Boosting Research for a Smart and Carbon Neutral Built Environment with Digital Twins – **SmartWins**

Big Data concepts specific to smart buildings

Nikos Tsalikidis, CERTH

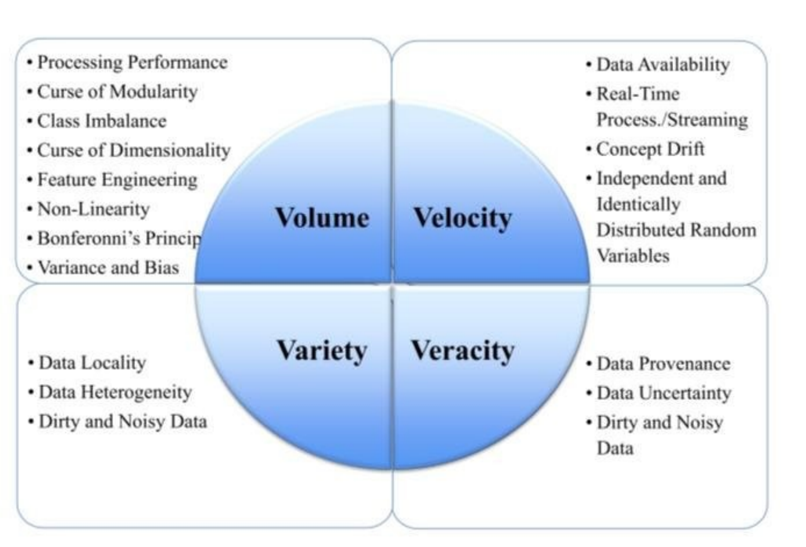
CERTH SmartWins Training Sessions: Day1
23 April 2024
Thessaloniki

Funded by the European Union

This project has received funding from the European Union's Horizon research and innovation programme under grant agreement No 101078997

Key concept Definitions

Definition: Big Data refers to extremely large datasets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions.



5 Vs of Big Data: Volume, Velocity, Variety, Veracity, and Value.

Retail giants like Amazon use Big Data to analyze customer purchases, searches, and online behavior to personalize shopping experiences and improve service delivery.

SmartWins

Big data and IoT

The management and analytics of big data generated from IoT sensors deployed in smart buildings pose a real challenge in today’s world.

The duology of IoT and, near real-time big data management and analytics is complex in nature. There is a lack of integrated and coherent frameworks today

