Webinar

Scalable Industrial Data Management

A Webinar Every Smart Manufacturer Must Attend

Presented by HIVEMQ

Speaker

Kudzai Manditereza

Developer Advocate at HiveMQ

kudzai.manditereza@hivemq.com

linkedin.com/in/kudzaimanditereza

@techbykudzi

Kudzai Manditereza is a Developer Advocate at HiveMQ and the Founder of Industry40.tv. He is the host of an IIoT Podcast and is involved in Industry4.0 research and educational efforts.

Why Does A Data Management Strategy Matter?

The Power of Effective Data Management for Smart Manufacturing

KEY TO SUCCESS

Aligning Data Management With Business Objective

Identifying, Acquiring and Integrating Plant-Floor Data for Smart Manufacturing

Computer Integrated Manufacturing (CIM) Pyramid

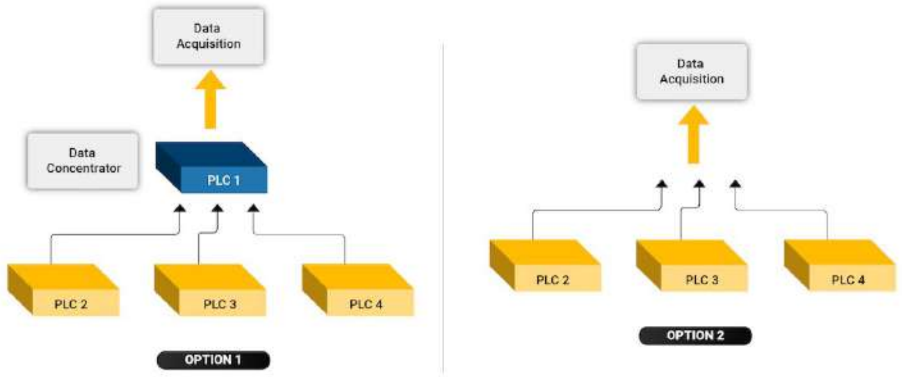
Potential Sources for Data Acquisition

01 PLCs at level 2

02 SCADA Systems at level 3

03 Historians at level 3

Integrating Data from Programmable Logic Controllers (PLCs)



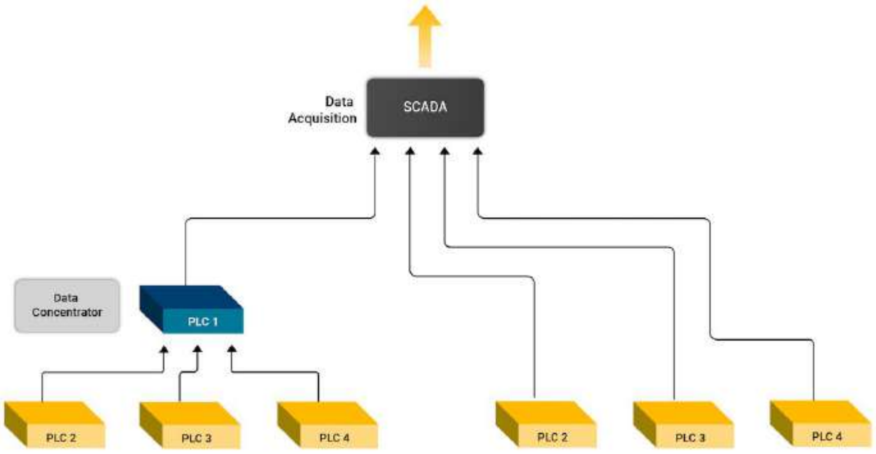
Pros

- Rapid scanning capabilities
- Reliable, stable and deterministic
- Minimal downtime guaranteed

Cons

- Low levels of data abstraction
- Inconsistent naming conventions across plants/areas.

