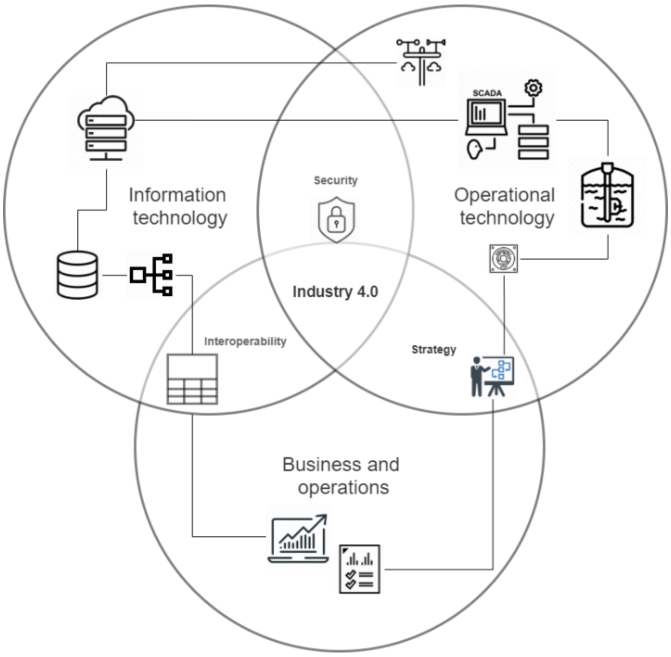
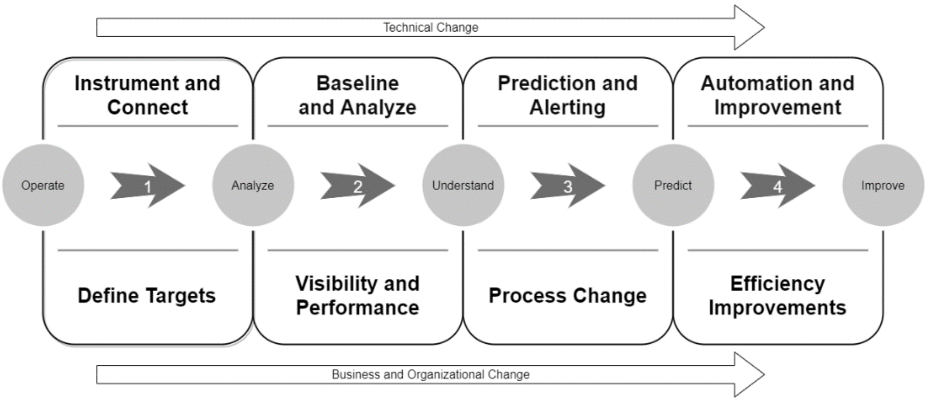
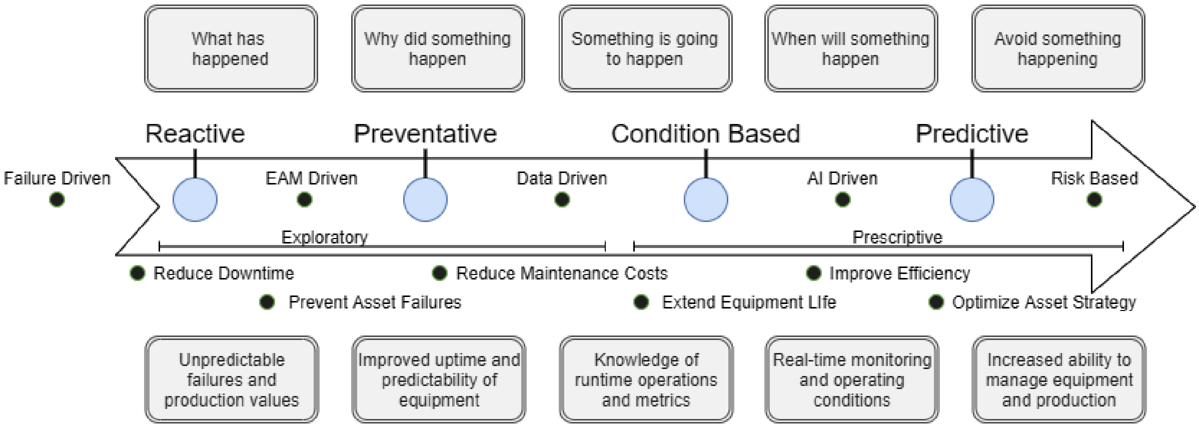
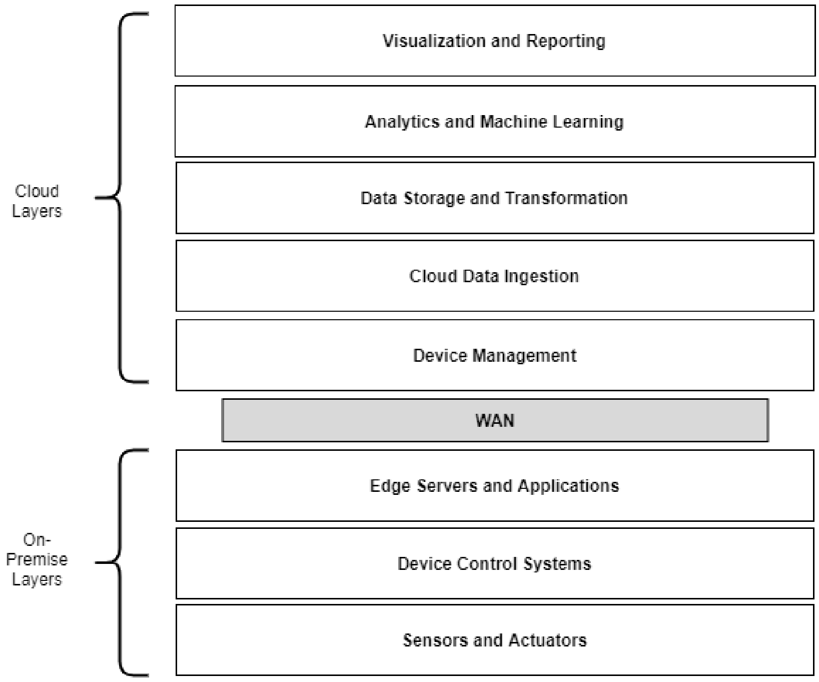
Chapter 1: Welcome to the IoT Revolution

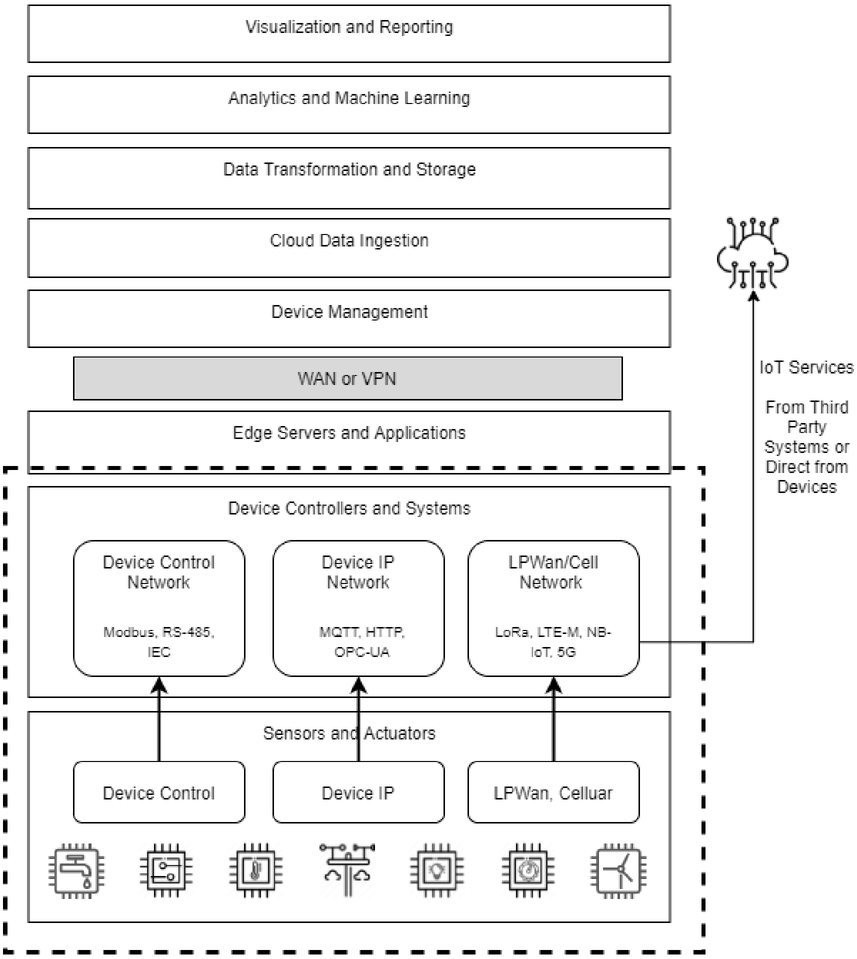


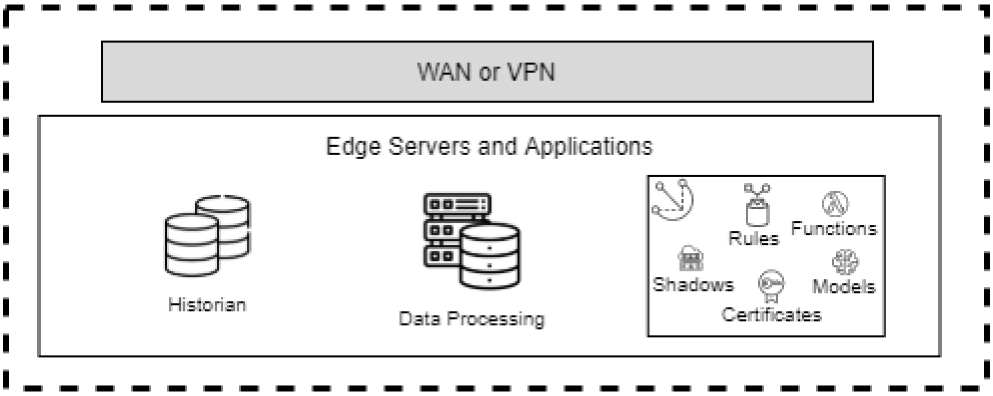


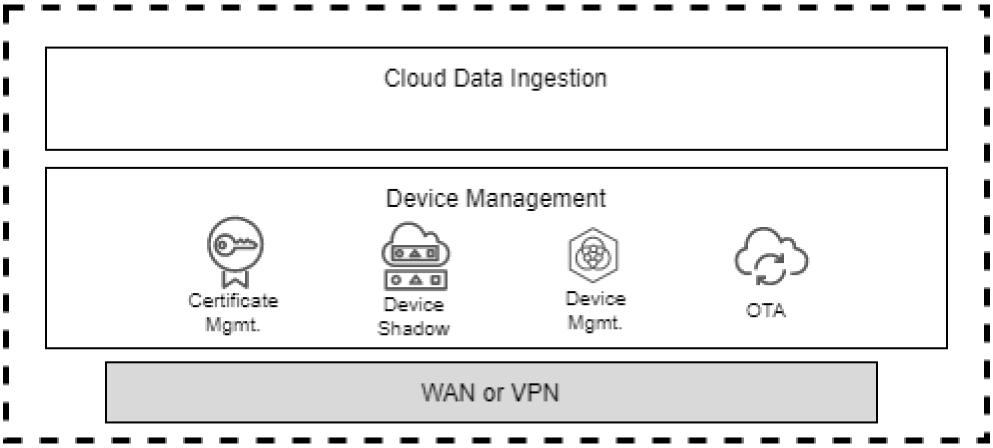

What has happened
Why did something happen
Something is going to happen
When will something happen
Avoid something happening
Reactive
Preventative
Condition Based
Predictive
Failure Driven
EAM Driven
Data Driven
AI Driven
Risk Based
Exploratory
Prescriptive
Reduce Downtime
Reduce Maintenance Costs
Improve Efficiency
Prevent Asset Failures
Extend Equipment LIfe
Optimize Asset Strategy
Unpredictable failures and production values
Improved uptime and predictability of equipment
Knowledge of runtime operations and metrics
Real-time monitoring and operating conditions
Increased ability to manage equipment and production

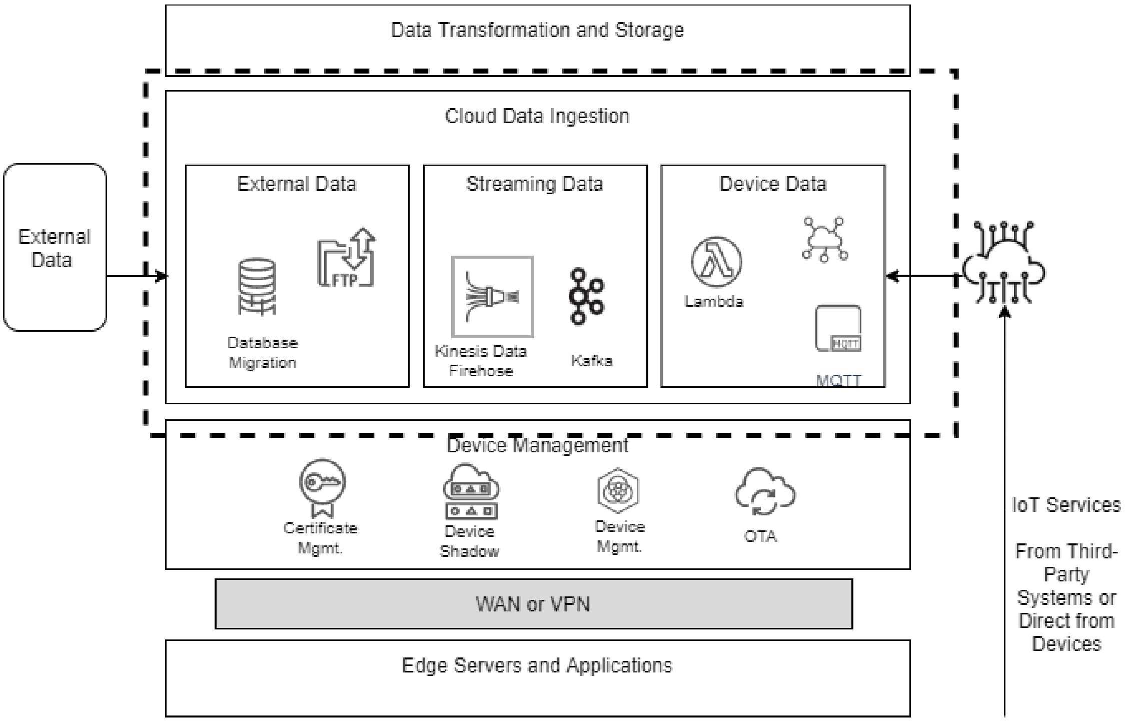
Chapter 2: Anatomy of an IoT Architecture

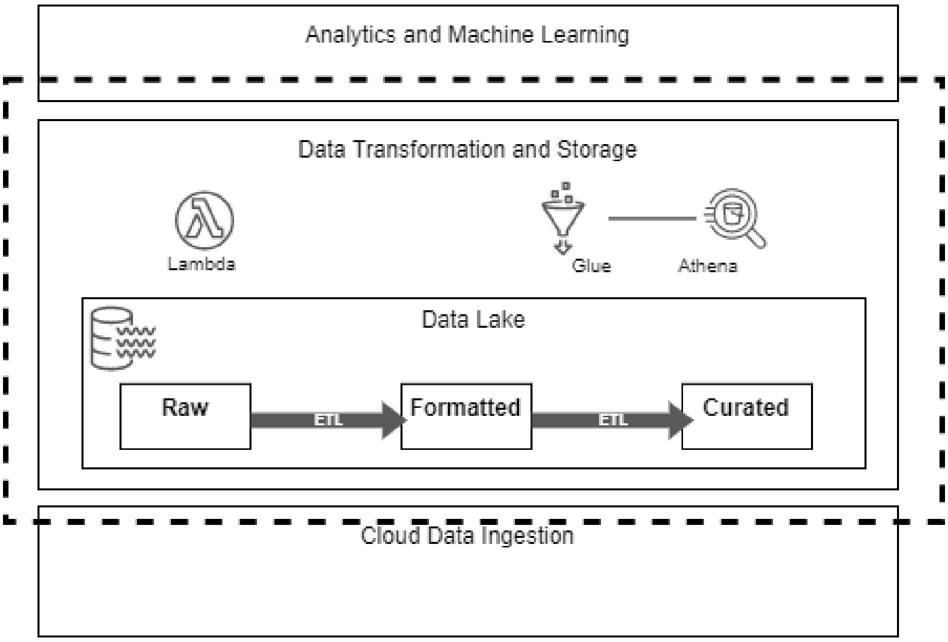

Cloud Layers
Visualization and Reporting
Analytics and Machine Learning
Data Storage and Transformation
Cloud Data Ingestion
Device Management
WAN
On-Premise Layers
Edge Servers and Applications
Device Control Systems
Sensors and Actuators