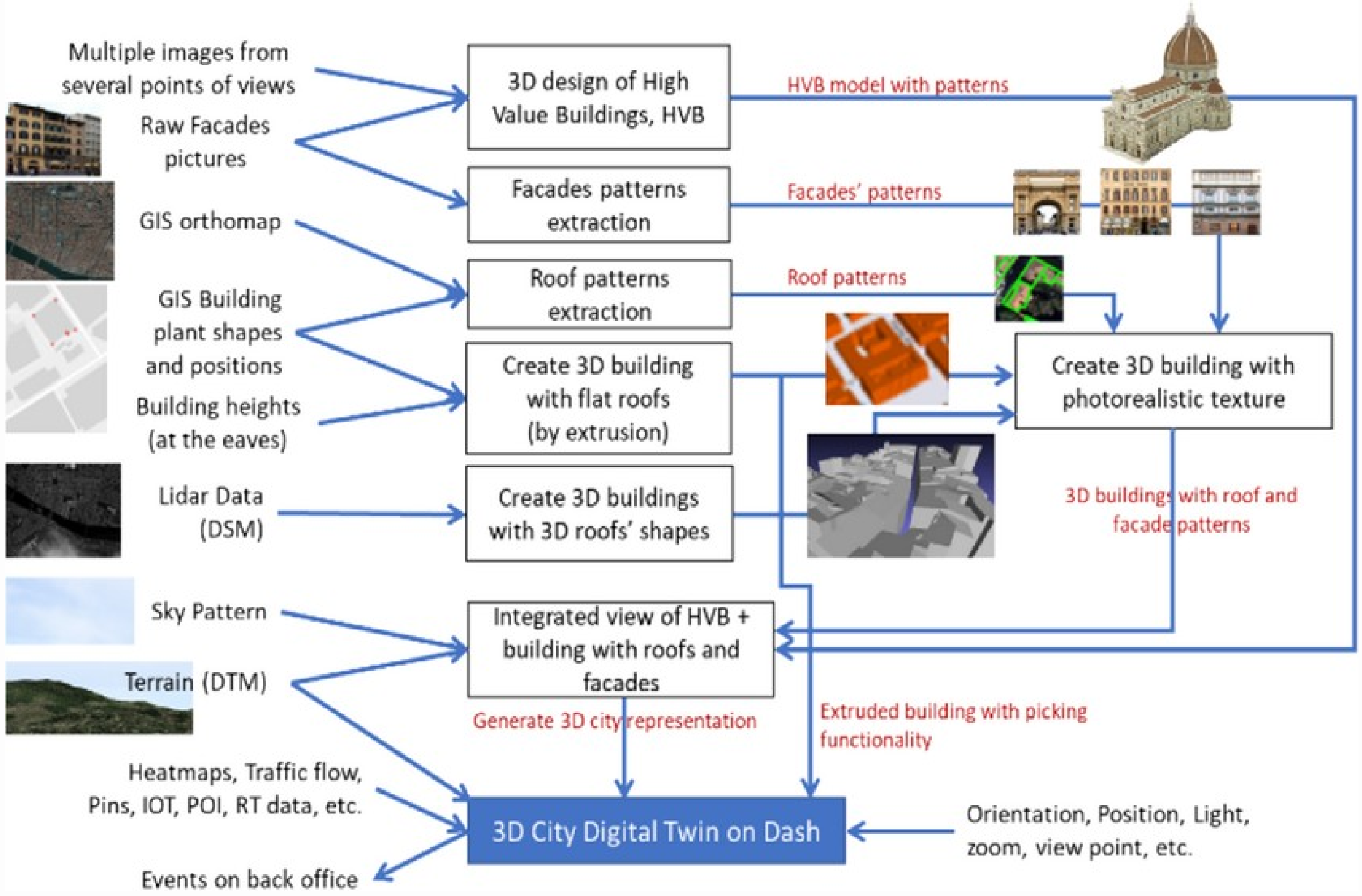
A clear need for an IoT focused Integrated Big Data Management and Analytics framework to enable the near real-time autonomous control and management of smart buildings.

SmartWins

Interconnection of Digital Twins and Big Data

Digital Twins leverage Big Data for enhanced predictive analytics by utilizing vast amounts of real-time and historical data to simulate future conditions and make informed decisions.

- Example: In urban planning, Digital Twins of cities integrate Big Data from traffic patterns, weather conditions, and population dynamics to optimize energy usage and reduce congestion.



(Source : https://www.researchgate.net/publication/374475646/figure/fig1/AS:11431281233773058@1712110392071/Data-flow-of-the-production-process-for-creating-a-Digital-Twin-for-smart-cities.png)

SmartTwins

Generic data management system in smart building

1. Infrastructure level of the input data

- Represents all the data sources generated by the connected objects in the building such as energy consumption, humidity level, indoor and outdoor temperature, number of alarm activation and deactivation, etc…