Integrating Data from Supervisory Control and Data Acquisition (SCADA)



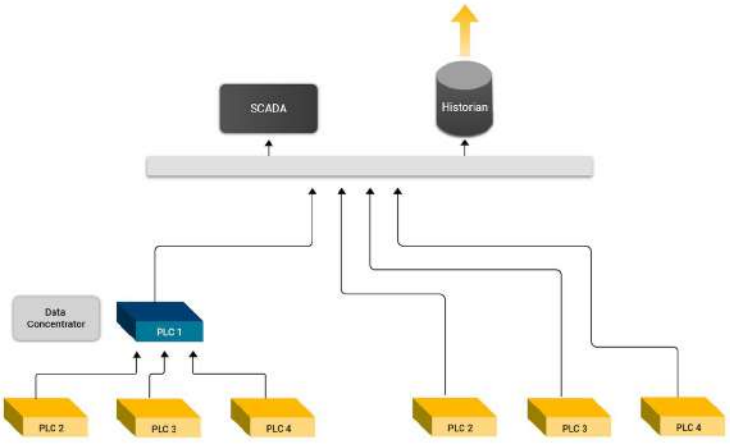
Pros

- Built for data acquisition, robust data concentrator.
- Common data model
- Already integrated into the plant or corporate network

Cons

- Less reliable, regular updates
- Lower data resolution
- Direct connection to a SCADA via SDK or API

Integrating Data from Historians



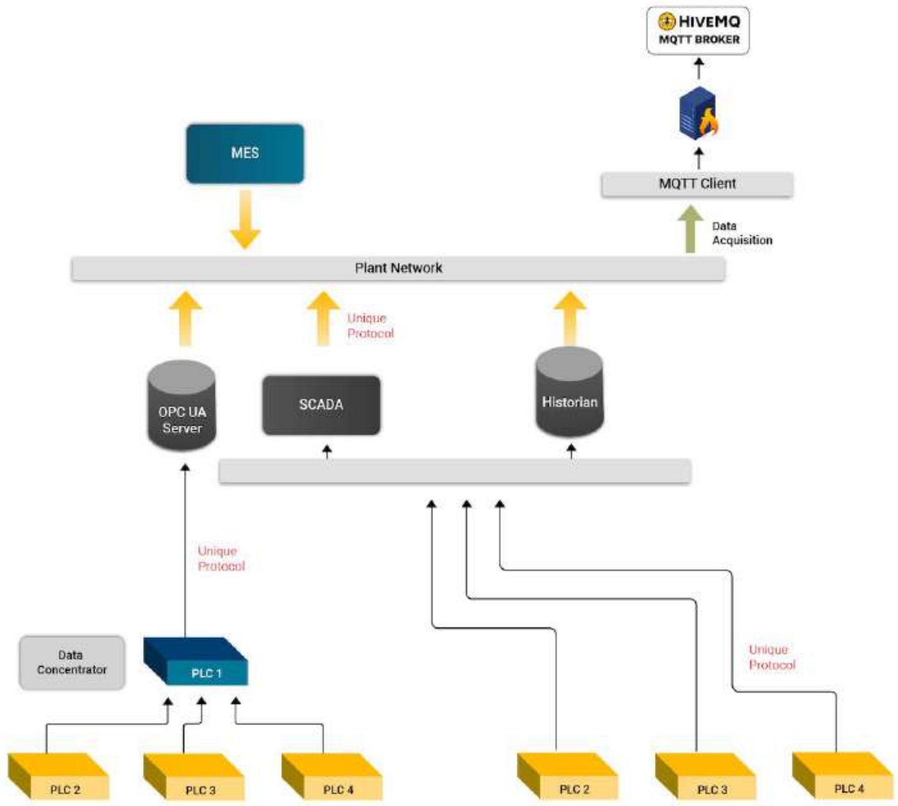
Pros

- Uses hierarchical asset model for arranging data.
- Data buffering and store-and-forward capabilities
- Already integrated into the plant or corporate network

Cons

- Might only gather a portion of the data.
- Raw data that does not deliver a time-specific snapshot of an asset
- Direct connection to a Historian via SDK or API

Integrating Plant-Floor Data to The Enterprise



Scalable

MQTT's publish/subscribe model is scalable, especially when dealing with many devices.

