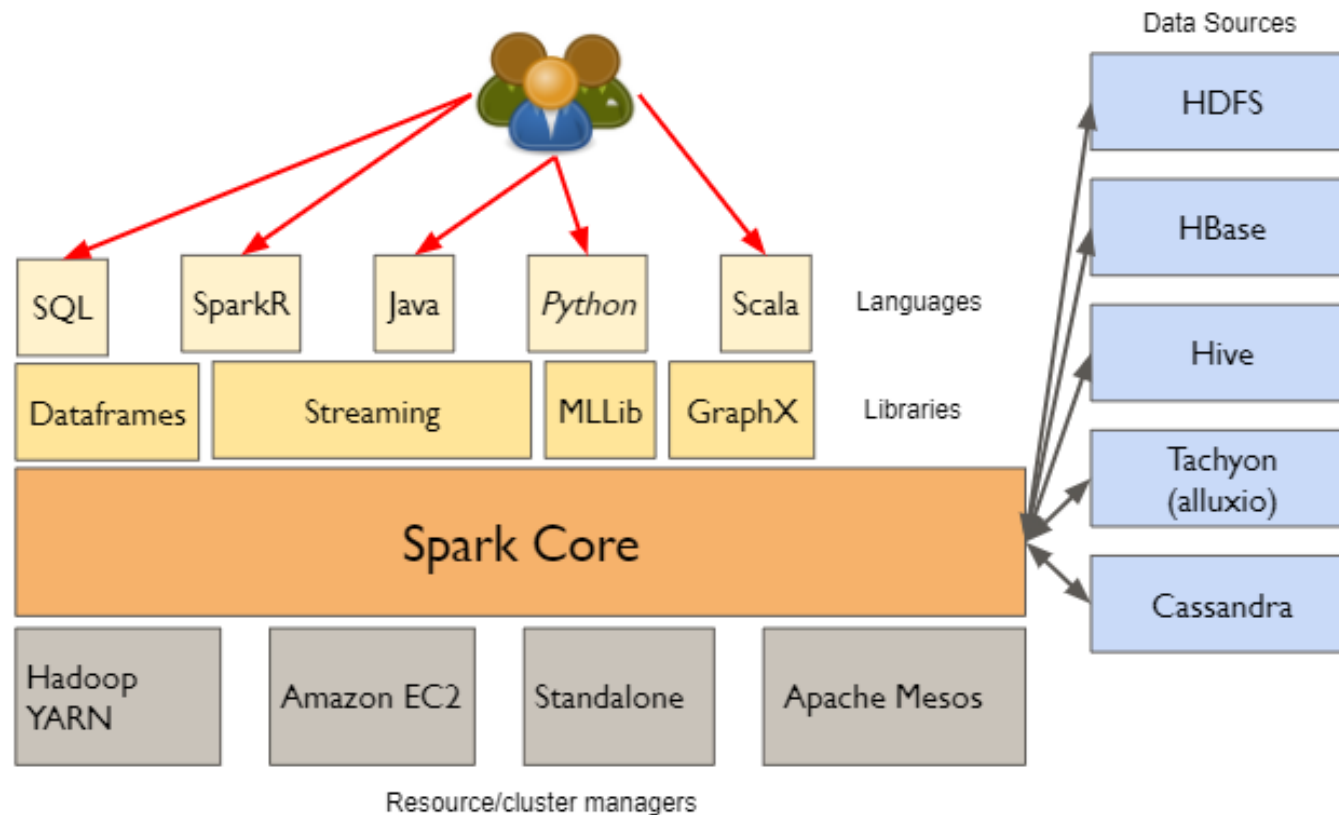
Hadoop Ecosystem



Apache Spark is a unified, open-source engine designed for large-scale data processing.