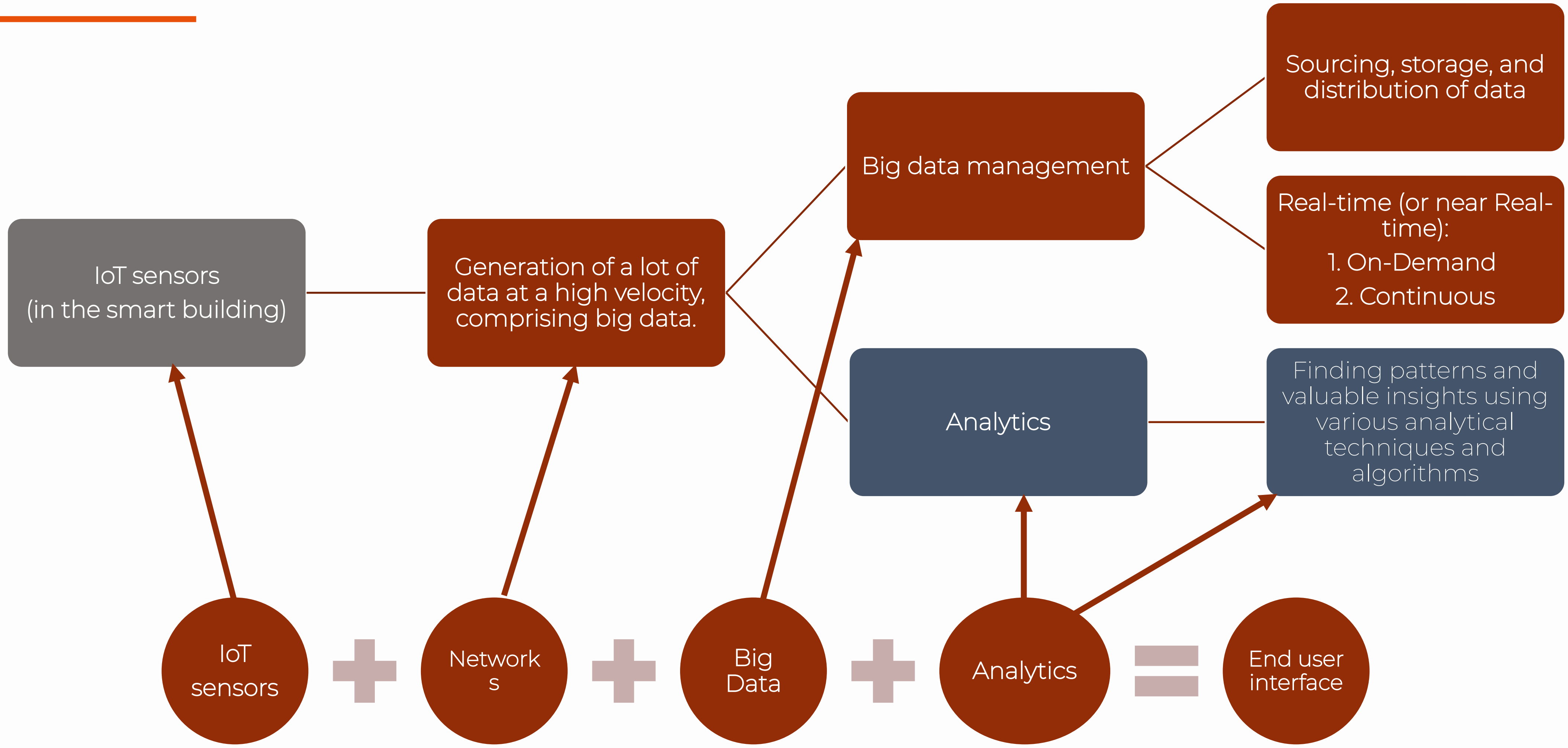
2. System infrastructure level

- Represents the core of the intelligent system since it allows the collection, processing, and merging and storage data in a NoSQL database.
- Allows use of data for knowledge extraction through data mining algorithms, automatic learning through ML algorithms or simply offering reporting services