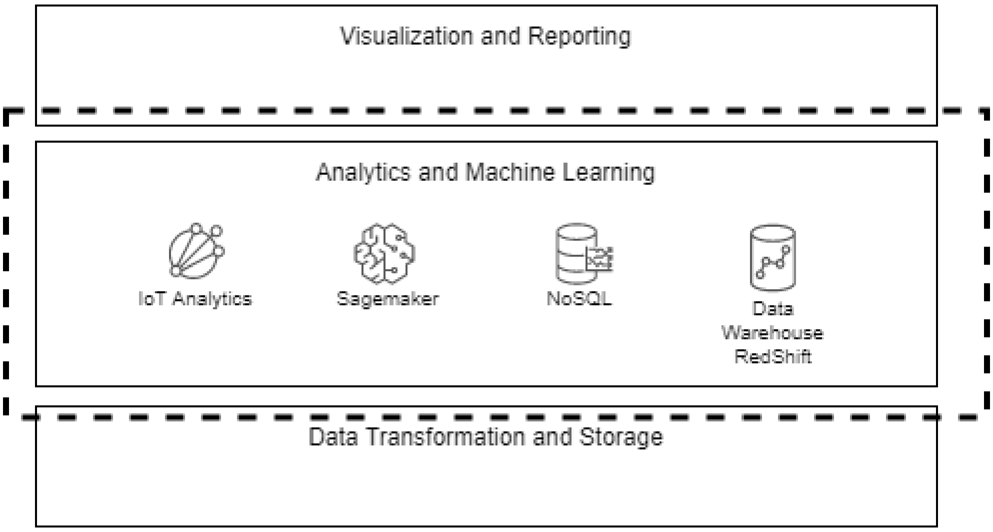


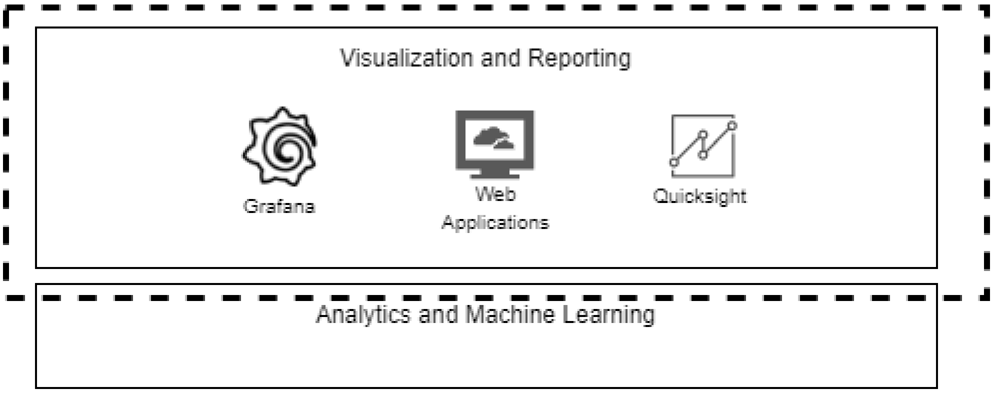

WAN or VPN
Edge Servers and Applications
Historian
Data Processing
Rules
Functions
Shadows
Models
Certificates

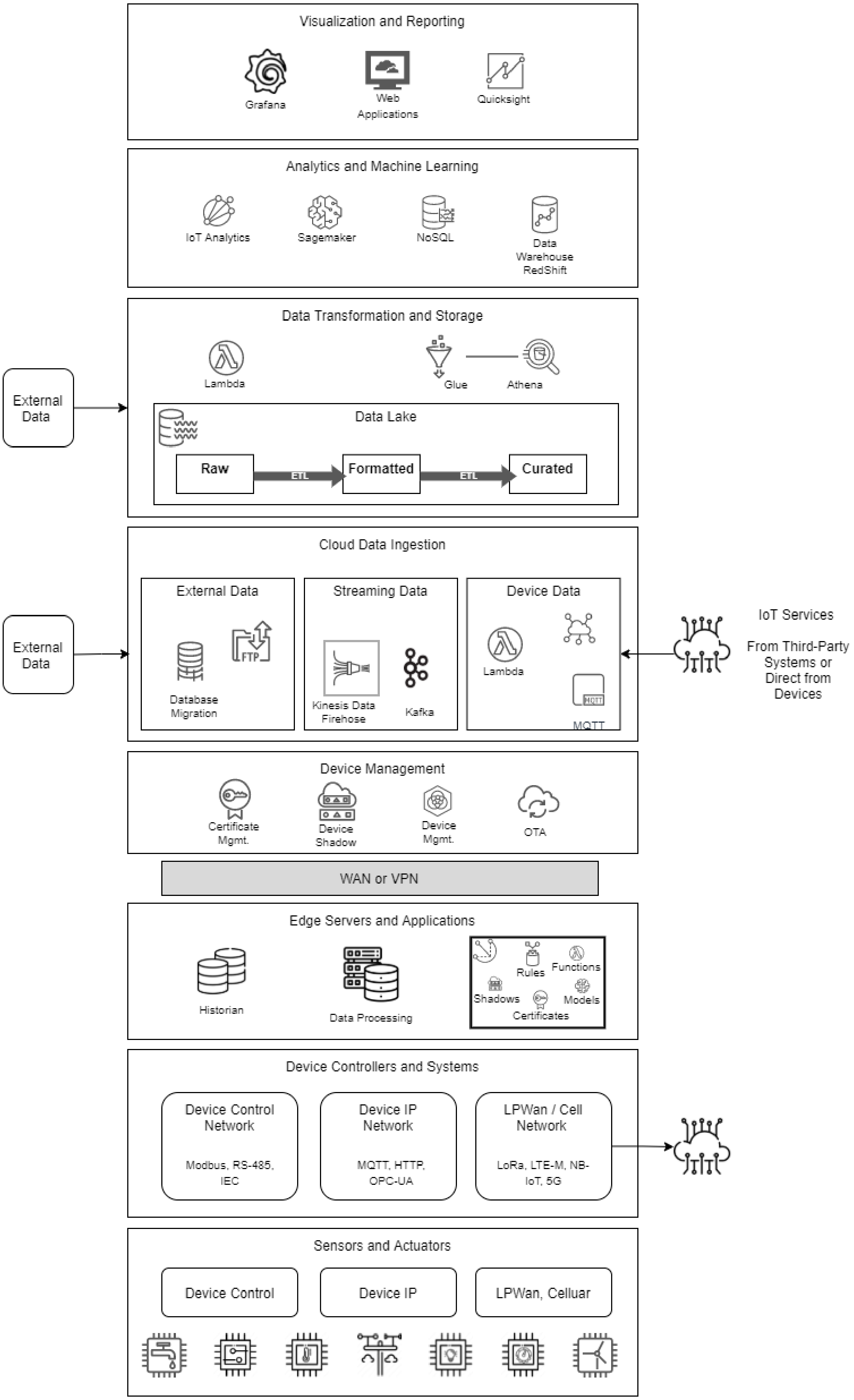

Cloud Data Ingestion
Device Management
Certificate Mgmt.
Device Shadow
Device Mgmt.
OTA
WAN or VPN


Data Transformation and Storage
Cloud Data Ingestion
External Data
Streaming Data
Device Data
External Data
Database Migration
FTP
Kinesis Data Firehose
Kafka
Lambda
MQTT
MQTT
Device Management
Certificate Mgmt.
Device Shadow
Device Mgmt.
OTA
WAN or VPN
Edge Servers and Applications
IoT Services
From Third-Party Systems or Direct from Devices

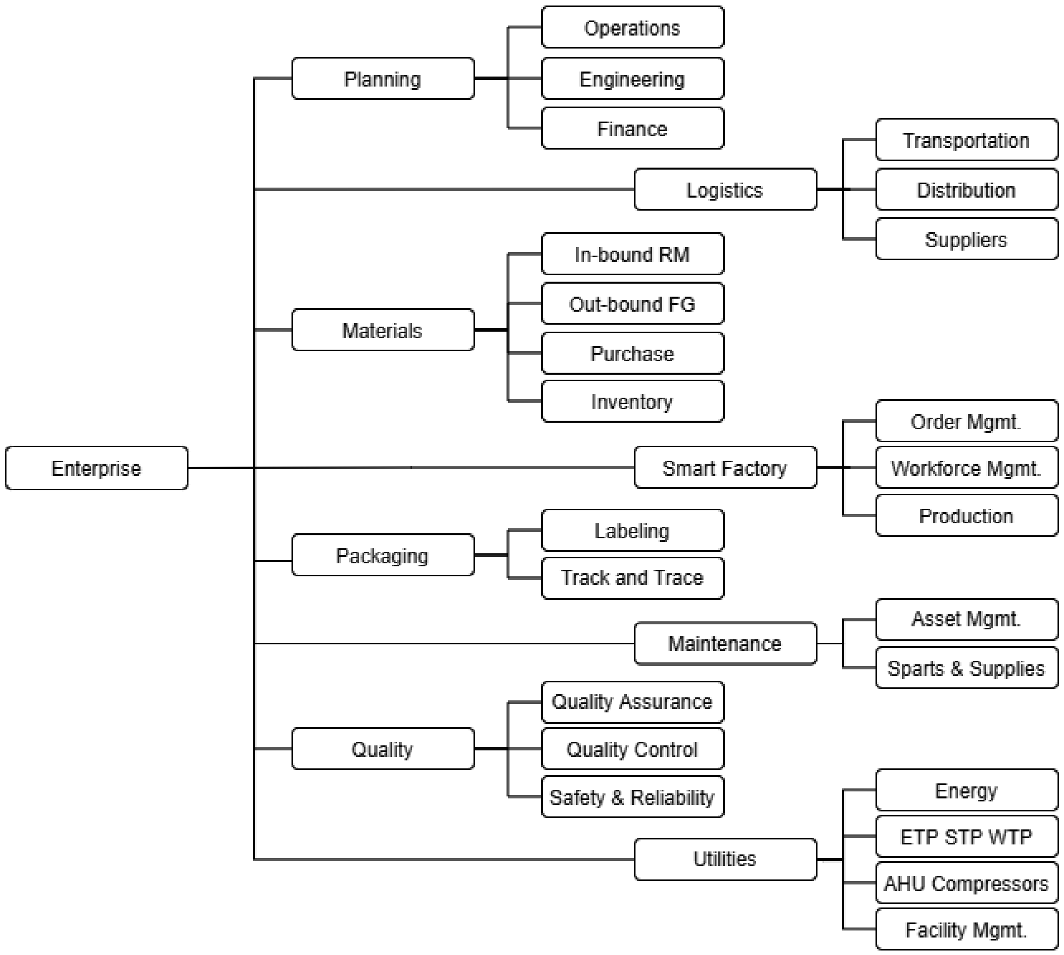

Analytics and Machine Learning
Data Transformation and Storage
Lambda
Glue
Athena
Data Lake
Raw
ETL
Formatted
ETL
Curated
Cloud Data Ingestion

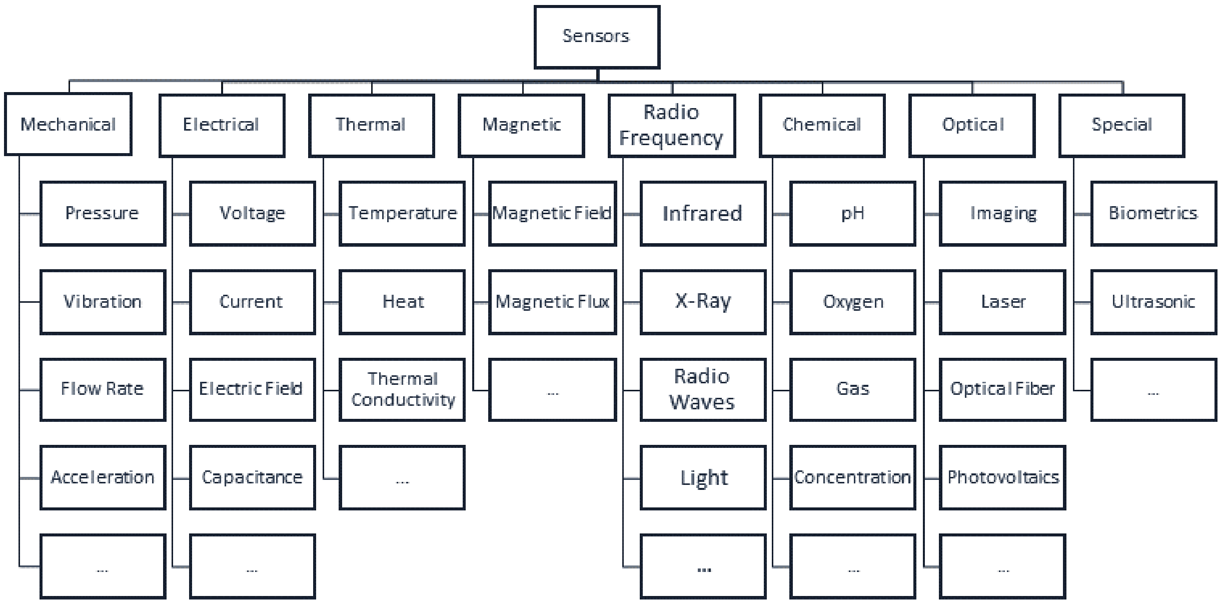

Visualization and Reporting
Analytics and Machine Learning
IoT Analytics
Sagemaker
NoSQL
Data Warehouse RedShift
Data Transformation and Storage

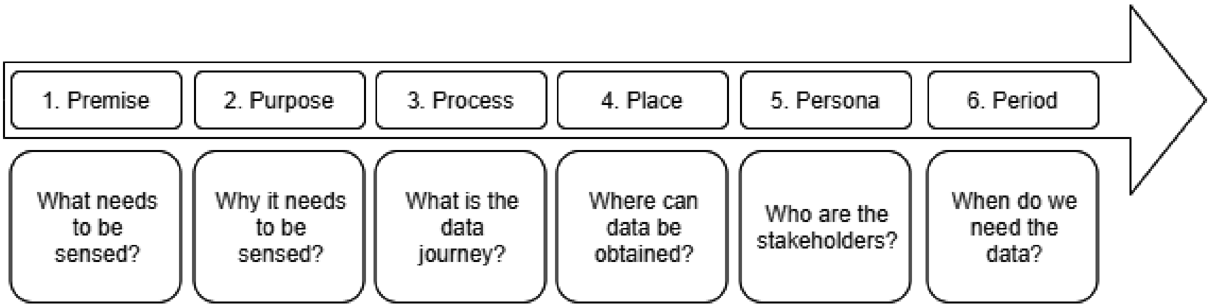

Visualization and Reporting
Grafana
Web Applications
Quicksight
Analytics and Machine Learning

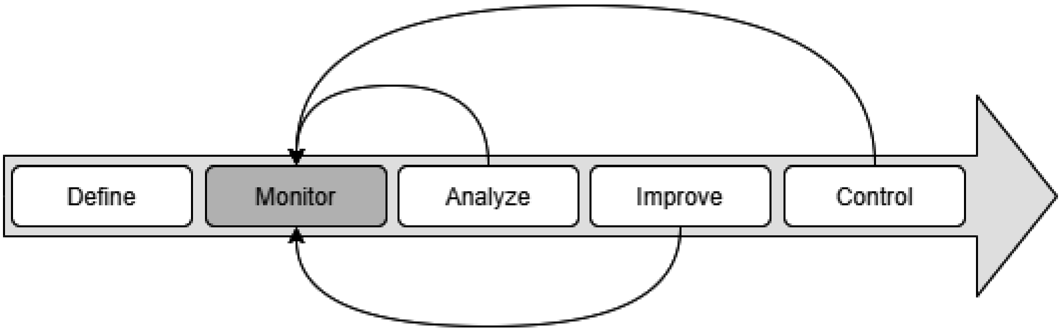

Visualization and Reporting
Grafana
Web Applications
Quicksight
Analytics and Machine Learning
IoT Analytics
Sagemaker
NoSQL
Data Warehouse RedShift
Data Transformation and Storage
Lambda
Glue
Athena
External Data
Data Lake
Raw
ETL
Formatted
ETL
Curated
Cloud Data Ingestion
External Data
External Data
FTP
Database Migration
Streaming Data
Kinesis Data Firehose
Kafka
Device Data
Lambda
MQTT
MQTT
IoT Services
From Third-Party Systems or Direct from Devices
Device Management
Certificate Mgmt.
Device Shadow
Device Mgmt.
OTA
WAN or VPN
Edge Servers and Applications
Historian
Data Processing
Rules
Functions
Shadows
Models
Certificates
Device Controllers and Systems
Device Control Network
Modbus, RS-485, IEC
Device IP Network
MQTT, HTTP, OPC-UA
LPWan / Cell Network
LoRa, LTE-M, NB-IoT, 5G
Sensors and Actuators
Device Control
Device IP
LPWan, Celluar