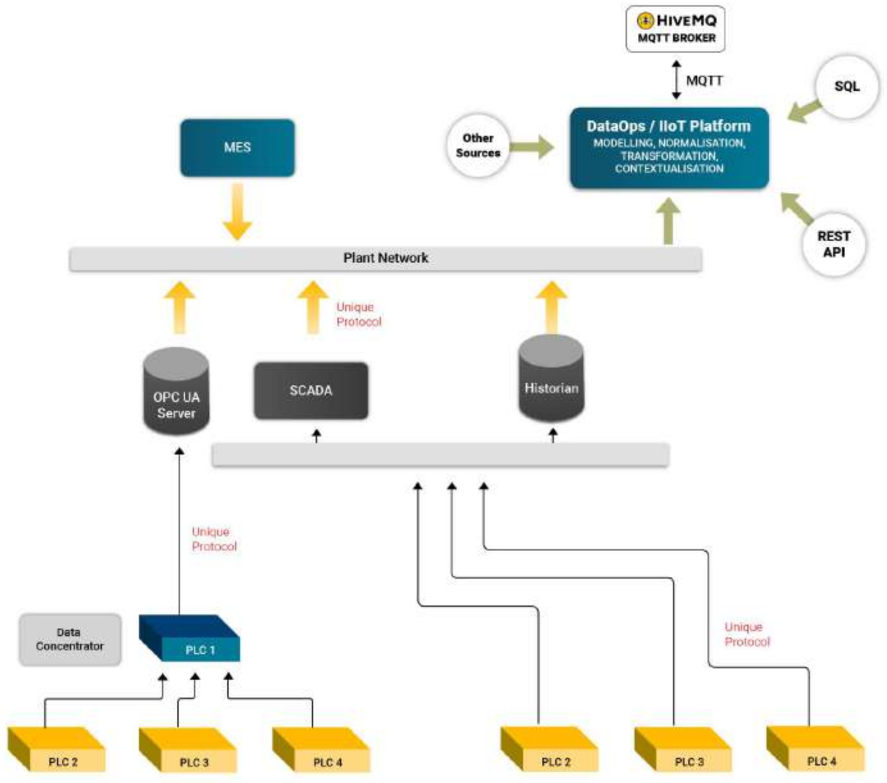
Cloud Connectivity

MQTT is a popular choice for cloud integration due to its native support on many IoT platforms.

Real-time Processing

MQTT is suitable for real-time data processing thanks to its lightweight and real-time capabilities.

DataOps for Industrial IoT Data Management



- **Data Modelling**
- **Data Normalisation**
- **Data Transformation**
- **Data Contextualisation**

Data Modelling for IIoT

Machine ID (unique identifier for the machine)

Timestamp (time when the reading was taken)

Temperature (current temperature of the machine)

Vibration (current vibration level of the machine)

Output Speed (current speed of the output from the machine)

A simple data model for a machine

Define the structure, relationships, and characteristics of the data

Data Normalization for IIoT

Machine ID: 001

Timestamp: 2023-05-12T14:00:00Z (UTC time)

Temperature: 75 (degrees Celsius)

Vibration: 3.2 (mm/s)

Output Speed: 50 (units per hour)

Machine data after normalisation

Streamline the data, reducing redundancy, and enhancing data integrity

Data Transformation for IIoT

Machine ID: 001

Shift: Morning (2023-05-12T06:00:00Z to 2023-05-12T14:00:00Z)

Average Output Speed: 48 (units per hour)

Machine data after transformation

Convert raw data into a more suitable format for analysis or further processing

Data Contextualisation for IIoT

Machine ID: 001

Shift: Morning (2023-05-12T06:00:00Z to 2023-05-12T14:00:00Z)

Average Output Speed: 48 (units per hour)

Product: Widget A

Maintenance Activity: None

Environmental Anomalies: Power surge reported at 2023-05-12T13:00:00Z

Machine data after contextualisation

Enhance the interpretability of the data, facilitating more accurate and effective decision-making.