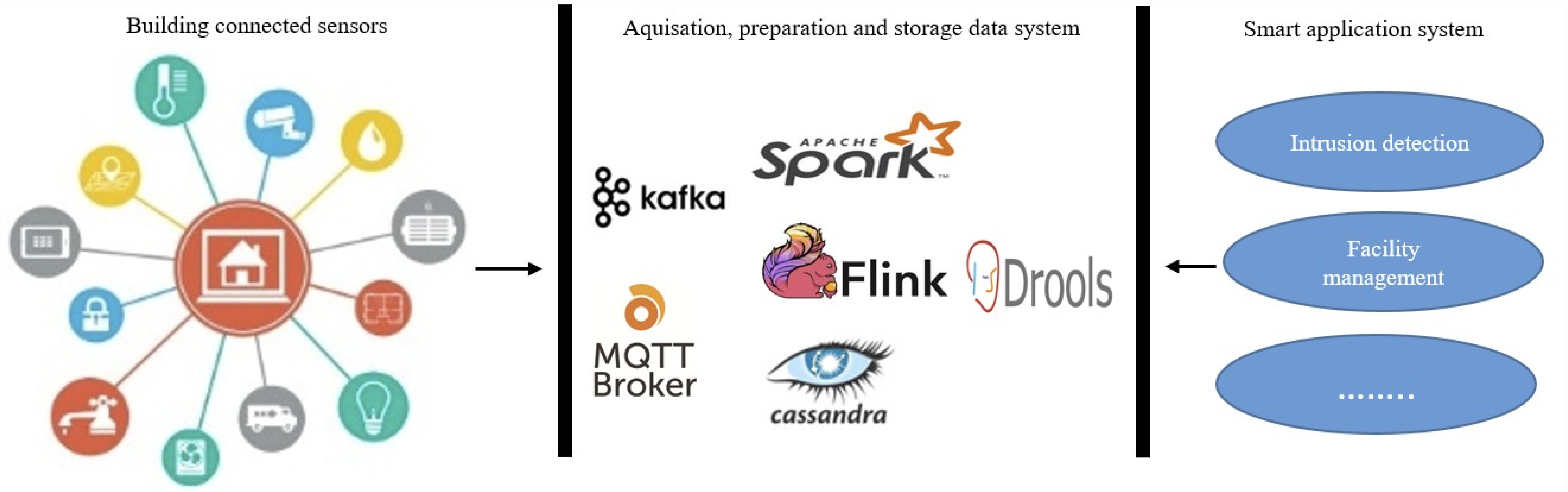
3. Service layer (smart building context)

- Represents the list of services offered by the system to building managers, residents and energy suppliers

Big Data management conceptual workflow



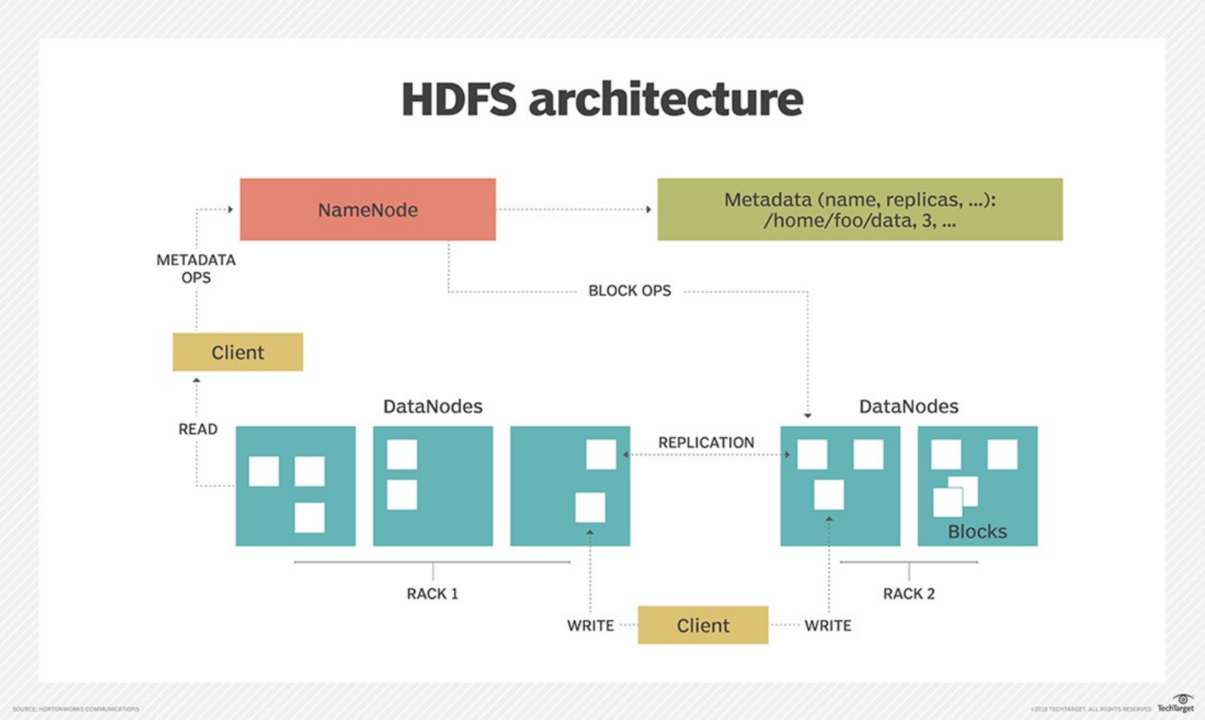
Big Data management conceptual workflow



Key Big data tools and technologies (in particular utilized with IoT data)