Databrick's definition "Apache Spark is a lightning-fast unified analytics engine for big data and machine learning. It was originally developed at UC Berkeley in 2009

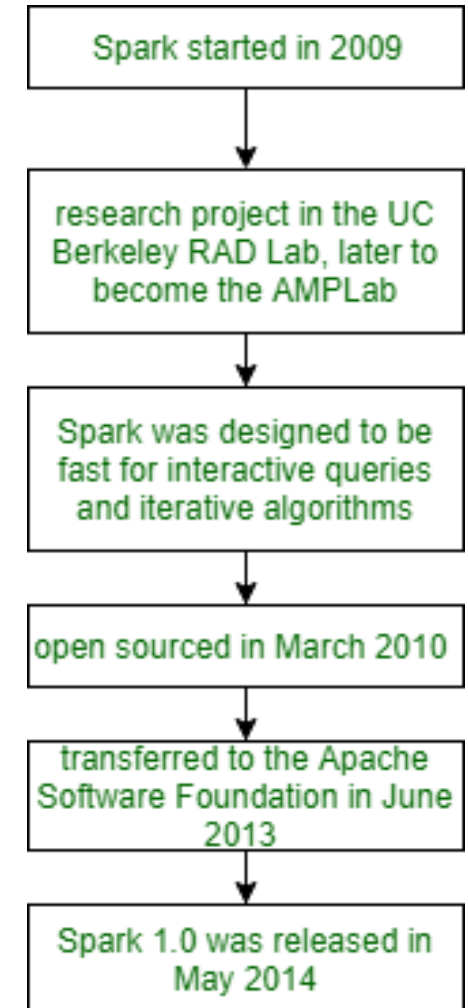
Spark Architecture

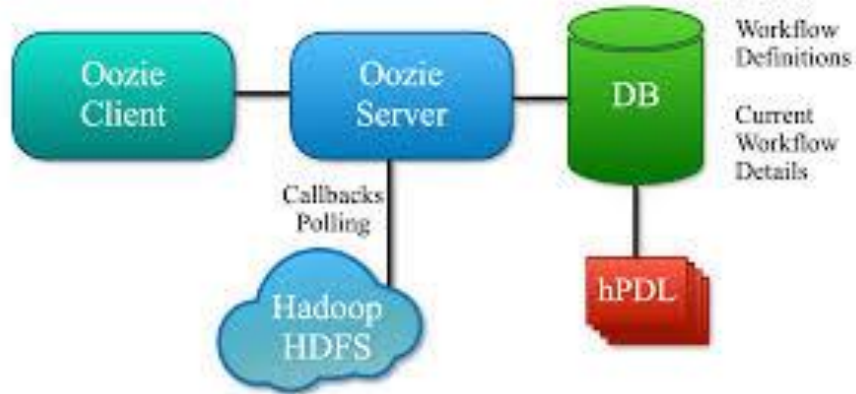


Databricks is one of the major contributors to Spark includes yahoo! Intel etc.

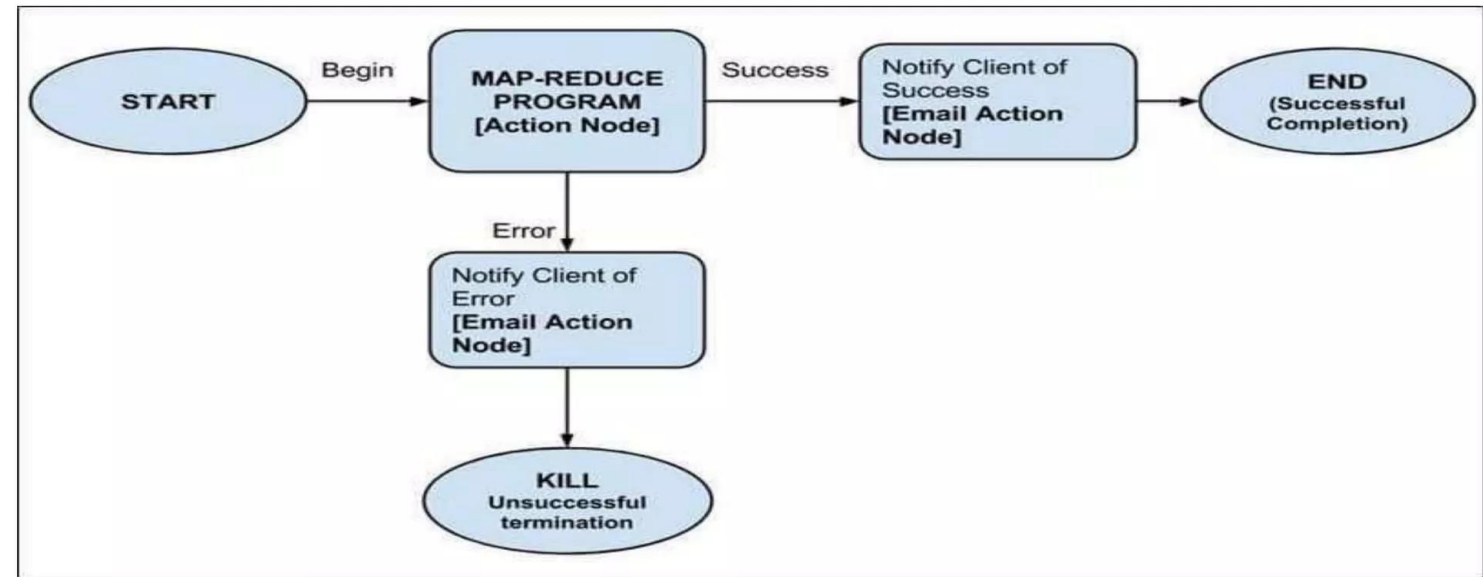
Features of Spark :