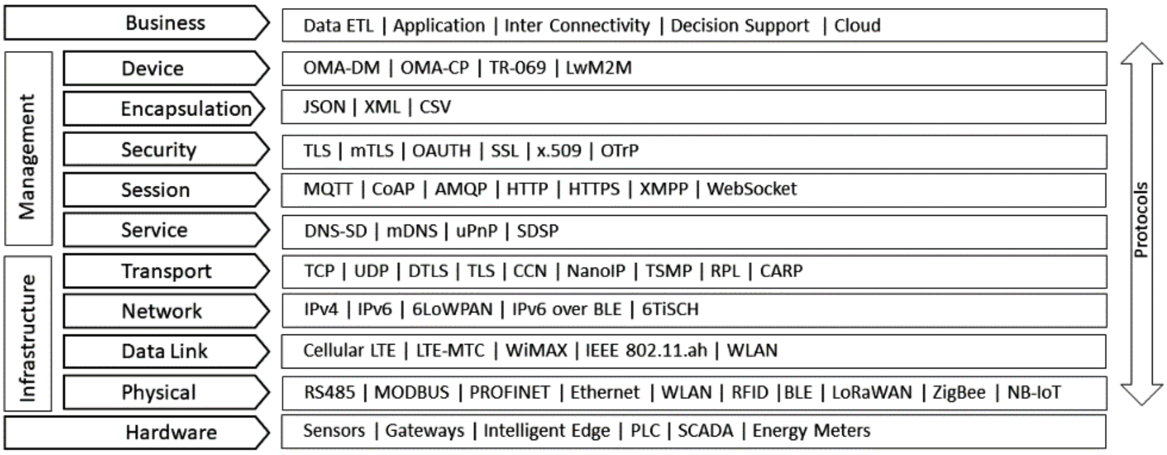
Chapter 3: In-Situ Environmental Monitoring



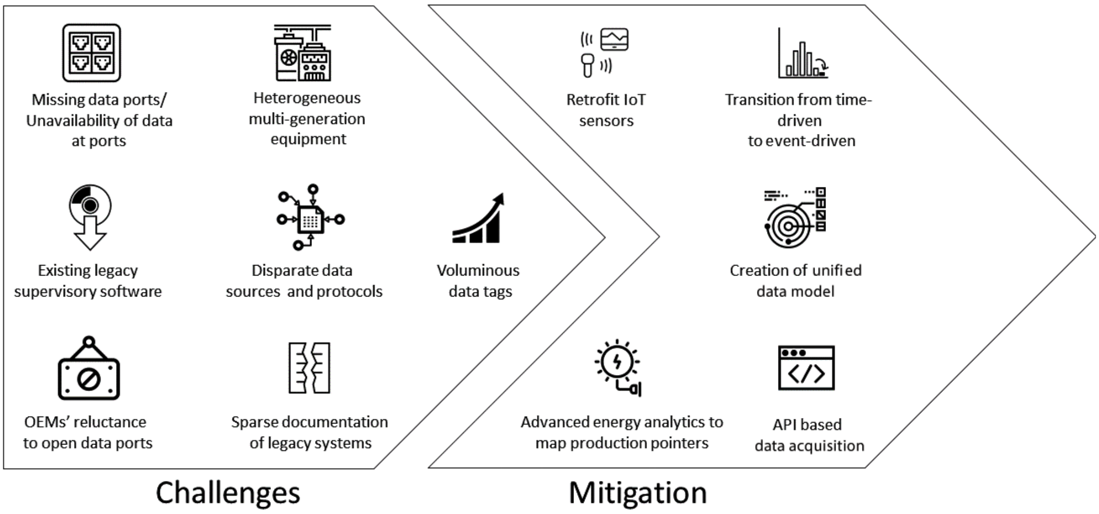


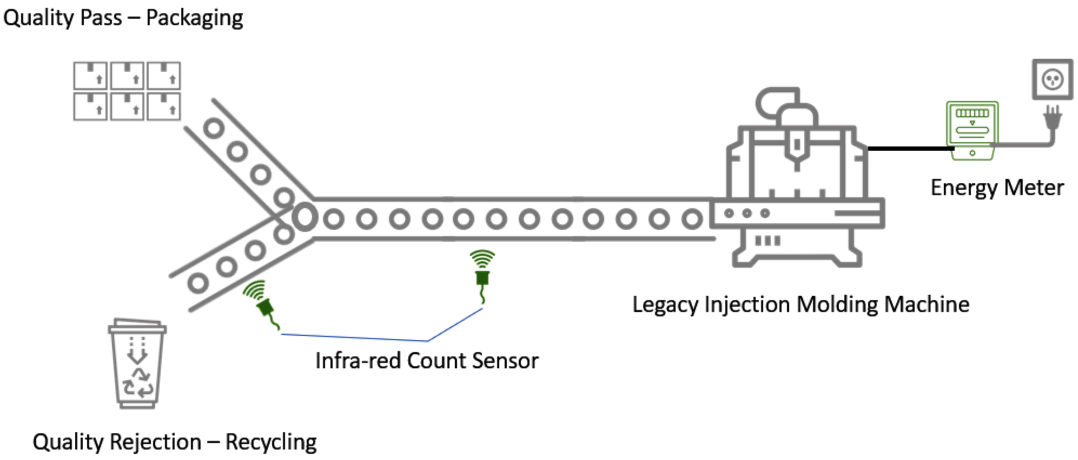

1. Premise
What needs to be sensed?
2. Purpose
Why it needs to be sensed?
3. Process
What is the data journey?
4. Place
Where can data be obtained?
5. Persona
Who are the stakeholders?
6. Period
When do we need the data?

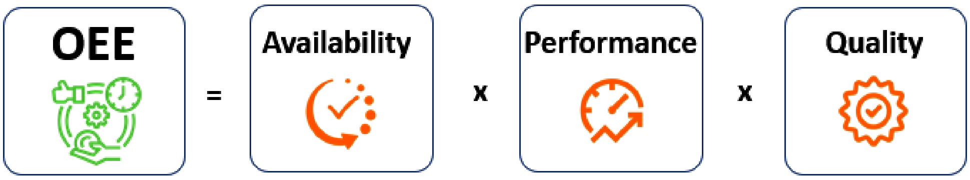




	Business	Data ETL | Application | Inter Connectivity | Decision Support | Cloud
Management	Device	OMA-DM | OMA-CP | TR-069 | LwM2M
	Encapsulation	JSON | XML | CSV
	Security	TLS | mTLS | OAUTH | SSL | x.509 | OTrP
	Session	MQTT | CoAP | AMQP | HTTP | HTTPS | XMPP | WebSocket
	Service	DNS-SD | mDNS | uPnP | SDSP
Infrastructure	Transport	TCP | UDP | DTLS | TLS | CCN | NanoIP | TSMP | RPL | CARP
	Network	IPv4 | IPv6 | 6LoWPAN | IPv6 over BLE | 6TiSCH
	Data Link	Cellular LTE | LTE-MTC | WiMAX | IEEE 802.11.ah | WLAN
	Physical	RS485 | MODBUS | PROFINET | Ethernet | WLAN | RFID |BLE | LoRaWAN | ZigBee | NB-IoT
	Hardware	Sensors | Gateways | Intelligent Edge | PLC | SCADA | Energy Meters


Missing data ports/ Unavailability of data at ports
Heterogeneous multi-generation equipment
Existing legacy supervisory software
Disparate data sources and protocols
Voluminous data tags
OEMs' reluctance to open data ports
Sparse documentation of legacy systems
Challenges
Retrofit IoT sensors
Transition from time-driven to event-driven
Creation of unified data model
Advanced energy analytics to map production pointers
API based data acquisition
Mitigation


Quality Pass – Packaging
Energy Meter
Legacy Injection Molding Machine
Infra-red Count Sensor
Quality Rejection – Recycling

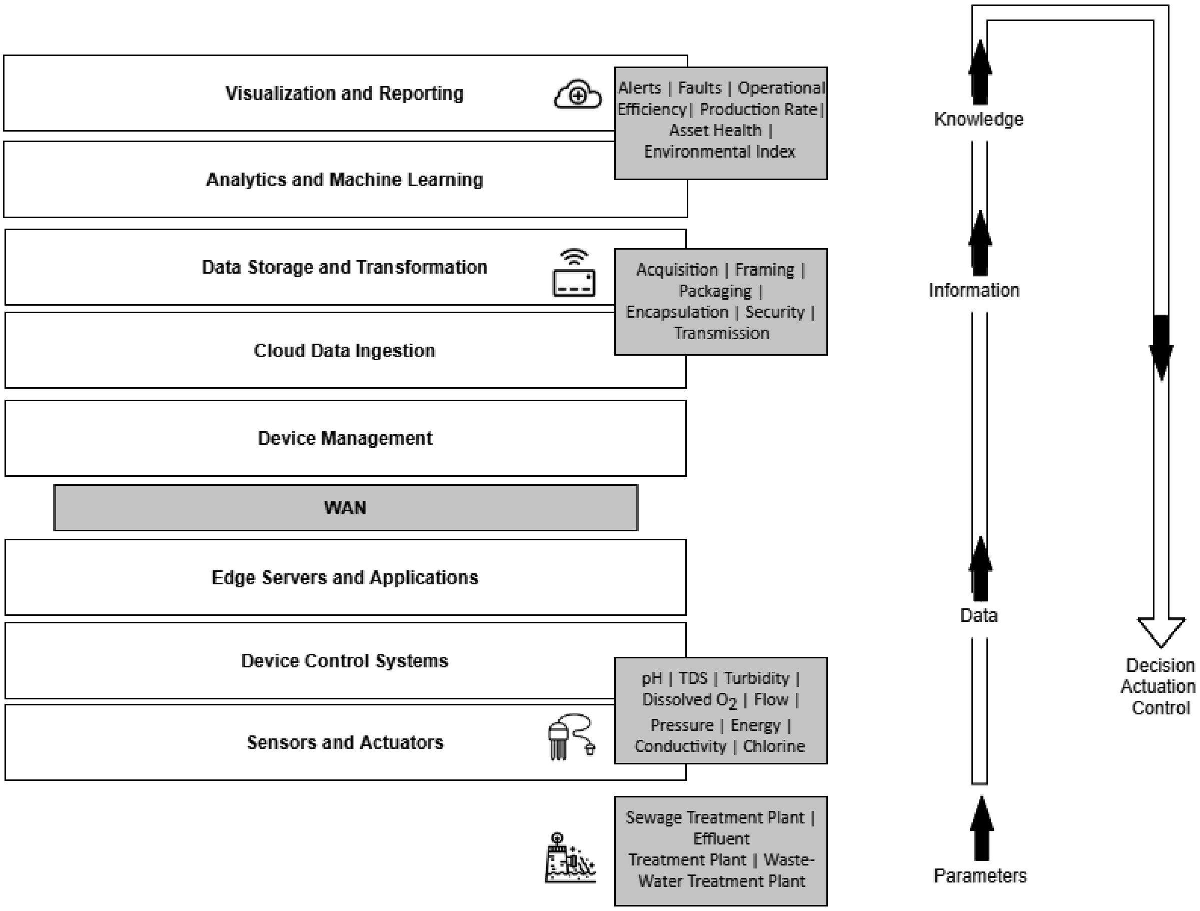


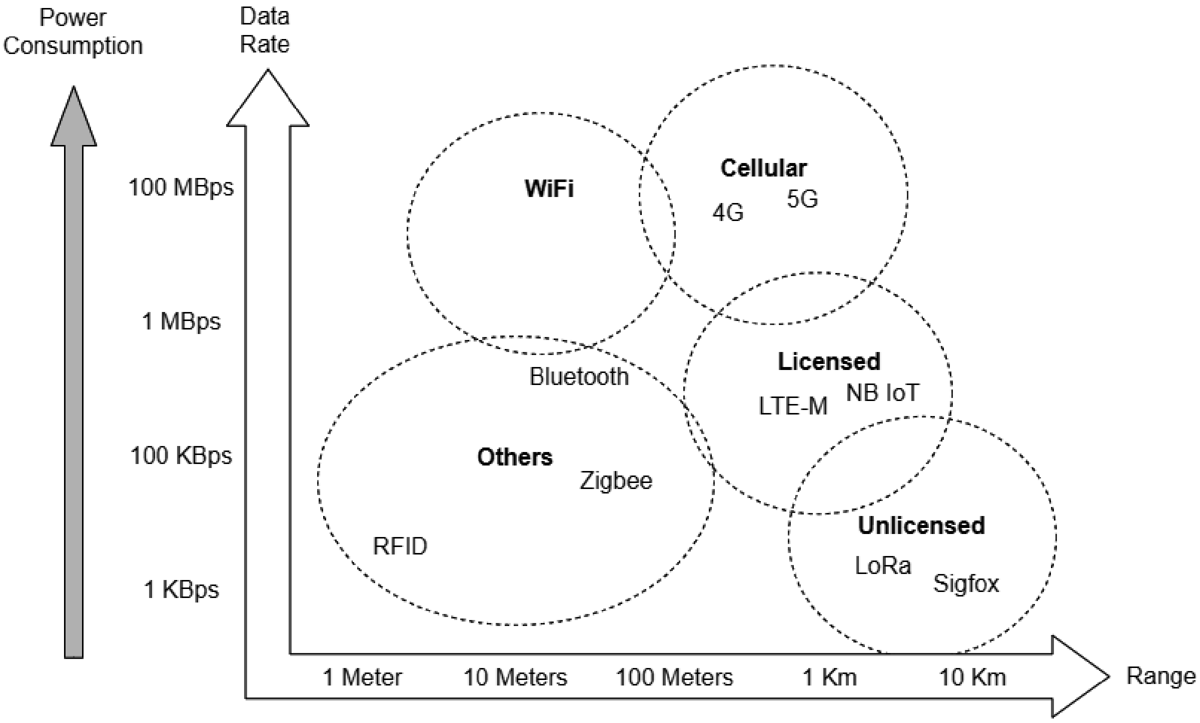
Overall **E**quipment **E**ffectiveness = **A**vailability x **P**erformance x **Q**uality

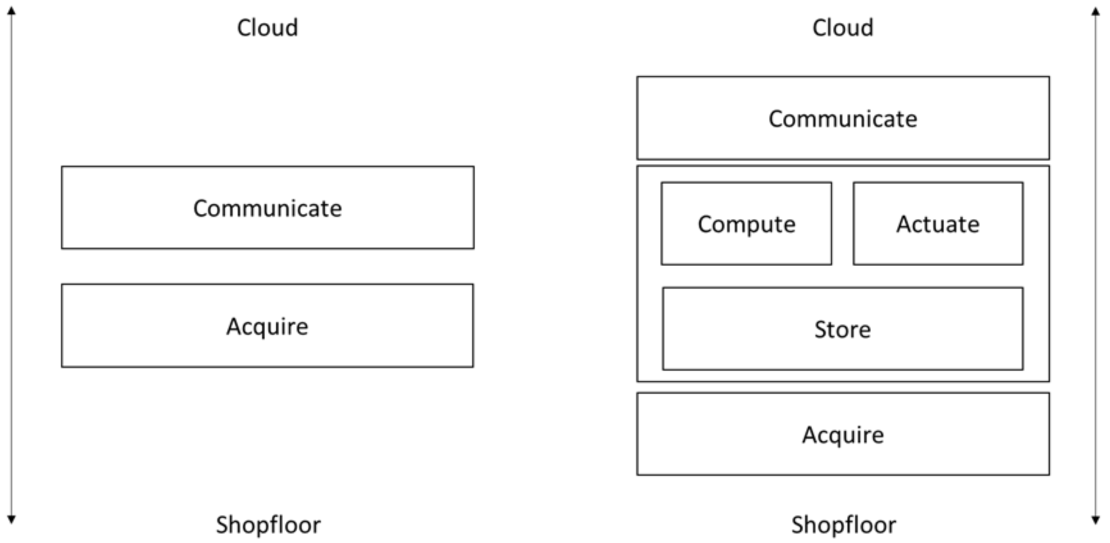
Availability = Run Time / Planned Production Time