Hadoop Distributed File System (HDFS)

- **Overview:** primary data storage system used by Hadoop applications.
- **Key Features:**
  - Employs a NameNod/DataNode architecture to implement a distributed file system→provides high-performance access to data across scalable clusters
  - Data written on server once, then read/reused. NameNode→keeps track file data location in each cluster.



Apache Spark

Overview:

- Open-source distributed computing system for processing large data sets rapidly
- Unified analysis engine handling structured, semi-structured, and unstructured data.
- Used to analyze the data generated by IoT sensors

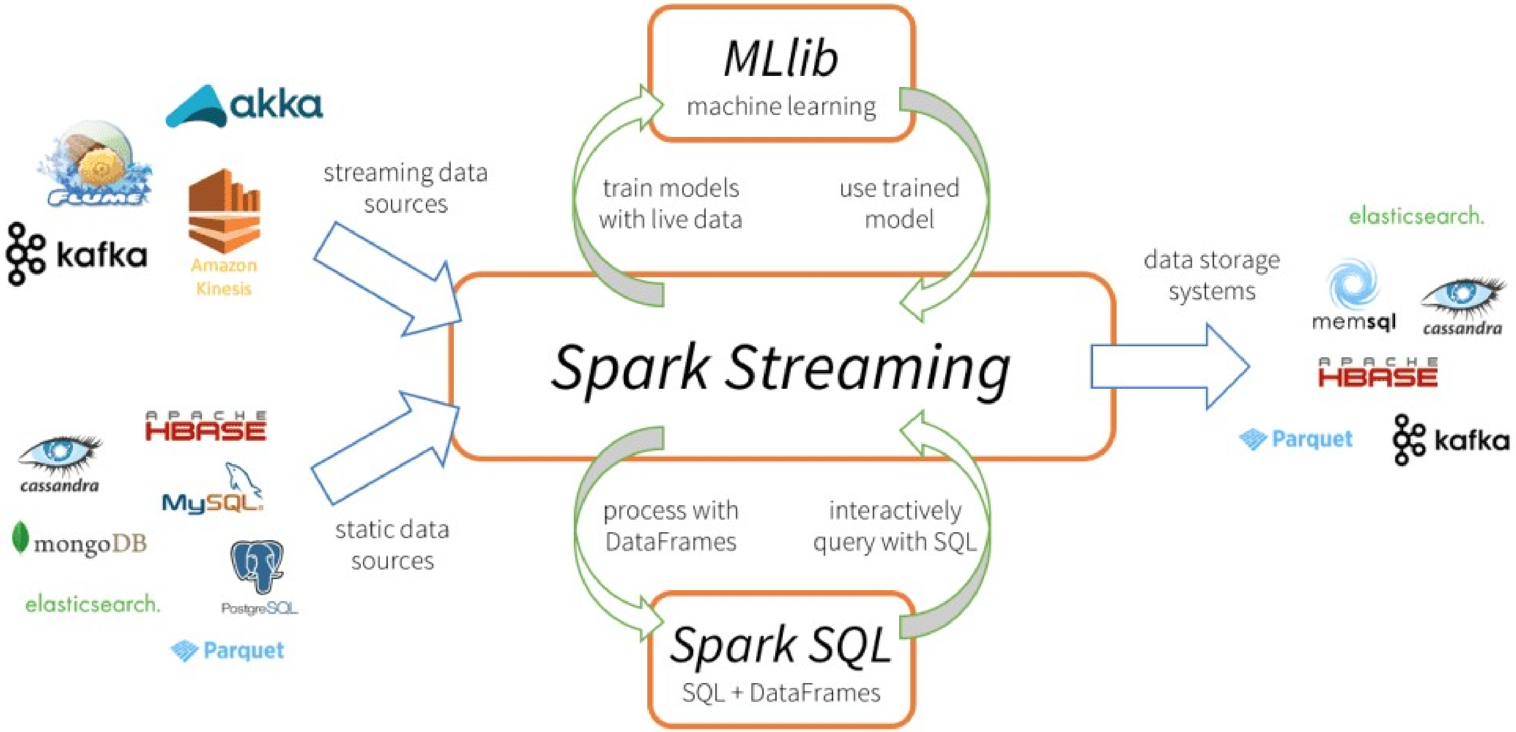
Key Features:

- **In-Memory Processing:**
  - Whereas Hadoop reads/writes files to HDFS, Spark processes data in RAM using a concept known as Resilient Distributed Dataset (RDD)
  - Ideal for real-time processing applications (e.g. financial fraud detection, sensor data analysis, recommendation systems)
- **Compatibility:**
  - Works on Hadoop Distributed File System (HDFS), supporting both real-time and batch mode.
  - Multiple APIs and libraries, including Spark SQL for structured data, Spark Streaming for real-time data, and MLlib for ML.
- **Programming Language Support:**
  - Supports Java, Python, Scala, and R, making it accessible to a wide range of developers.

SmartWins

Spark Streaming

- Extension of the core Spark API
- Process real-time data from various sources including Kafka, Flume, and Amazon Kinesis.
- Processed data can be pushed out to filesystems, databases, and live dashboards.
- Key abstraction: a Discretized Stream (Dstream)→represents a stream of data divided into small batches.



Apache Kafka

Overview:

- Open-source distributed event streaming platform initially developed by LinkedIn, later donated to Apache Software Foundation.
- Designed for handling high-performance real-time data streams with features for publishing, subscribing, and processing log streams.

Key Architecture:

- **Distributed, Partitioned, Replicated Registry:**
  - Allows multiple producers and consumers to interact with the same topic in a scalable and fault-tolerant manner.
  - Data stored in Kafka topics, which can be partitioned for scalability and performance.
- **APIs and Tools:**
  - Provides producer API, consumer API, and Kafka Streams API for building real-time data processing applications, messaging systems, or event-driven architectures.

Use Cases:

- Widely used in various industries such as financial services, social media, e-commerce, and IoT for its scalability, reliability, and fast real-time data management.

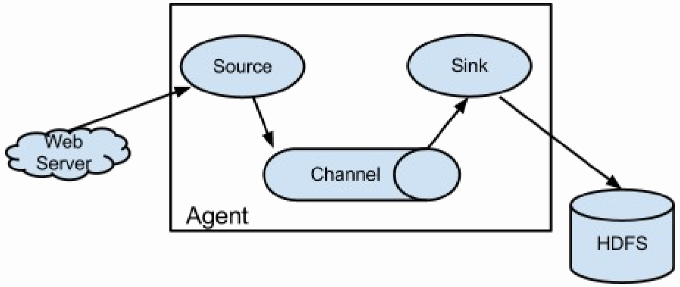
SmartWins

Apache Flume

Overview:

- Distributed, reliable, and available service for log data.
- Efficiently collects, aggregates, and moves large volumes of diverse data

Key Features:



- **Data Ingestion:**
  - Sensors' data ingested into HDFS through Flume's pipelines.
  - Handles batched and streaming data - logs, IoT, financial data
- **Efficient Data Movement:**
  - Facilitates seamless flow of data types.
  - Transfers data to a centralized location for streamlined analysis.

Strengths:

- Fault-tolerant tool with failover and recovery mechanisms.
- Simple, extensible data model for online analytical applications.

REST API

Overview: REST API layer facilitates standardized communication between systems. Enables integration of Digital Twin Front End with third-party applications.

Key Features:

1. **Simplicity and Ease of Use:**
   - Utilizes straightforward design with widely accepted HTTP methods.
   - Supports standard data formats like JSON or XML for developer convenience.
2. **Platform and Language Independence:**
   - Based on universally supported HTTP protocol.
   - Ensures compatibility with various platforms, programming languages, and devices.
3. **Flexibility and Modularity:**
   - Promotes software development flexibility by exposing specific resources through well-defined endpoints.
   - Facilitates easy upgrades and addition of new features without impacting the overall system.