Data Modelling Standards for Smart Manufacturing

Digital Twin Definition Language

- Creating models for Digital Twin use cases
- Capture relationships between different devices, their properties, commands, and telemetry.
- Uses JSON-LD, DTDL to provide a common structure and semantics for digital twin models


```
{
  "@context": {
     "@vocab": "dtmi:com:example:Machine;1",
     "lastMaintenance": {
        "@id": "dtmi:com:example:Machine;1#lastMaintenance",
        "@type": "xsd:date"
     },
     "output": {
         "@id": "dtmi:com:example:Machine;1#output",
         "@type": "xsd:integer"
     }
},
"id": "Machine123",
"type": "CNCMachine",
"status": "Active",
"lastMaintenance": "2022-12-01",
"output": 2000
}
```


JSON-LD representation using DTDL for a CNC Machine

MQTT Sparkplug

- Interoperable SCADA/IIoT solutions, leveraging MQTT as the core messaging technology
- Consistent definition of the MQTT Topic Namespace, session state management and payload definition.
- Primitive data types, datasets, complex data types via templates.


```
{
    "TemperatureSensor": {
    "device_id": "string",
    "metrics": {
      "current_temperature": {
        "type": "float",
        "unit": "°C",
        "description": "Current temperature reading"
      },
      "max_temperature": {
        "type": "float",
        "unit": "°C",
        "description": "Maximum temperature recorded"
      },
      "min_temperature": {
        "type": "float",
        "unit": "°C",
        "description": "Minimum temperature recorded"
      },
      "status": {
        "type": "boolean",
        "description": "Operational status of the sensor"
      }
    }
  }
}
```


Sparkplug Template schema in JSON for the temperature sensor

OPC UA Companion Specifications

- Unified approach for interacting with different data sources in an industrial setting.
- Guidelines that define how a particular industry, vendor, or group of products should use OPC UA.
- Developed by industry-specific working groups within the OPC Foundation or by external organizations

```xml
<opc:UANodeSet
xmlns:opc="http://opcfoundation.org/UA/2011/03/UANodeSet.xsd">
    <opc:Aliases>
        <!-- Aliases for data types, like Boolean, String, etc. -->
    </opc:Aliases>
    <opc:UAObject NodeId="ns=1;s=MachineTool" BrowseName="1:MachineTool">
        <!-- Object representing a machine tool -->
        <opc:UAVariable NodeId="ns=1;s=MachineTool.Status"
BrowseName="1:Status">
        <!-- Variable representing the status of the machine tool -->
    </opc:UAVariable>
    <opc:UAObject NodeId="ns=1;s=MachineTool.Controller"
BrowseName="1:Controller">
        <!-- Object representing the controller of the machine tool -->
    <opc:UAVariable NodeId="ns=1;s=MachineTool.Controller.Program"
BrowseName="1:Program">
        <!-- Variable representing the program running on the controller -->
    </opc:UAVariable>
    <!-- More variables or objects representing other aspects of the controller -->
    </opc:UAObject>
    <!-- More components or other aspects of the machine tool -->
    </opc:UAObject>
</opc:UANodeSet>
```

XML structure defined by the MTConnect OPC UA Companion Specification

Asset Administration Shell (AAS)