- Apache spark can use to perform **batch processing**.
- Apache spark can also use to perform **stream processing**. For stream processing, we were using Apache Storm / S4.
- It can be used for interactive processing.
- Previously we were using Apache Impala or Apache Tez for **interactive processing**.
- Spark is also useful to perform **graph processing**. Neo4j / Apache Graph was using for graph processing.



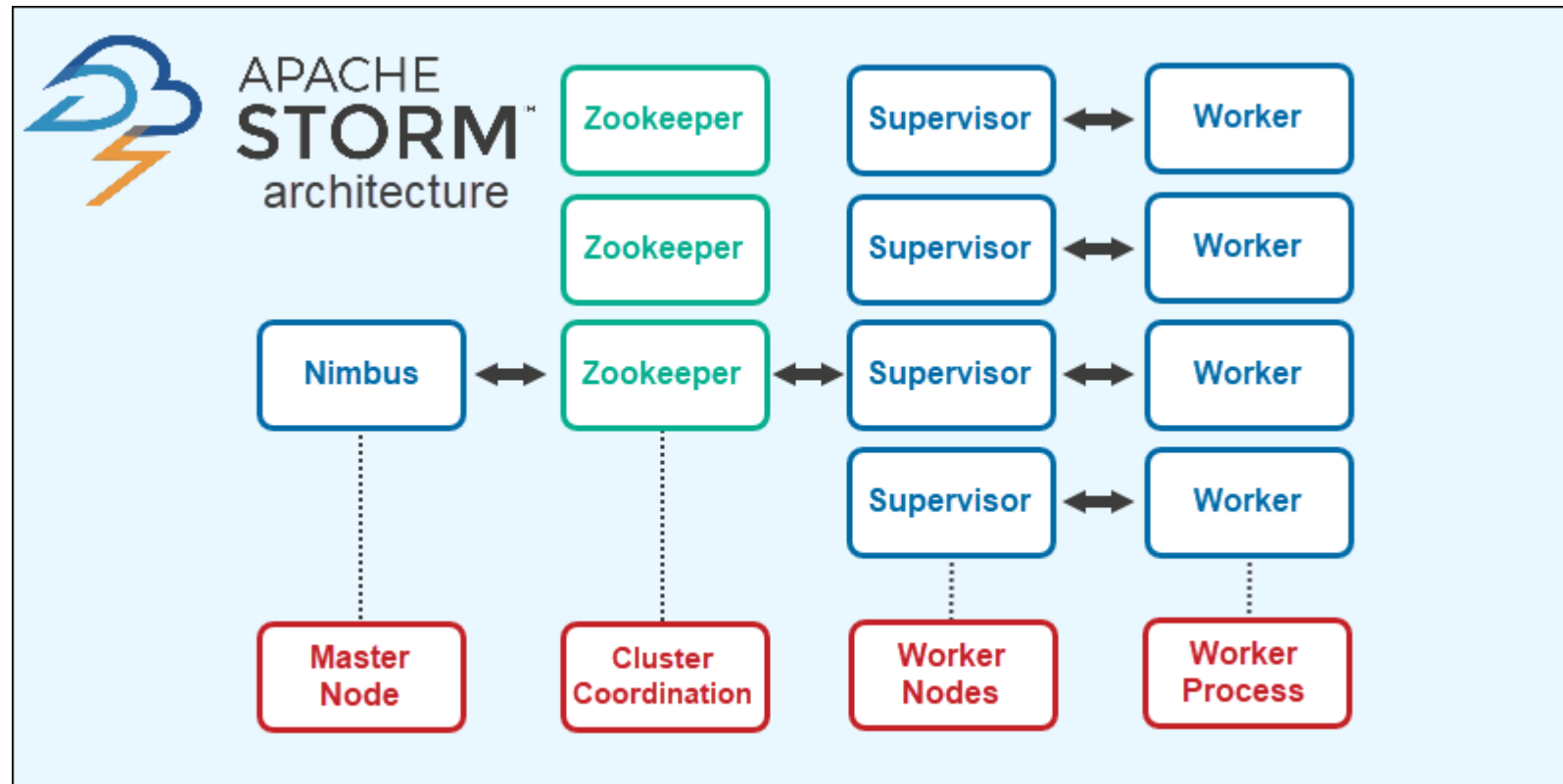


- Oozie is **scalable** and can **manage the timely execution** of thousands of workflow in a Hadoop cluster.
- Oozie is very much **flexible** as well one can easily **start ,stop, suspend and rerun jobs**.
- Oozie makes it very easy to **return failed workflow**.



Apache Storm is a distributed real-time big data-processing system. Storm is designed to process vast amount of data in a fault-tolerant and horizontal scalable method. It is a streaming data framework that has the capability of highest ingestion rates. Though Storm is stateless, it manages distributed environment and cluster state via Apache ZooKeeper. It is simple and you can execute all kinds of manipulations on real-time data in parallel.

Apache Storm is **continuing to be a leader in real-time data analytics**. Storm is easy to setup, operate and it guarantees that every message will be processed through the topology at least once.



Fault Tolerance: If a worker or supervisor dies, Storm restarts it. If the supervisor fails, Nimbus reassigns the tasks to another node.

Stateless: Nimbus and Supervisors are stateless; all state is stored in ZooKeeper, allowing them to be "fail-fast".

Parallelism Levels: Storm has four levels—Nodes, Workers (JVMs), Executors (Threads), Tasks (actual instances).

Stream Groupings: Defines how data is routed between bolts (e.g., **Shuffle** for random distribution, **Fields** for directing identical values to the same task).