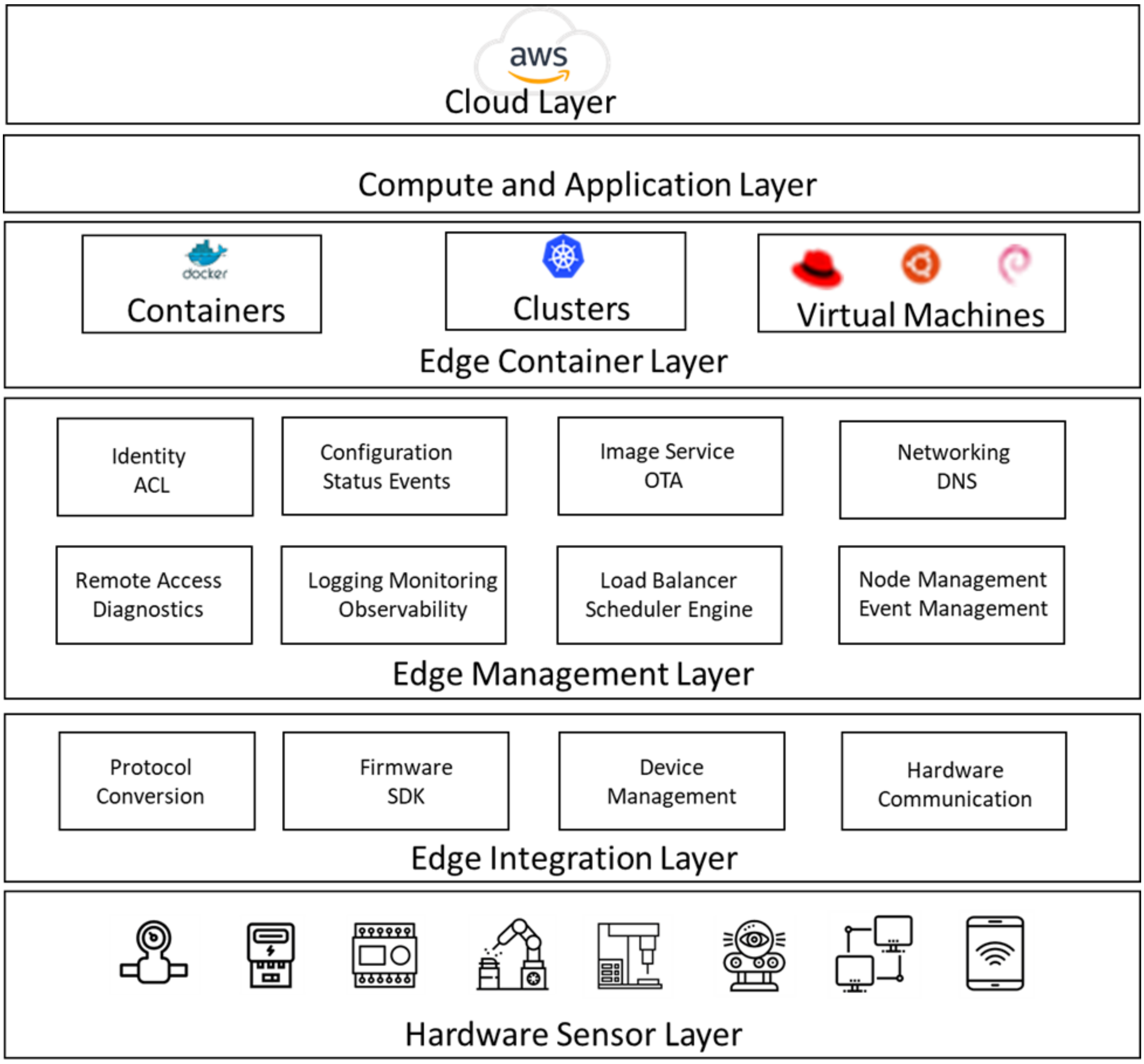
Performance = (Ideal Cycle Time x Total Count) / Run Time

Quality = Good Count / Total Count


Visualization and Reporting
Analytics and Machine Learning
Data Storage and Transformation
Cloud Data Ingestion
Device Management
WAN
Edge Servers and Applications
Device Control Systems
Sensors and Actuators
Alerts | Faults | Operational Efficiency| Production Rate| Asset Health | Environmental Index
Acquisition | Framing | Packaging | Encapsulation | Security | Transmission
pH | TDS | Turbidity | Dissolved O_2 | Flow | Pressure | Energy | Conductivity | Chlorine
Sewage Treatment Plant | Effluent Treatment Plant | Waste-Water Treatment Plant
Knowledge
Information
Data
Parameters
Decision Actuation Control

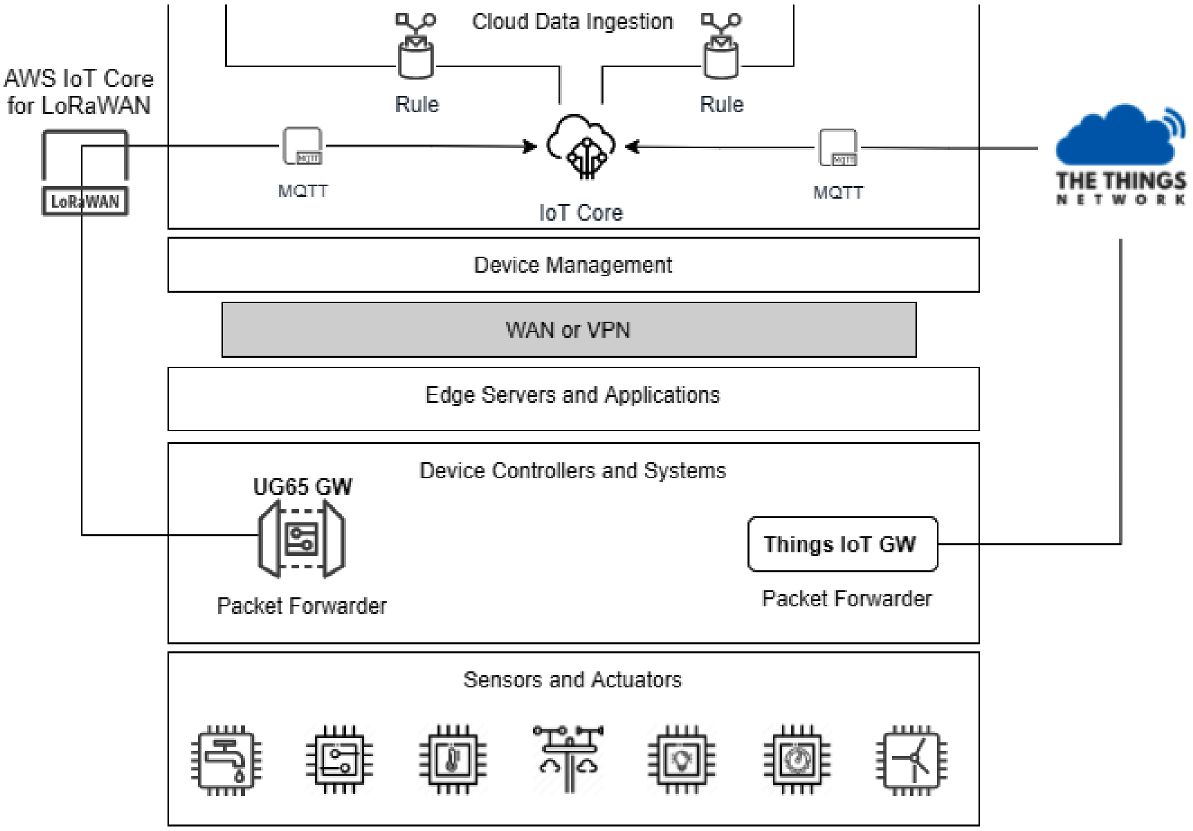

Power Consumption
Data Rate
100 MBps
1 MBps
100 KBps
1 KBps
WiFi
Cellular
4G
5G
Bluetooth
Licensed
LTE-M
NB IoT
Others
Zigbee
RFID
Unlicensed
LoRa
Sigfox
1 Meter
10 Meters
100 Meters
1 Km
10 Km
Range

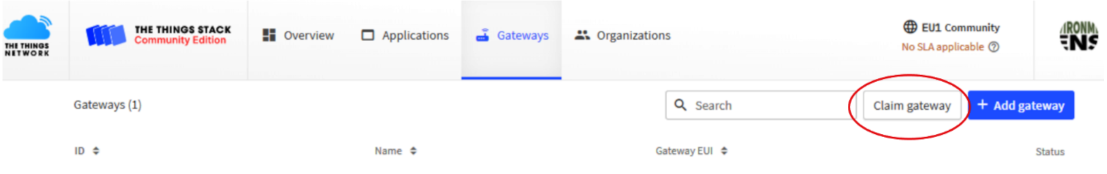

Cloud
Communicate
Acquire
Shopfloor
Cloud
Communicate
Compute
Actuate
Store
Acquire
Shopfloor


aws
Cloud Layer
Compute and Application Layer
docker
Containers
Clusters
Virtual Machines
Edge Container Layer
Identity
ACL
Configuration
Status Events
Image Service
OTA
Networking
DNS
Remote Access
Diagnostics
Logging Monitoring
Observability
Load Balancer
Scheduler Engine
Node Management
Event Management
Edge Management Layer
Protocol
Conversion
Firmware
SDK
Device
Management
Hardware
Communication
Edge Integration Layer
Hardware Sensor Layer

Chapter 4: Real-World Environmental Monitoring

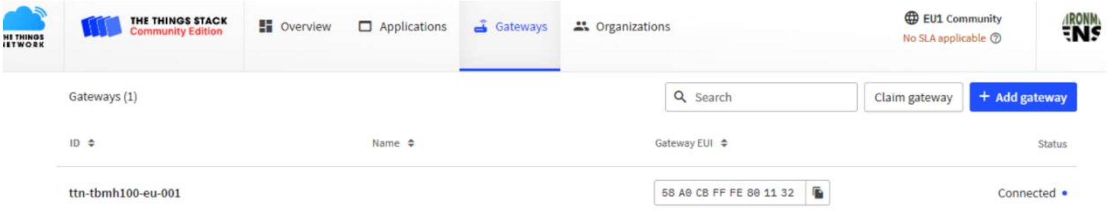
Sensor (environment) → **Gateway** (strategically placed) → **Network Server** (cloud based)





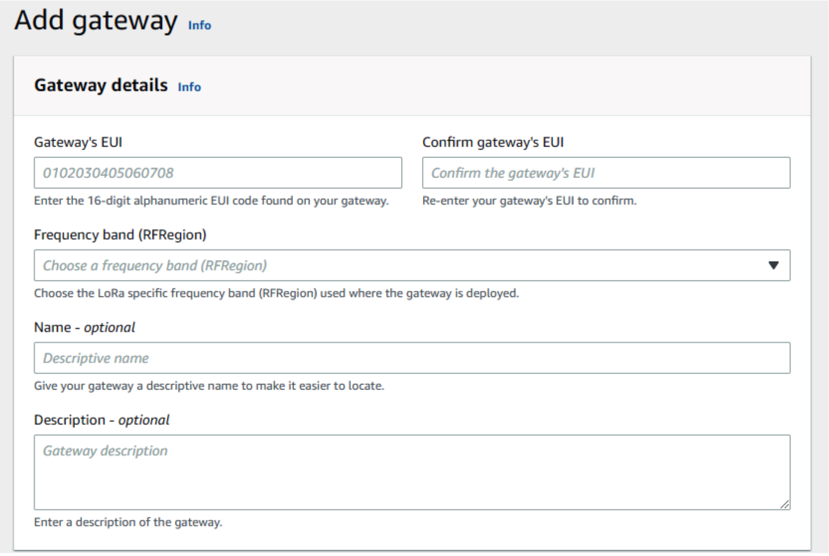
Gateways (1)

ID	Name	Gateway EUI	Status



Gateways (1)

ID	Name	Gateway EUI	Status
ttn-tbmh100-eu-001		58 A0 CB FF FE 80 11 32	Connected •



Add gateway Info

Gateway details Info

Gateway's EUI

0102030405060708

Enter the 16-digit alphanumeric EUI code found on your gateway.

Confirm gateway's EUI

Confirm the gateway's EUI

Re-enter your gateway's EUI to confirm.

Frequency band (RFRegion)

Choose a frequency band (RFRegion) ▼

Choose the LoRa specific frequency band (RFRegion) used where the gateway is deployed.

Name - *optional*

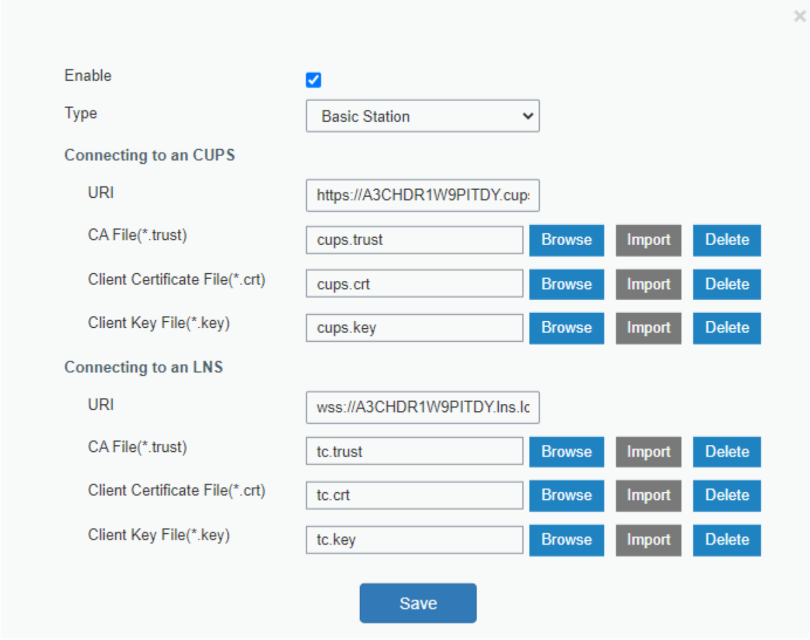
Descriptive name

Give your gateway a descriptive name to make it easier to locate.

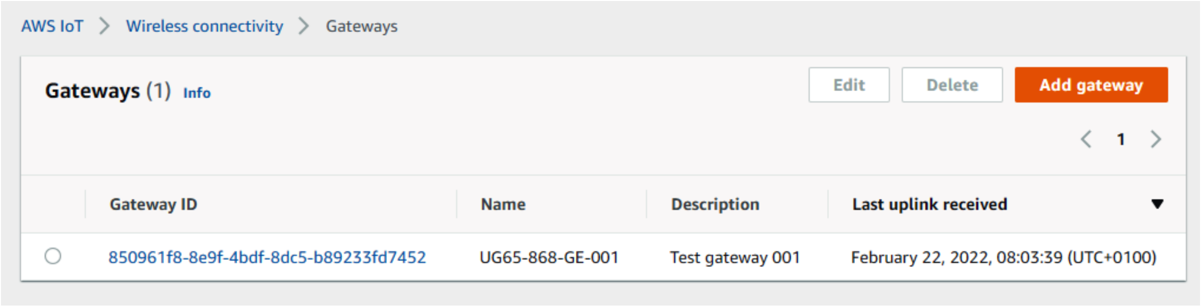
Description - *optional*

Gateway description

Enter a description of the gateway.



Enable	☑
Type	Basic Station
Connecting to an CUPS	
URI	https://A3CHDR1W9PITDY.cup:
CA File(*.trust)	cups.trust
Client Certificate File(*.crt)	cups.crt
Client Key File(*.key)	cups.key
Connecting to an LNS	
URI	wss://A3CHDR1W9PITDY.lns.lc
CA File(*.trust)	tc.trust
Client Certificate File(*.crt)	tc.crt
Client Key File(*.key)	tc.key



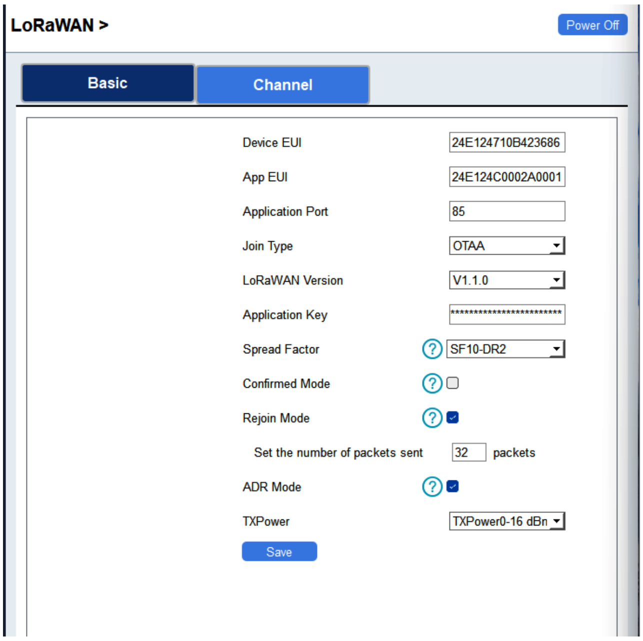
AWS IoT > Wireless connectivity > Gateways

Gateways (1) Info

Edit | Delete | Add gateway

< 1 >

	Gateway ID	Name	Description	Last uplink received
○	850961f8-8e9f-4bdf-8dc5-b89233fd7452	UG65-868-GE-001	Test gateway 001	February 22, 2022, 08:03:39 (UTC+0100)



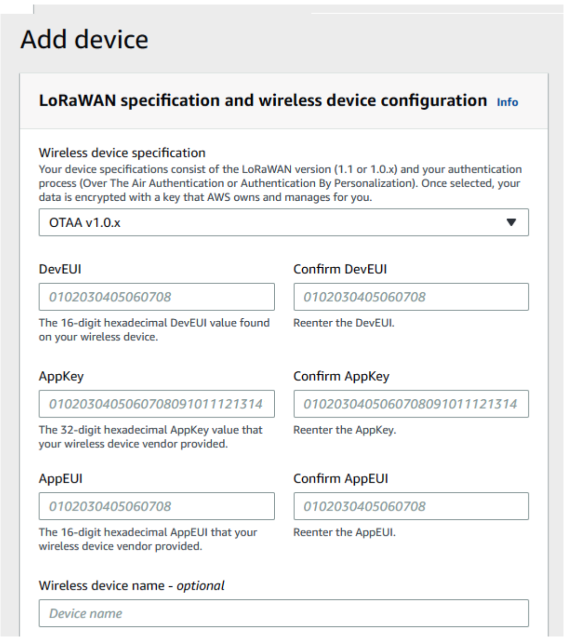
LoRaWAN >

Power Off

Basic | Channel

Device EUI	24E124710B423686
App EUI	24E124C0002A0001
Application Port	85
Join Type	OTAA
LoRaWAN Version	V1.1.0
Application Key	************************
Spread Factor	SF10-DR2
Confirmed Mode	□
Rejoin Mode	☑
Set the number of packets sent	32 packets
ADR Mode	☑
TXPower	TXPower0-16 dBm

Save



Add device

LoRaWAN specification and wireless device configuration Info

Wireless device specification

Your device specifications consist of the LoRaWAN version (1.1 or 1.0.x) and your authentication process (Over The Air Authentication or Authentication By Personalization). Once selected, your data is encrypted with a key that AWS owns and manages for you.