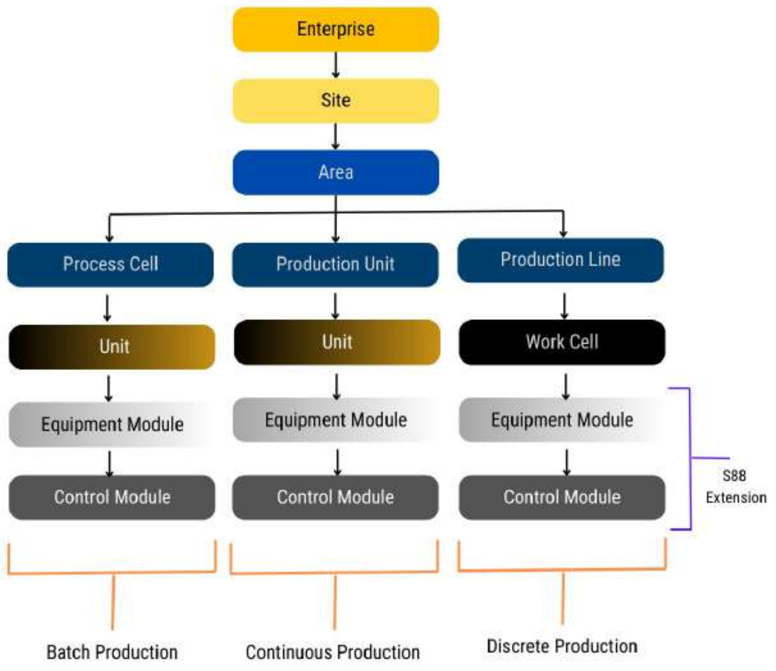
- Encapsulates all information about an asset, including technical features, functionalities, and lifecycle data
- API for exposing the properties and capabilities of a physical asset to facilitate its administration
- Often represented in JSON or XML format


```json
{
    "id": "conveyor123",
    "type": "Conveyor System",
    "manufacturer": "Conveyor Co.",
    "dateOfManufacture": "2022-01-01",
    "configuration": {
        "capacity": 500,
        "speed": 100,
        "powerRequirements": "220V AC"
    },
    "conditionMonitoring": {
        "currentSpeed": 75,
        "currentLoad": 250,
        "temperature": 30,
        "status": "Running"
    },
    "events": [
    {
    "timeStamp": "2023-05-14T09:00:00",
    "eventType": "Maintenance",
    "description": "Routine maintenance performed"
    },
    {
    "timeStamp": "2023-05-14T10:00:00",
    "eventType": "Error",
    "description": "Overload error",
    "resolved": true
    }
  ],
    "specificServices": {
    "adjustSpeed": "http://example.com/api/conveyor123/adjustSpeed",
    "initiateMaintenance": "http://example.com/api/conveyor123/initiateMaintenance"
    }
}
```


Conveyor belt system example of AAS represented in JSON format

Semantic Data Structuring with MQTT Sparkplug and Unified Namespace

Create your information hierarchy using ISA 95 Part 2



- Shared framework for managing production processes and resources
- Globally recognized standard for merging enterprise and control systems
- Comprises both models and a standard vocabulary
- Presents a hierarchical structure of equipment entities, each having distinct organizational functions and relationships

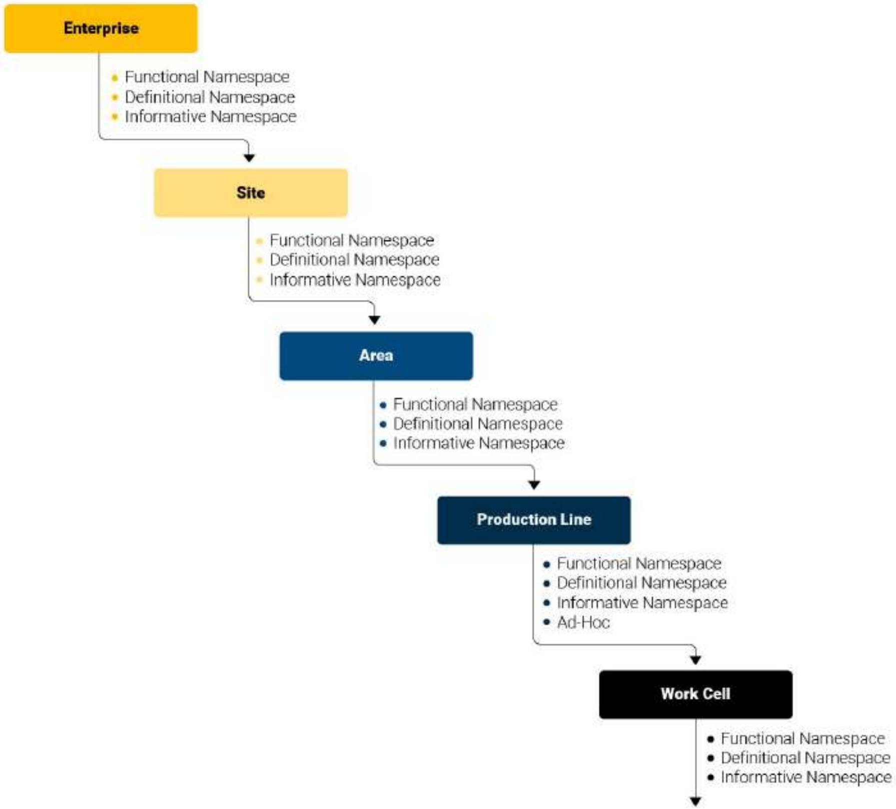
Key Steps to Designing Your UNS Data Architecture