Applications:

4. **Seamless System Integration:**
   - Enables integration of Digital Twin Front End with third-party applications.
5. **Additional Sensor Integration:**
   - Simplifies the process of incorporating additional sensors by leveraging API-generated data format compatibility.
6. **Handling Substantial Data Volumes:**
   - In complex scenarios w/ substantial data volumes, specialized strategies related to Big Data principles

PostgreSQL

- Open-source relational database management system
- Highly reliable, scalable, and extensible
- Supports multiple programming languages
- Advanced features: concurrency management, full-text search, triggers, JSON support
- Runs on various platforms (Windows, Linux, macOS, Unix)
- Powerful for diverse applications

Reference Architecture for IoT-Enabled Smart Buildings

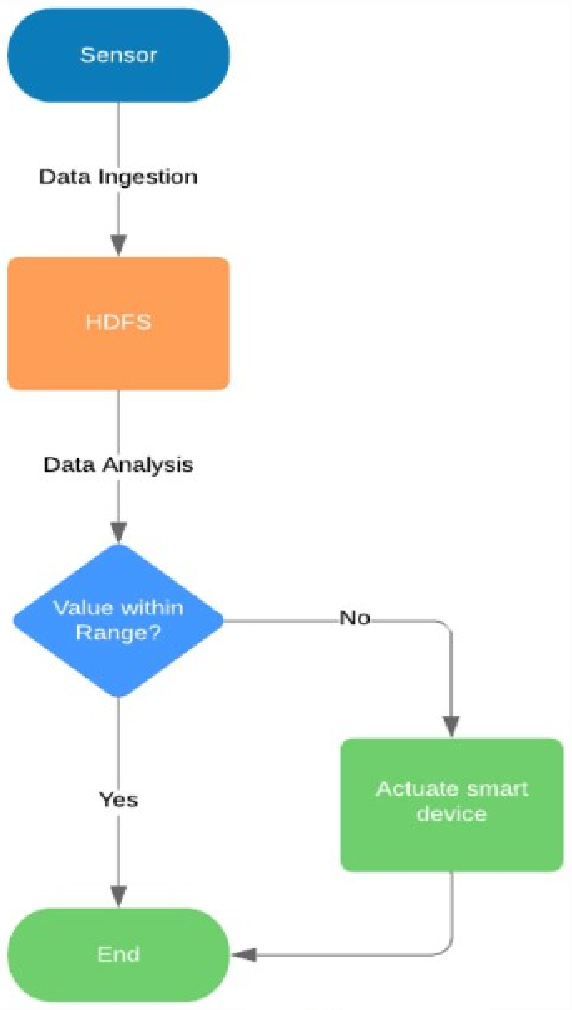
- Data Source: IoT sensors (oxygen sensors, smoke detection sensors, light sensors) generate data.
- Data Ingestion/Storage: sensors send the data to 2 sinks:
    1. Apache Flume[1] ingests data into HDFS over TCP.
    2. Elasticsearch indexes the sensor streaming data to be visualized
- Data Analysis and decision-making: Apache Spark algorithm using PySpark for near real-time analysis.