OTAA v1.0.x

DevEUI: 0102030405060708
The 16-digit hexadecimal DevEUI value found on your wireless device.

Confirm DevEUI: 0102030405060708
Reenter the DevEUI.

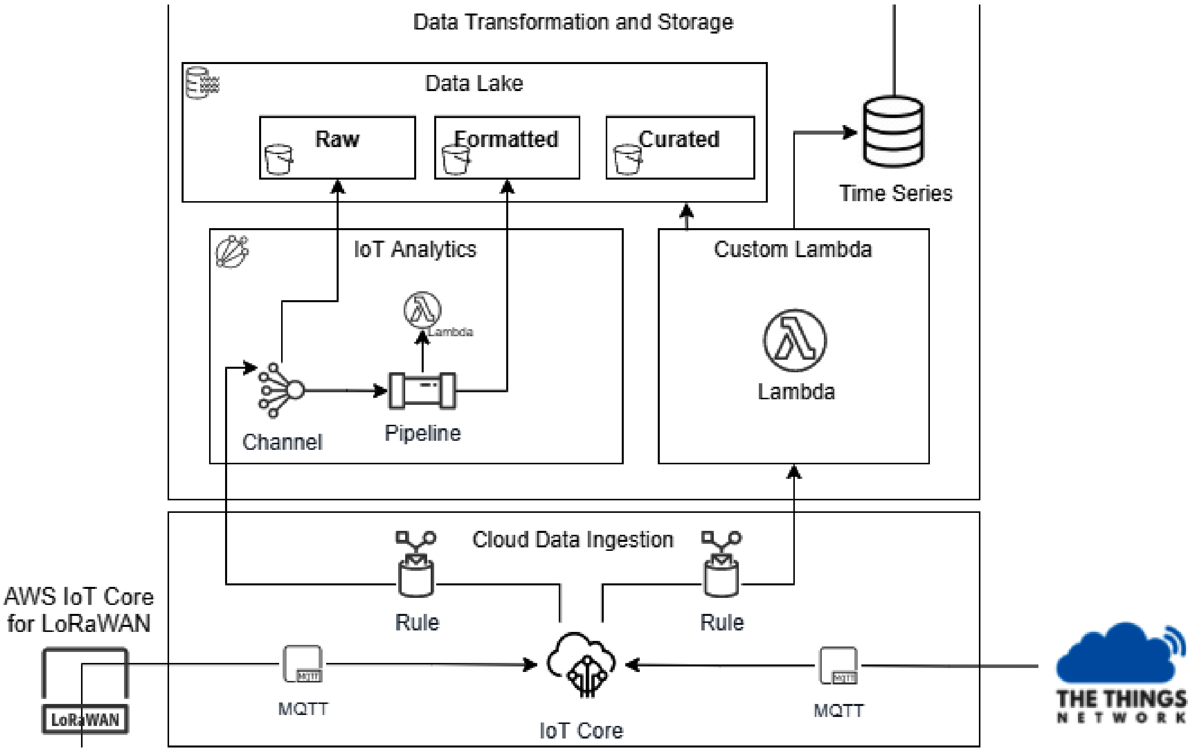
AppKey: 0102030405060708091011121314
The 32-digit hexadecimal AppKey value that your wireless device vendor provided.

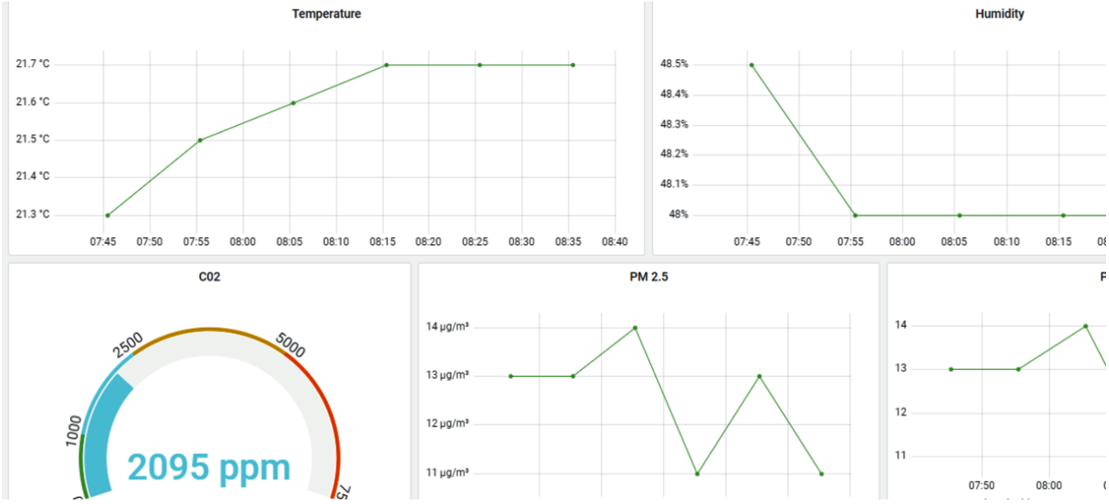
Confirm AppKey: 0102030405060708091011121314
Reenter the AppKey.

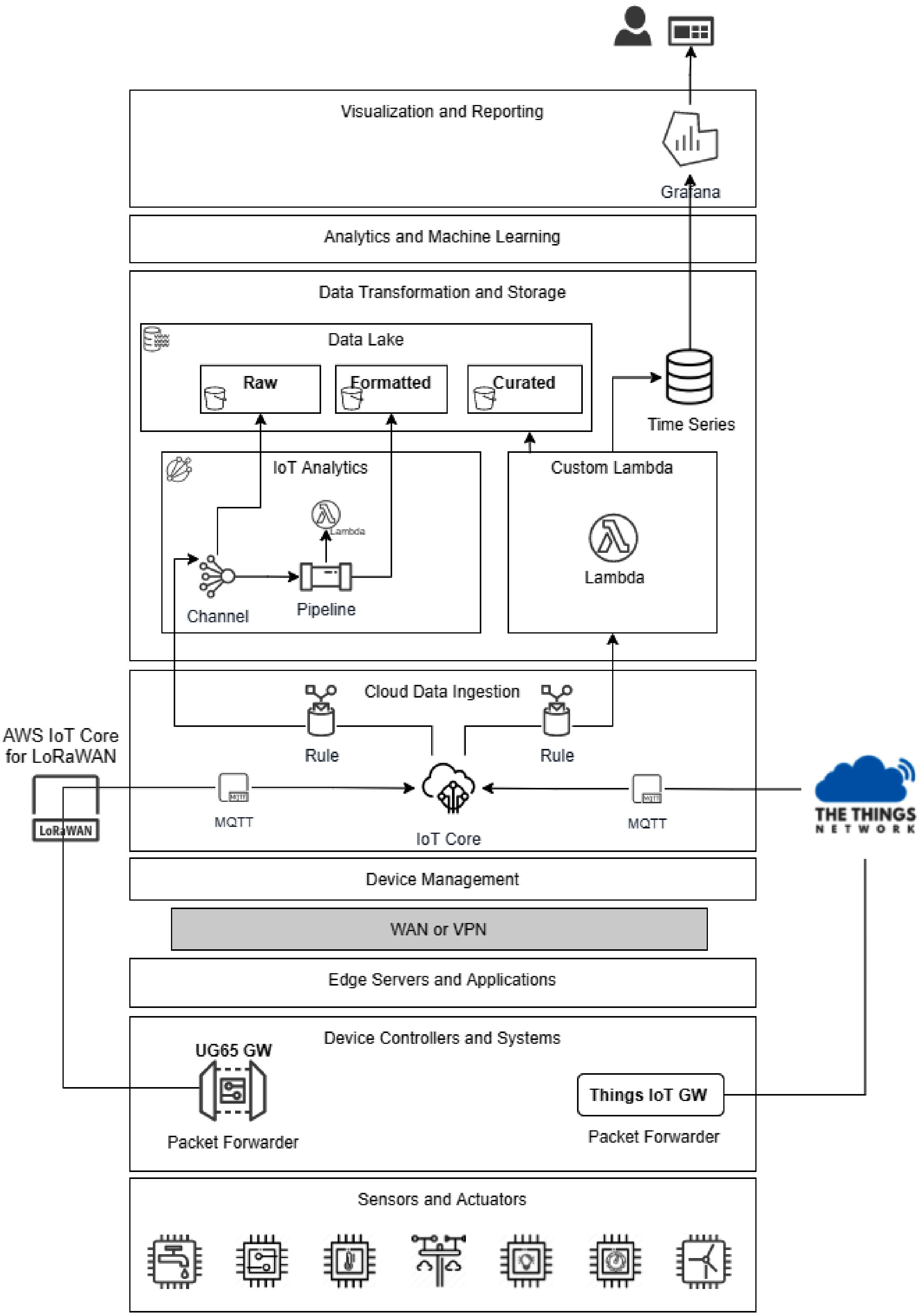
AppEUI: 0102030405060708
The 16-digit hexadecimal AppEUI that your wireless device vendor provided.

Confirm AppEUI: 0102030405060708
Reenter the AppEUI.

Wireless device name - *optional*: Device name

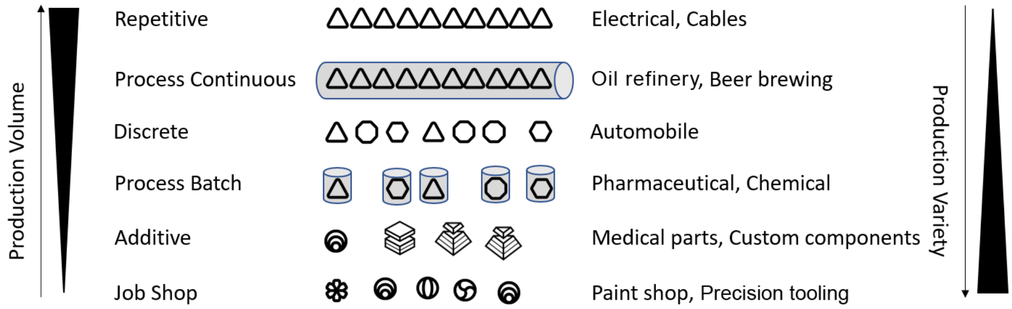

Data Transformation and Storage
Data Lake
Raw
Formatted
Curated
Time Series
IoT Analytics
Lambda
Channel
Pipeline
Custom Lambda
Lambda
Cloud Data Ingestion
Rule
Rule
AWS IoT Core
for LoRaWAN
LoRaWAN
MQTT
MQTT
MQTT
MQTT
IoT Core
THE THINGS
NETWORK

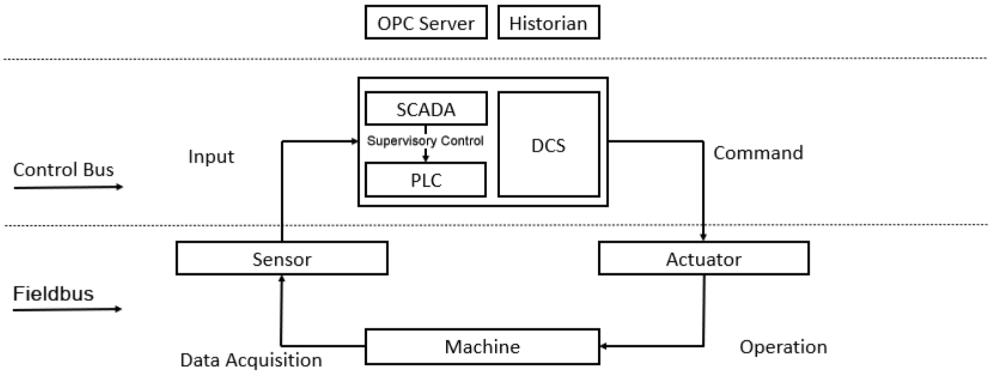

Temperature
21.7 °C
21.6 °C
21.5 °C
21.4 °C
21.3 °C
07:45
07:50
07:55
08:00
08:05
08:10
08:15
08:20
08:25
08:30
08:35
08:40
Humidity
48.5%
48.4%
48.3%
48.2%
48.1%
48%
07:45
07:50
07:55
08:00
08:05
08:10
08:15
C02
2500
5000
1000
2095 ppm
PM 2.5
14 µg/m³
13 µg/m³
12 µg/m³
11 µg/m³
14
13
12
11
07:50
08:00

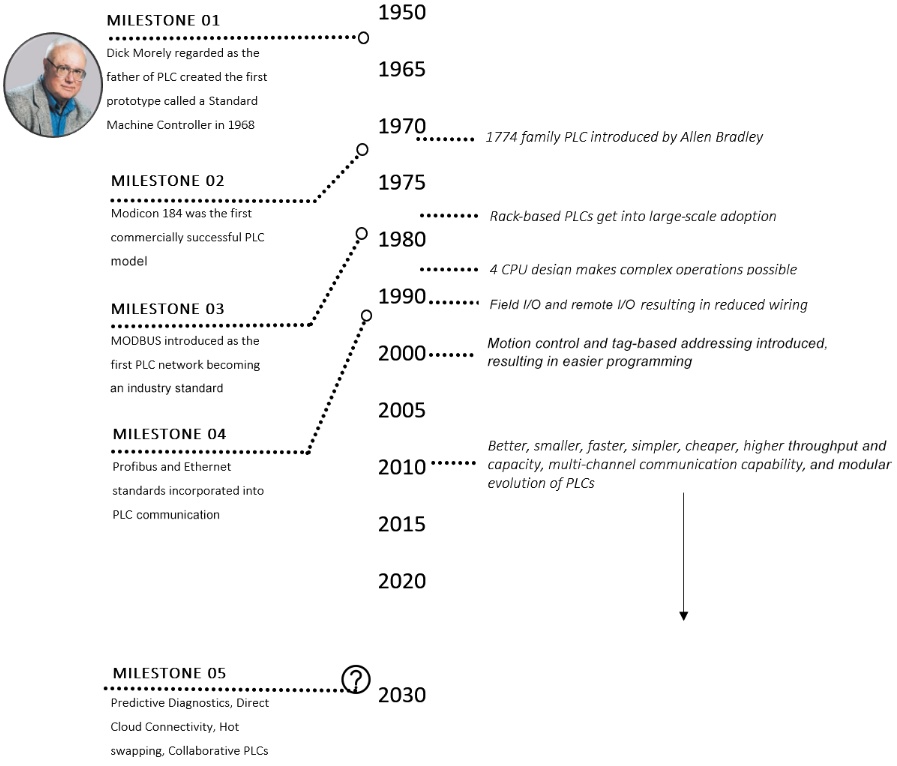


Visualization and Reporting
Grafana
Analytics and Machine Learning
Data Transformation and Storage
Data Lake
Raw
Formatted
Curated
Time Series
IoT Analytics
Lambda
Channel
Pipeline
Custom Lambda
Lambda
Cloud Data Ingestion
Rule
Rule
AWS IoT Core for LoRaWAN
LoRaWAN
MQTT
IoT Core
MQTT
THE THINGS NETWORK
Device Management
WAN or VPN
Edge Servers and Applications
Device Controllers and Systems
UG65 GW
Packet Forwarder
Things IoT GW
Packet Forwarder
Sensors and Actuators

Chapter 5: OT and Industrial Control Systems







MILESTONE 01

Dick Morely regarded as the father of PLC created the first prototype called a Standard Machine Controller in 1968

MILESTONE 02

Modicon 184 was the first commercially successful PLC model

MILESTONE 03

MODBUS introduced as the first PLC network becoming an industry standard

MILESTONE 04

Profibus and Ethernet standards incorporated into PLC communication

MILESTONE 05

Predictive Diagnostics, Direct Cloud Connectivity, Hot swapping, Collaborative PLCs

1774 family PLC introduced by Allen Bradley

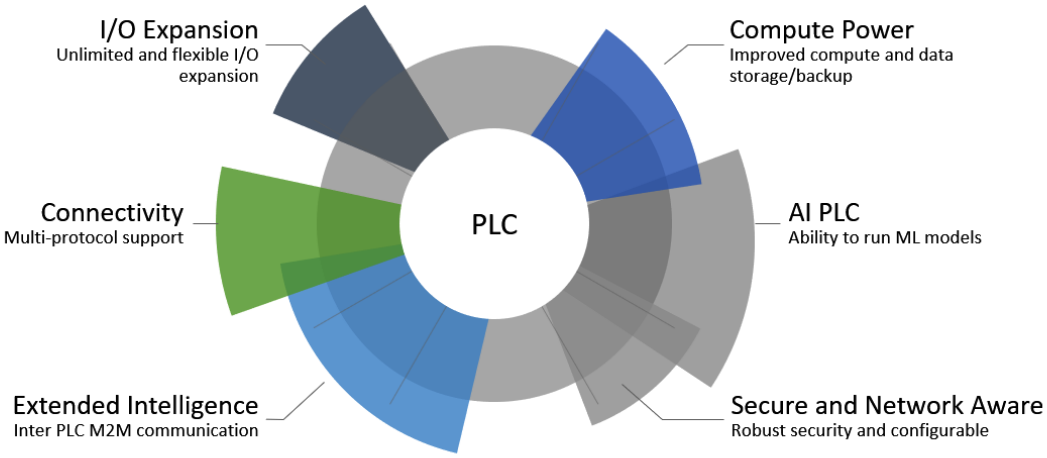
Rack-based PLCs get into large-scale adoption