01 Identifying existing namespaces within your (OT) environment

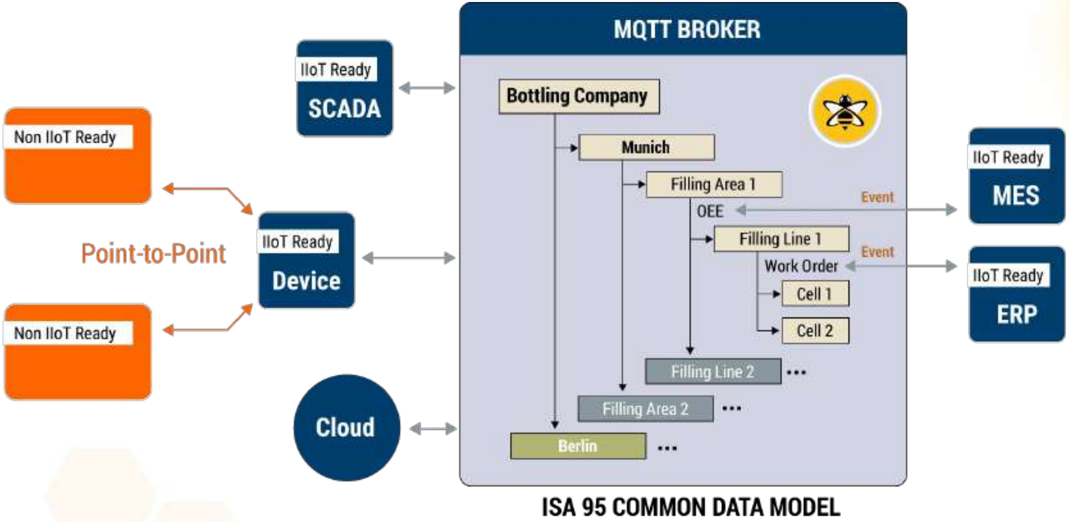
02 Collect and contextualise data from the Namespaces.

03 Plug the namespaces into the suitable level of your hierarchy

Typical Namespaces for UNS



Typical Namespaces for UNS



spBv1.0/group_id/message_type/edge_node_id/[device_id]

spBv1.0/BottlingCompany:Munich/DDATA/FillingArea1:FillingLine1/Cell1

IIoT Data Storage and Actionable Analytics Generation

Time-Series Databases for IIoT

- Handle time-stamped data, which is data collected over time.
- Crucial for storing, retrieving, and processing this data efficiently
- Useful for monitoring machine health and predicting failures

Measurement: "Temperature_Sensor_1"

Tags:

"machine_ID": "Machine_7"

"location": "Factory_A"

"operator": "Operator_John"

Fields:

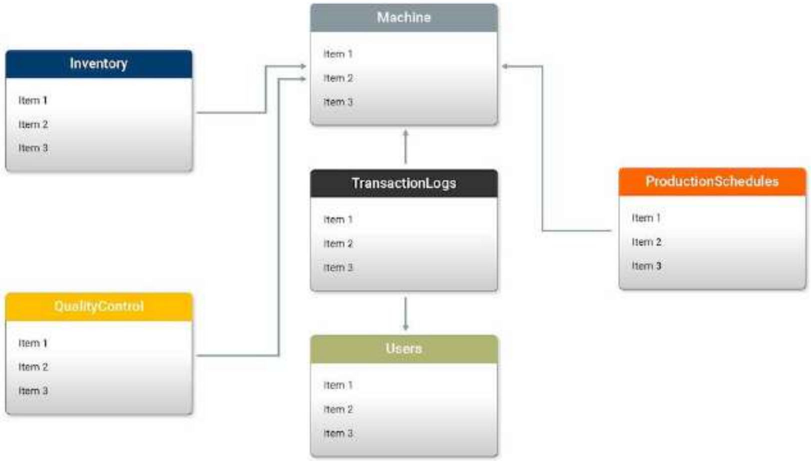
"temperature": 72.4

Time: 1676532340000000000

Example of InfluxDB Time-series data model for a machine

Structured Databases for IIoT

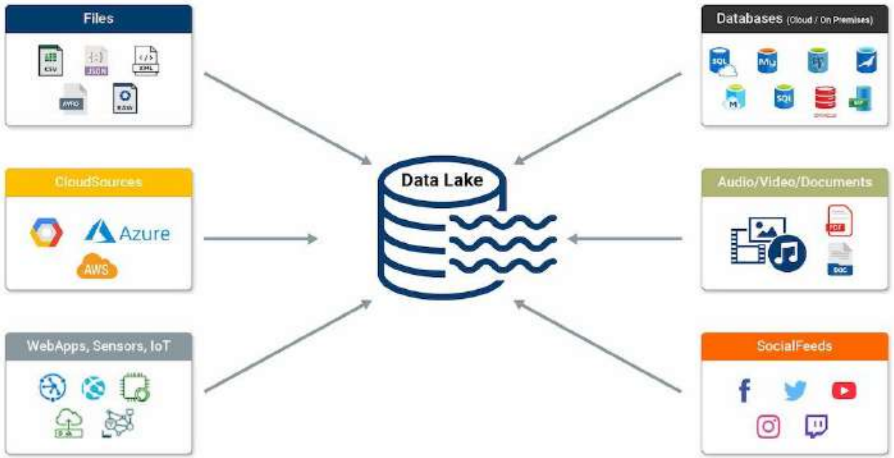
- Store data in predefined structures or tables
- Essential for managing structured data like device metadata, user information, and transaction logs.
- Provide robust data integrity and support complex queries



Simplified view of database tables in a relational database

Data Lakes for IIoT

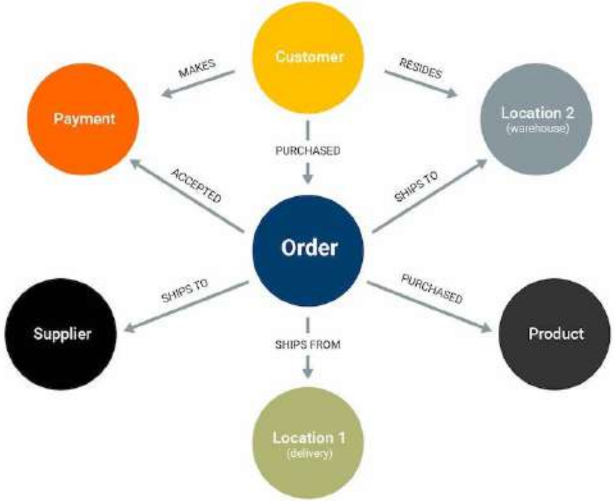
- Hold raw data in its native format until it's needed.
- Offer flexibility, scalability, and the ability to handle structured and unstructured data
- Schema-on-read data model is applied only when reading the data.



Data Lake ingesting data

Graph Databases for IIoT

- Utilizes graph theory to store, map, and query relationships within data
- Superior capability to handle complex, interconnected data
- Graph data model symbolizes data as interconnected nodes and edges.



Graph Database Representation

Unified view of your smart manufacturing analytics

01 cross-contextualisation

02 correlation.

03 Single pane of glass

ANY QUESTIONS?

Resources

[Get Started with MQTT]

[Unified Namespace (UNS) Essentials]

[A Comprehensive Guide To Industrial Data Management for Smart Manufacturing]

HIVEMQ [Evaluate HiveMQ]

HIVEMQ CLOUD [Try HiveMQ Cloud]

THANK YOU

Contact Details

Kudzai Manditereza: kudzai.manditereza@hivemq.com | linkedin.com/in/kudzaimanditereza/

Sign-up Free or Evaluate the Full Version of HiveMQ