Backpressure: Storm handles overloaded downstream components by throttling upstream spouts, preventing memory issues.

Performance Tuning: 1 worker per node/topology, 1 executor per core for CPU-bound tasks, and higher executor-to-core ratios for I/O-bound tasks.

Storm	Hadoop
Real-time stream processing	Batch processing
Stateless	Stateful
Master/Slave architecture with ZooKeeper based coordination. The master node is called as nimbus and slaves are supervisors .	Master-slave architecture with/without ZooKeeper based coordination. Master node is job tracker and slave node is task tracker .
A Storm streaming process can access tens of thousands messages per second on cluster.	Hadoop Distributed File System (HDFS) uses MapReduce framework to process vast amount of data that takes minutes or hours.
Storm topology runs until shutdown by the user or an unexpected unrecoverable failure.	MapReduce jobs are executed in a sequential order and completed eventually.
Both are distributed and fault-tolerant	
If nimbus / supervisor dies, restarting makes it continue from where it stopped, hence nothing gets affected.	If the JobTracker dies, all the running jobs are lost.

Twitter – Twitter is using Apache Storm for its range of Publisher Analytics products. Publisher Analytics Products process each and every tweets and clicks in the Twitter Platform. Apache Storm is deeply integrated with Twitter infrastructure.

NaviSite – NaviSite is using Storm for Event log monitoring/auditing system. Every logs generated in the system will go through the Storm. Storm will check the message against the configured set of regular expression and if there is a match, then that particular message will be saved to the database.

Wego – Wego is a travel metasearch engine located in Singapore. Travel related data comes from many sources all over the world with different timing. Storm helps Wego to search real-time data, resolves concurrency issues and find the best match for the end-user.