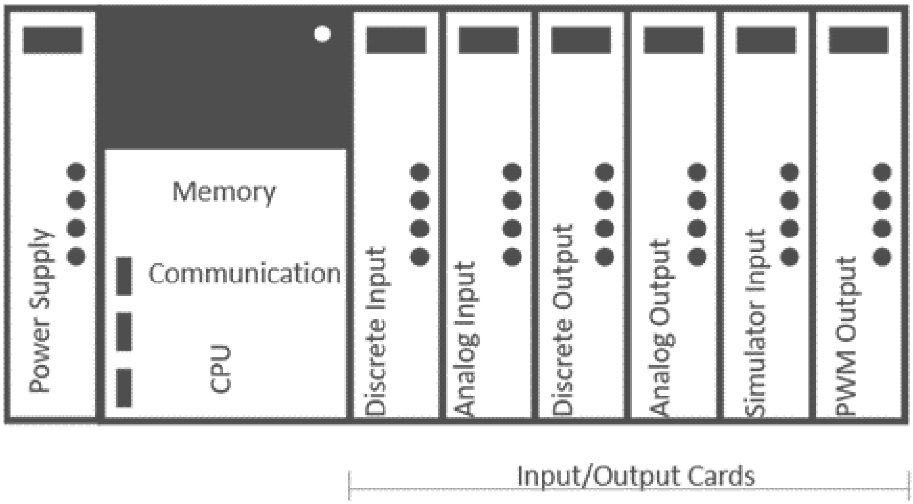
4 CPU design makes complex operations possible

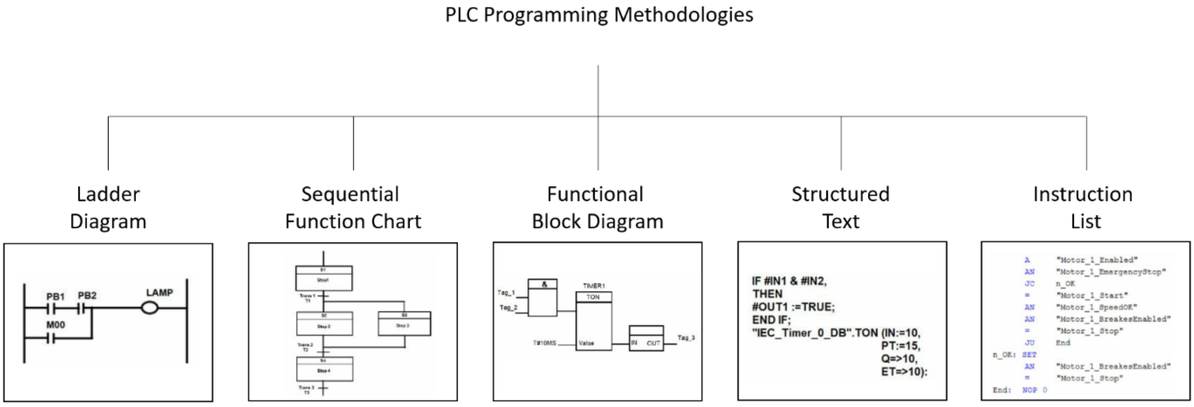
Field I/O and remote I/O resulting in reduced wiring

Motion control and tag-based addressing introduced, resulting in easier programming

Better, smaller, faster, simpler, cheaper, higher throughput and capacity, multi-channel communication capability, and modular evolution of PLCs

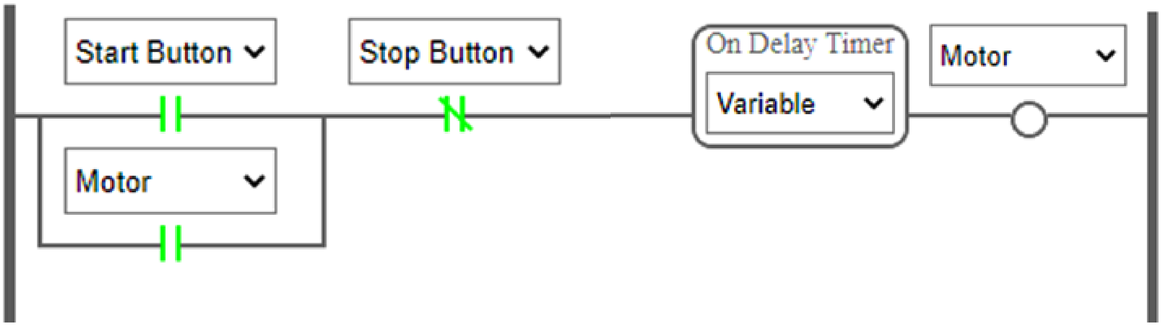


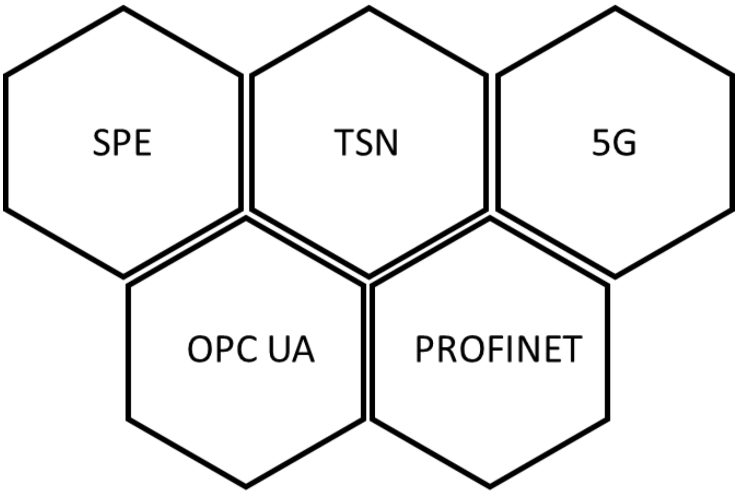


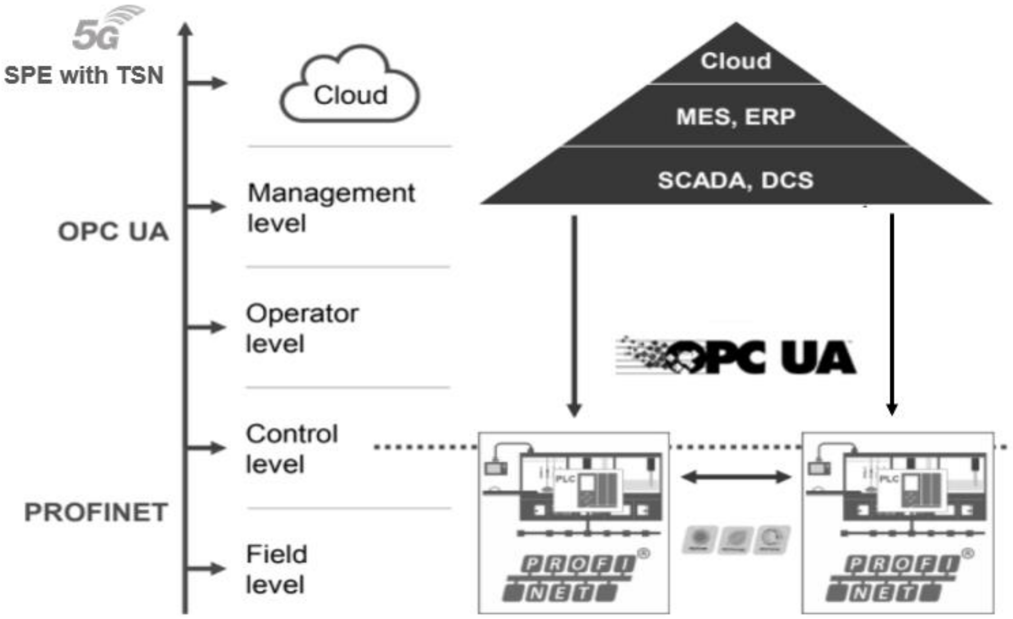


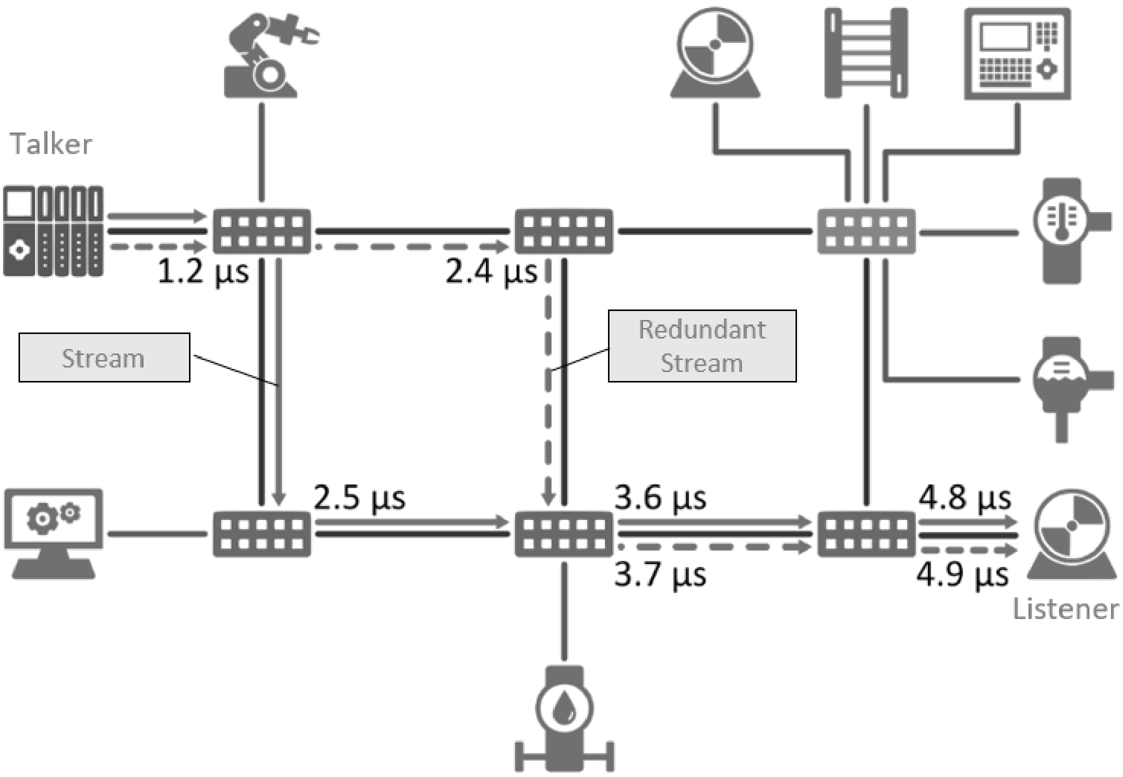
```
IF #IN1 & #IN2,
THEN
#OUT1 :=TRUE;
END IF;
"IEC_Timer_0_DB".TON (IN:=10,
                      PT:=15,
                      Q=>10,
                      ET=>10):
```

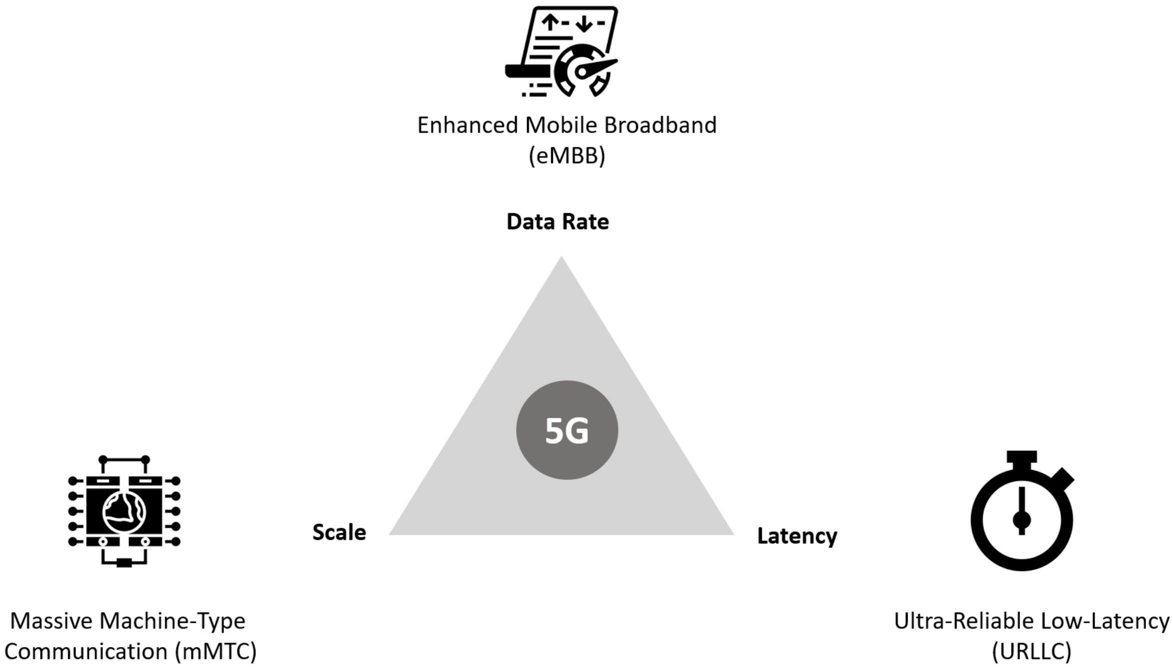
```
        A       "Motor_1_Enabled"
        AN      "Motor_1_EmergencyStop"
        JC      n_OK
        =       "Motor_1_Start"
        AN      "Motor_1_SpeedOK"
        AN      "Motor_1_BreakesEnabled"
        =       "Motor_1_Stop"
        JU      End
n_OK:   SET
        AN      "Motor_1_BreakesEnabled"
        =       "Motor_1_Stop"
End:    NOP 0
```