1. While MQTT is popular for IoT data, Flume chosen for direct data ingestion into HDFS.

SmartWins

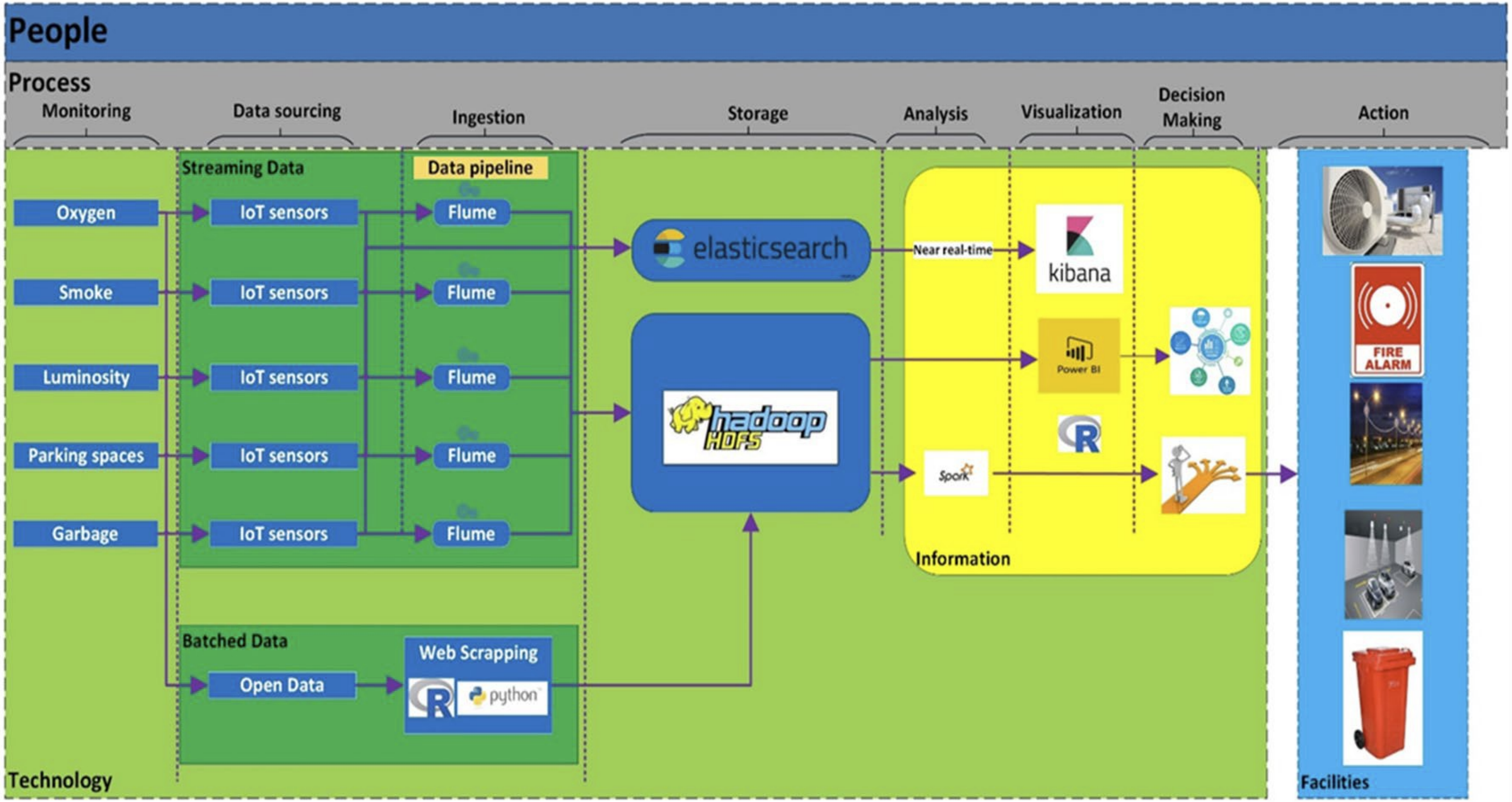
Reference Architecture for IoT-Enabled Smart Buildings

- Smart Building Control: PySpark Algorithm processes data from various IoT sensors→ enable decision-making for the effective management of a smart building services/devices (e.g. HVAC, lights, and alarms)
- Various messages were printed on the terminal screen simulating the feedback actuation behavior.
  - Example 1: Low luminosity activates lights in specific locations
  - Example 2: Low oxygen concentration triggers HVAC System.
- Visualization of data (stored in HDFS):near-real data visualizations in Kibana, Power BI dashboards for the IoT data visualization



SmartWins

Reference Architecture for IoT-Enabled Smart Buildings



Project Partners

kaunas university of technology (ktu 1922)

CERTH – CENTRE FOR RESEARCH & TECHNOLOGY HELLAS

POLITECNICO MILANO 1863

Contecht

INNOTROPE

Funded by the European Union

This project has received funding from the European Union's Horizon research and innovation programme under grant agreement No 101078997

SmartWins