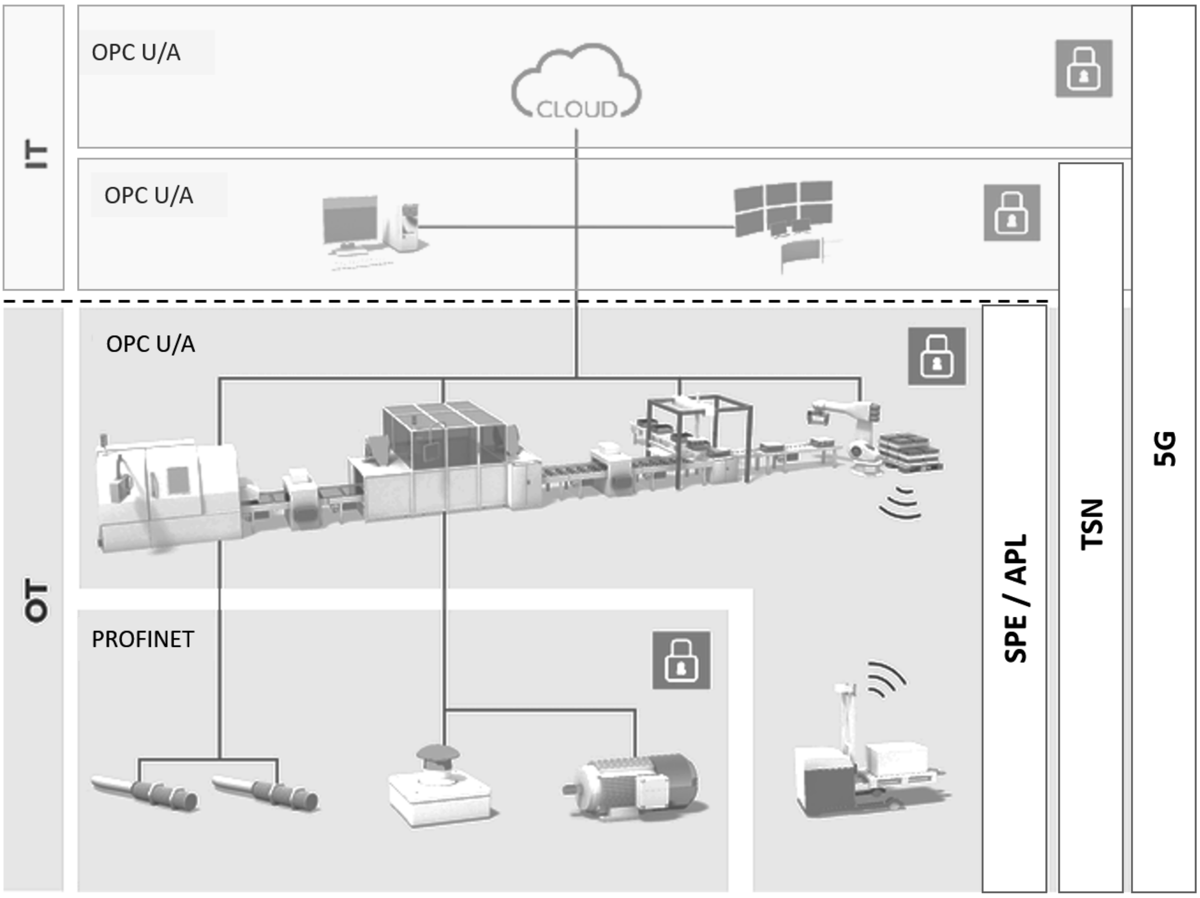

Start Button
Stop Button
On Delay Timer
Variable
Motor
Motor

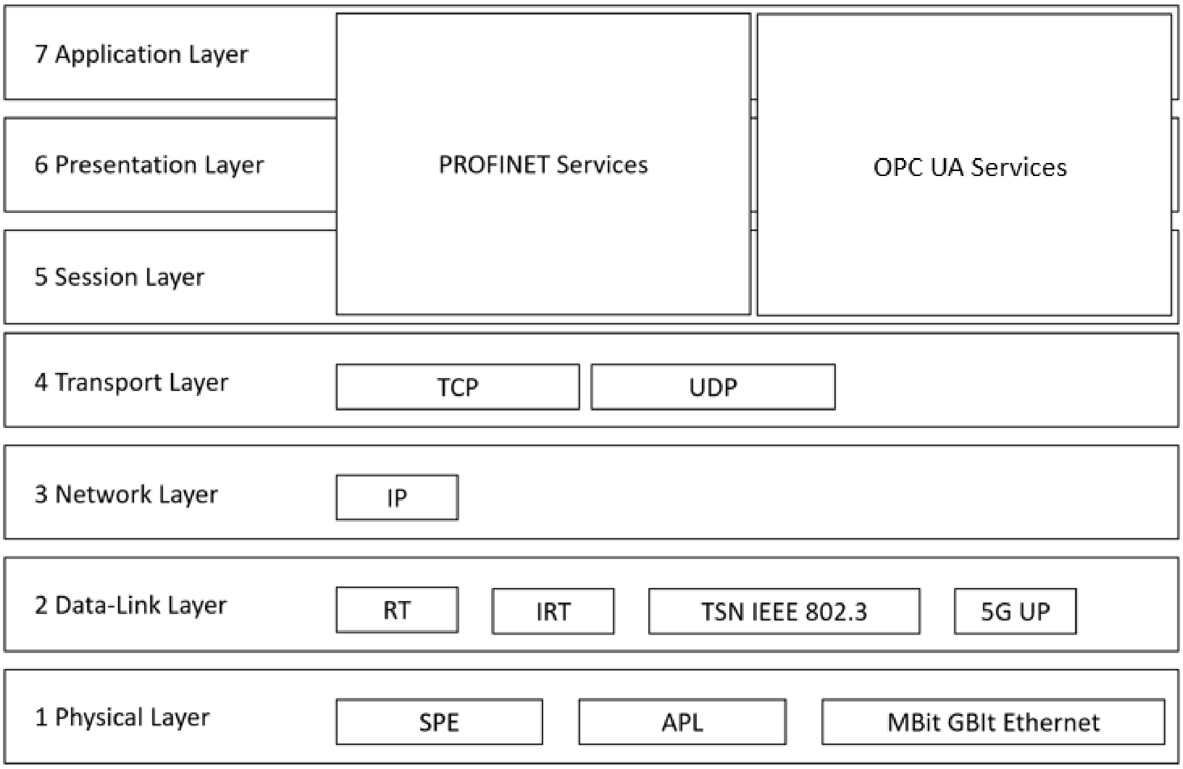

SPE
TSN
5G
OPC UA
PROFINET


5G
SPE with TSN
OPC UA
PROFINET
Cloud
Management level
Operator level
Control level
Field level
Cloud
MES, ERP
SCADA, DCS
OPC UA
PLC
PROFI NET
PLC
PROFI NET


Talker
1.2 µs
2.4 µs
Stream
Redundant Stream
2.5 µs
3.6 µs
3.7 µs
4.8 µs
4.9 µs
Listener


Enhanced Mobile Broadband (eMBB)
Data Rate
5G
Scale
Latency
Massive Machine-Type Communication (mMTC)
Ultra-Reliable Low-Latency (URLLC)





7 Application Layer

6 Presentation Layer

5 Session Layer

PROFINET Services

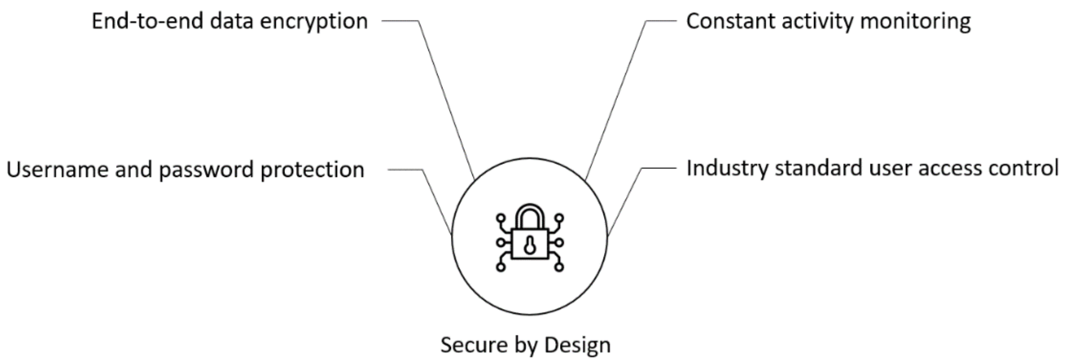
OPC UA Services

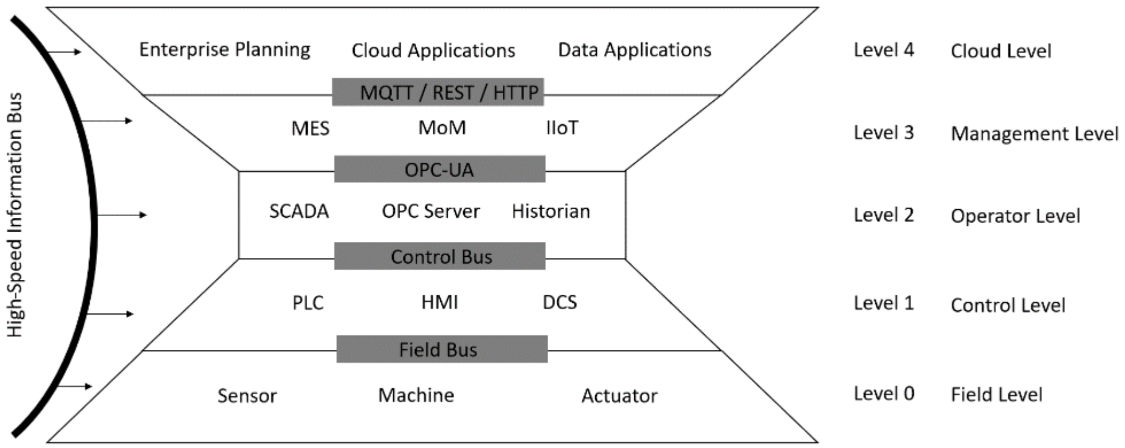
4 Transport Layer: TCP, UDP

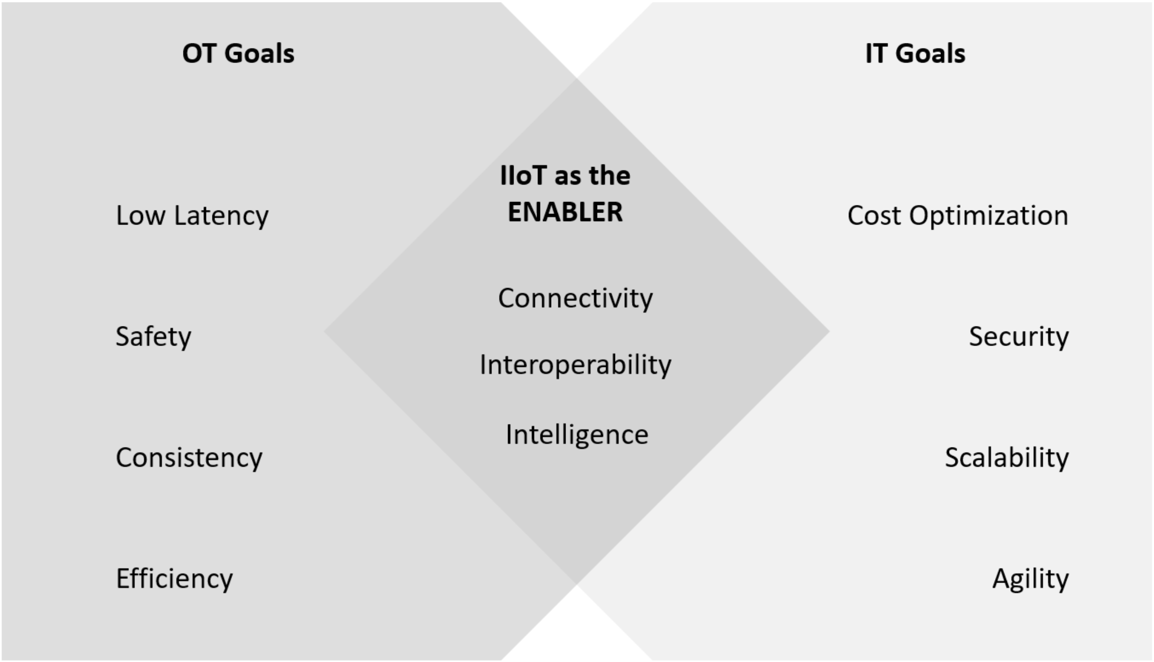
3 Network Layer: IP

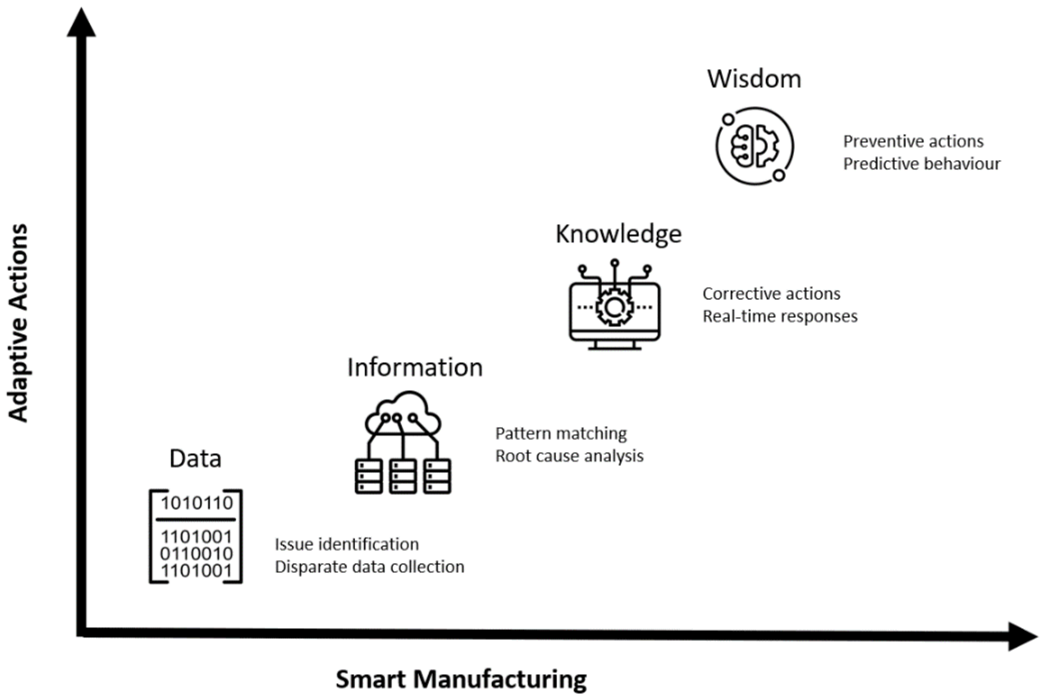
2 Data-Link Layer: RT, IRT, TSN IEEE 802.3, 5G UP

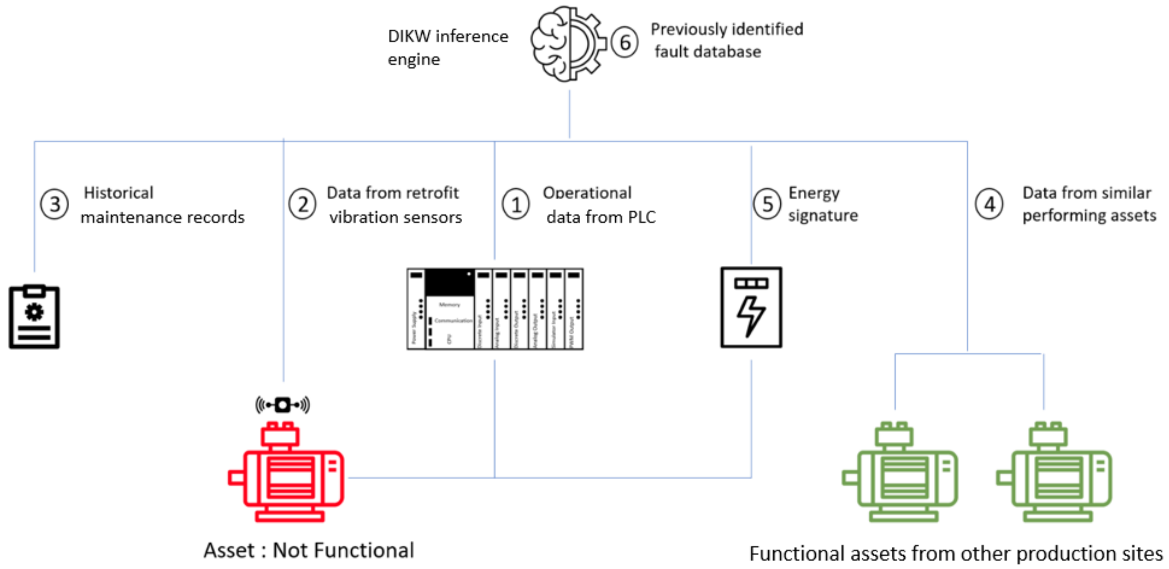
1 Physical Layer: SPE, APL, MBit GBIt Ethernet


End-to-end data encryption
Constant activity monitoring
Username and password protection
Industry standard user access control
Secure by Design

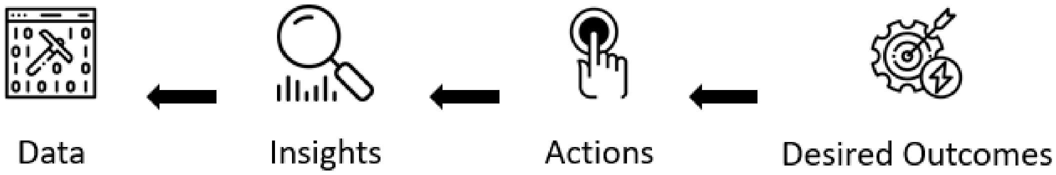

High-Speed Information Bus
Enterprise Planning
Cloud Applications
Data Applications
MQTT / REST / HTTP
MES
MoM
IIoT
OPC-UA
SCADA
OPC Server
Historian
Control Bus
PLC
HMI
DCS
Field Bus
Sensor
Machine
Actuator
Level 4
Cloud Level
Level 3
Management Level
Level 2
Operator Level
Level 1
Control Level
Level 0
Field Level

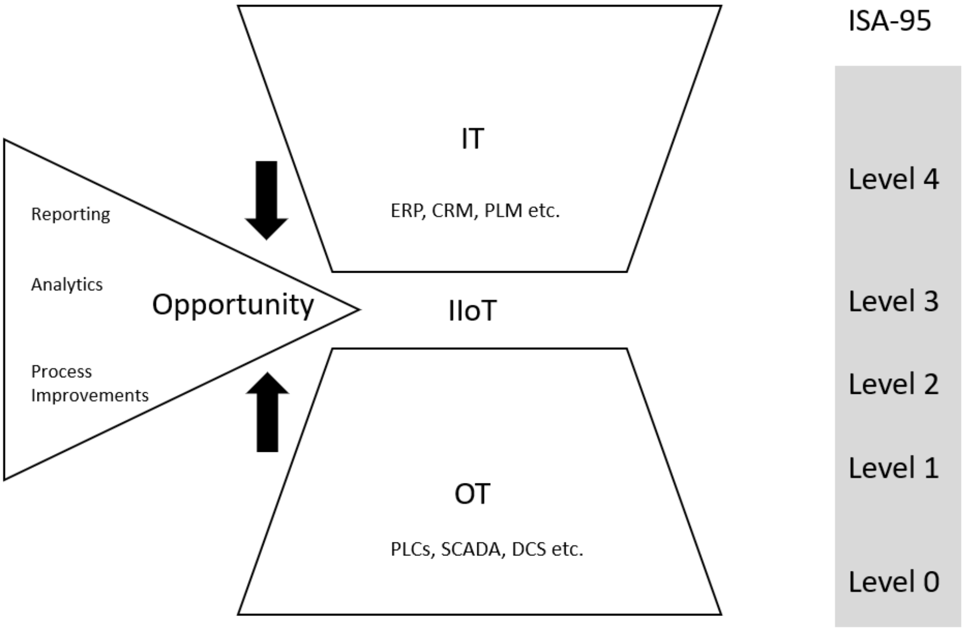

OT Goals
IT Goals
Low Latency
Safety
Consistency
Efficiency
IIoT as the ENABLER
Connectivity
Interoperability
Intelligence
Cost Optimization
Security
Scalability
Agility

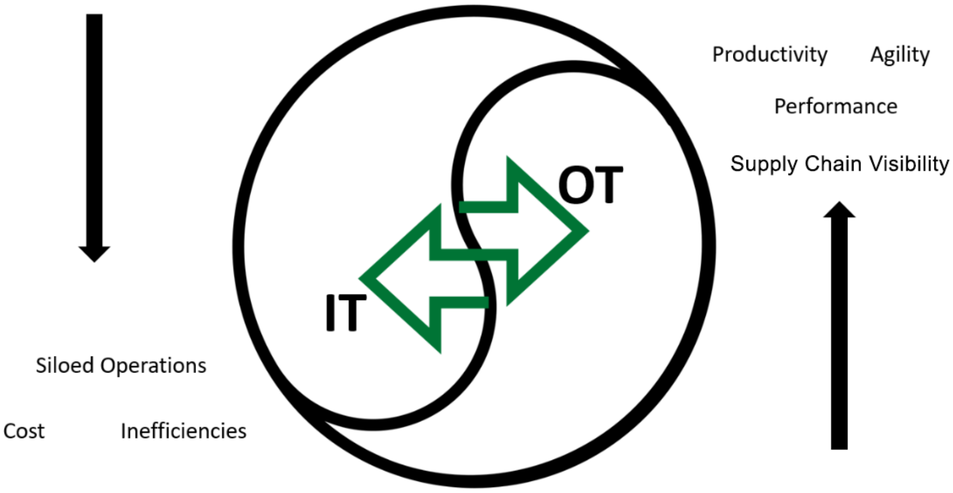

Wisdom
Preventive actions
Predictive behaviour
Knowledge
Corrective actions
Real-time responses
Information
Pattern matching
Root cause analysis
Data
1010110
1101001
0110010
1101001
Issue identification
Disparate data collection
Adaptive Actions
Smart Manufacturing

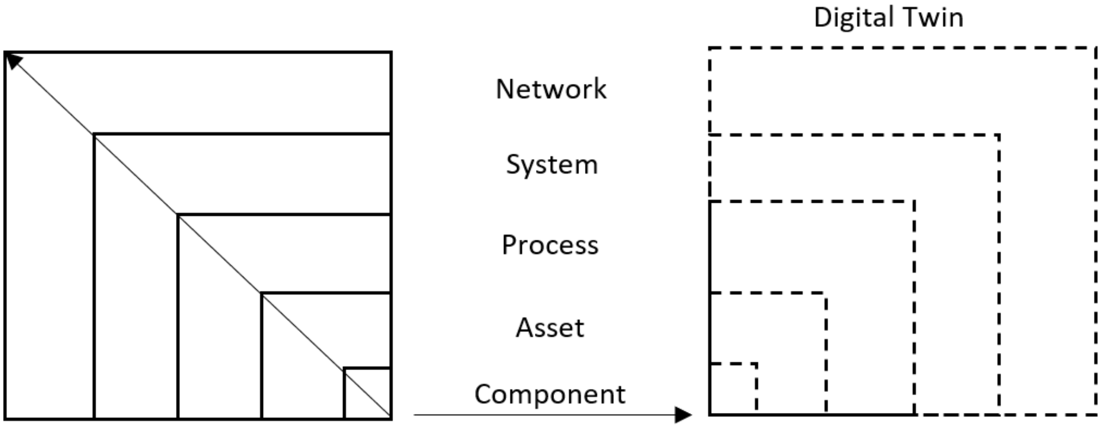

DIKW inference engine
6 Previously identified fault database
3 Historical maintenance records
2 Data from retrofit vibration sensors
1 Operational data from PLC
5 Energy signature
4 Data from similar performing assets
Asset : Not Functional
Functional assets from other production sites

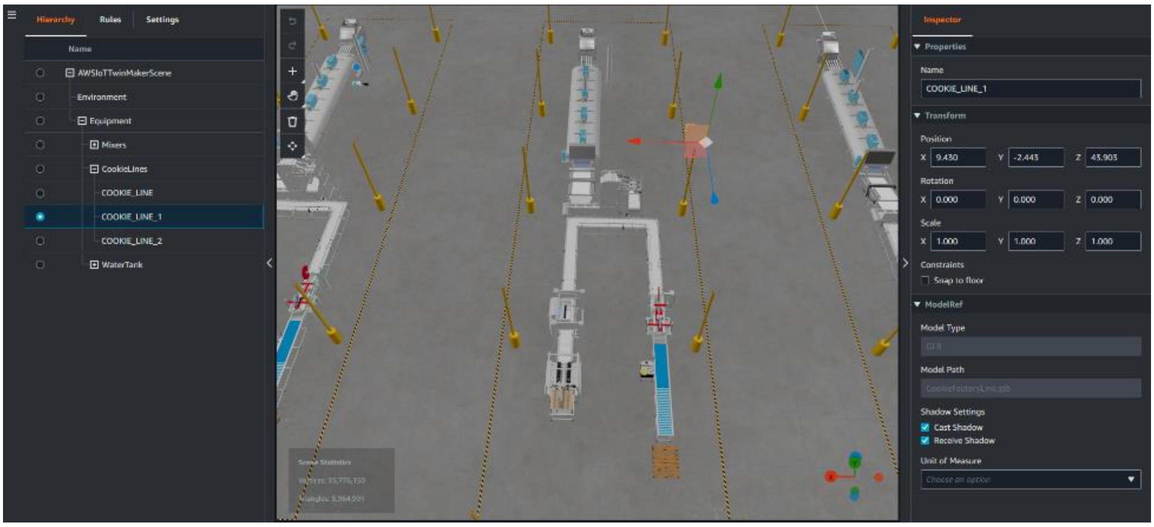
Chapter 6: Enabling Industrial IoT











Hierarchy | Rules | Settings

Name

○ ⊟ AWSIoTTwinMakerScene
○ Environment
○ ⊟ Equipment
○ ⊞ Mixers
○ ⊟ CookieLines
○ COOKIE_LINE
● COOKIE_LINE_1
○ COOKIE_LINE_2
○ ⊞ WaterTank

Inspector

▼ Properties

Name: COOKIE_LINE_1

▼ Transform

Position: X 9.430 Y -2.443 Z 43.903

Rotation: X 0.000 Y 0.000 Z 0.000

Scale: X 1.000 Y 1.000 Z 1.000

Constraints

□ Snap to floor

▼ ModelRef

Model Type: GLB

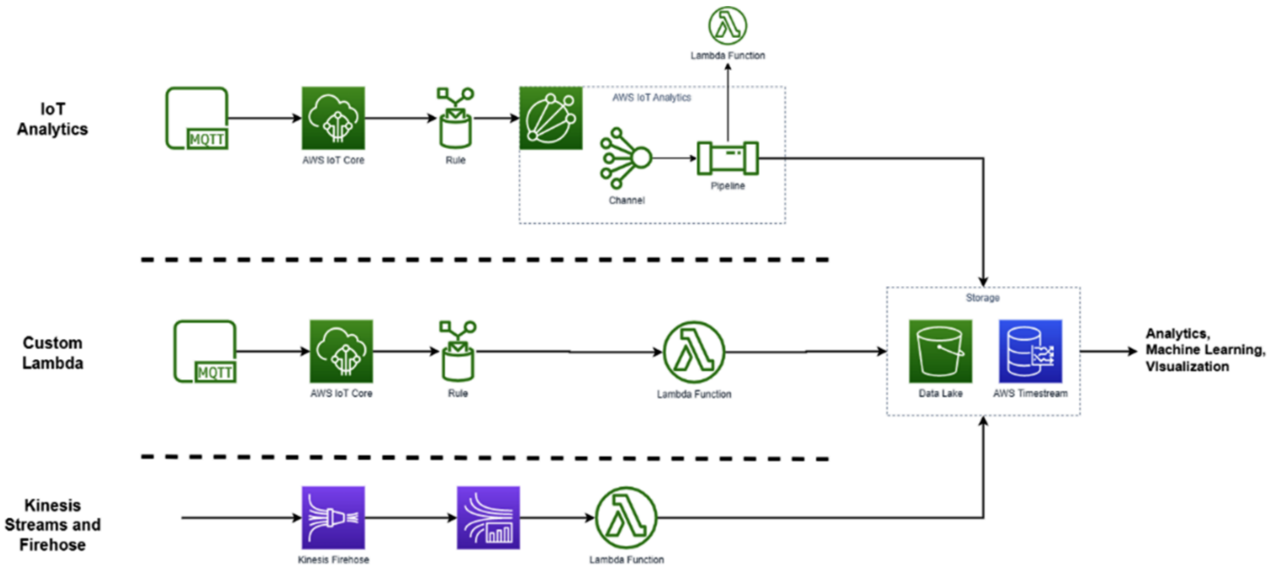
Model Path: CookieFactoryLine.glb

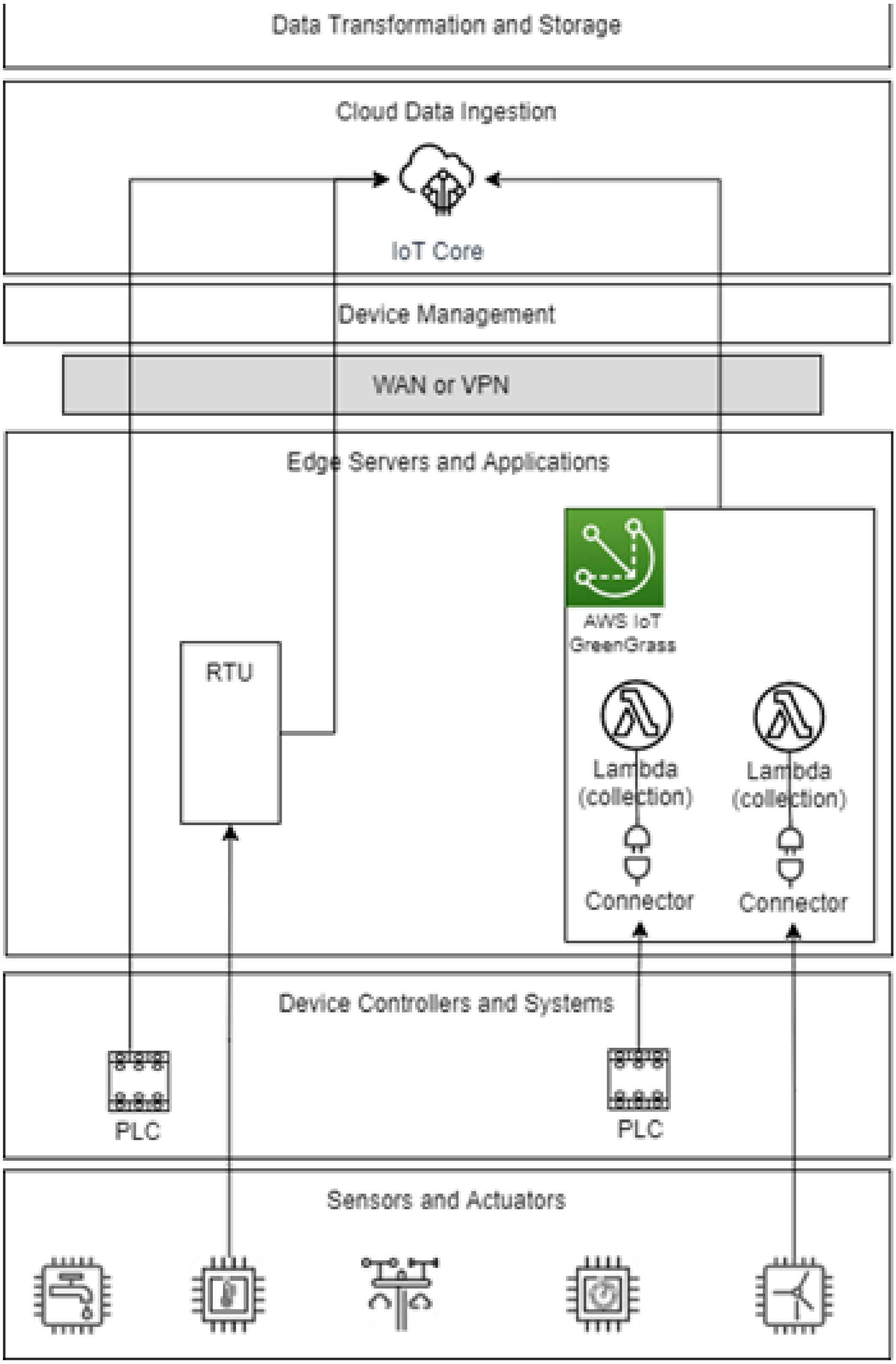
Shadow Settings

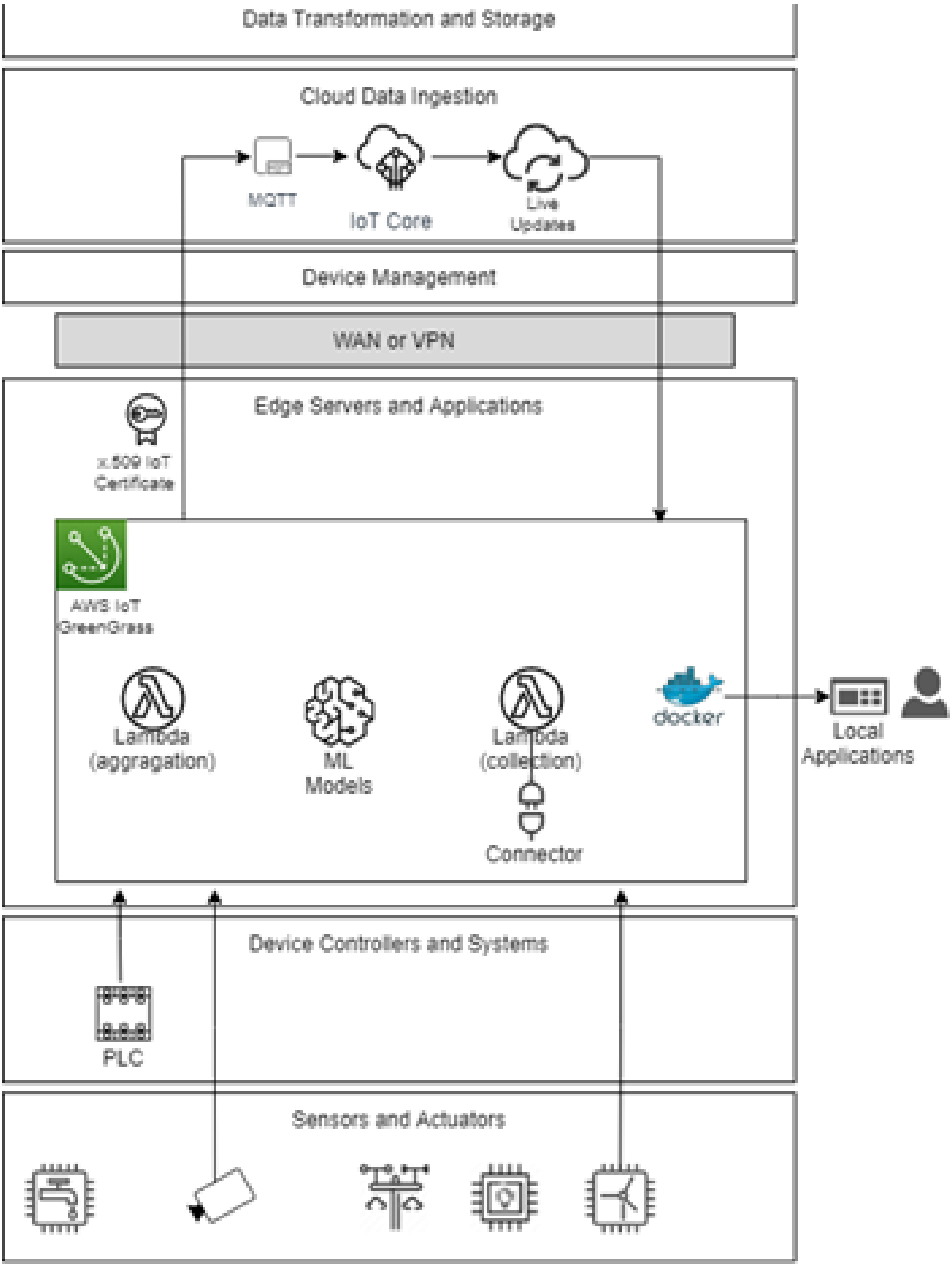
☑ Cast Shadow

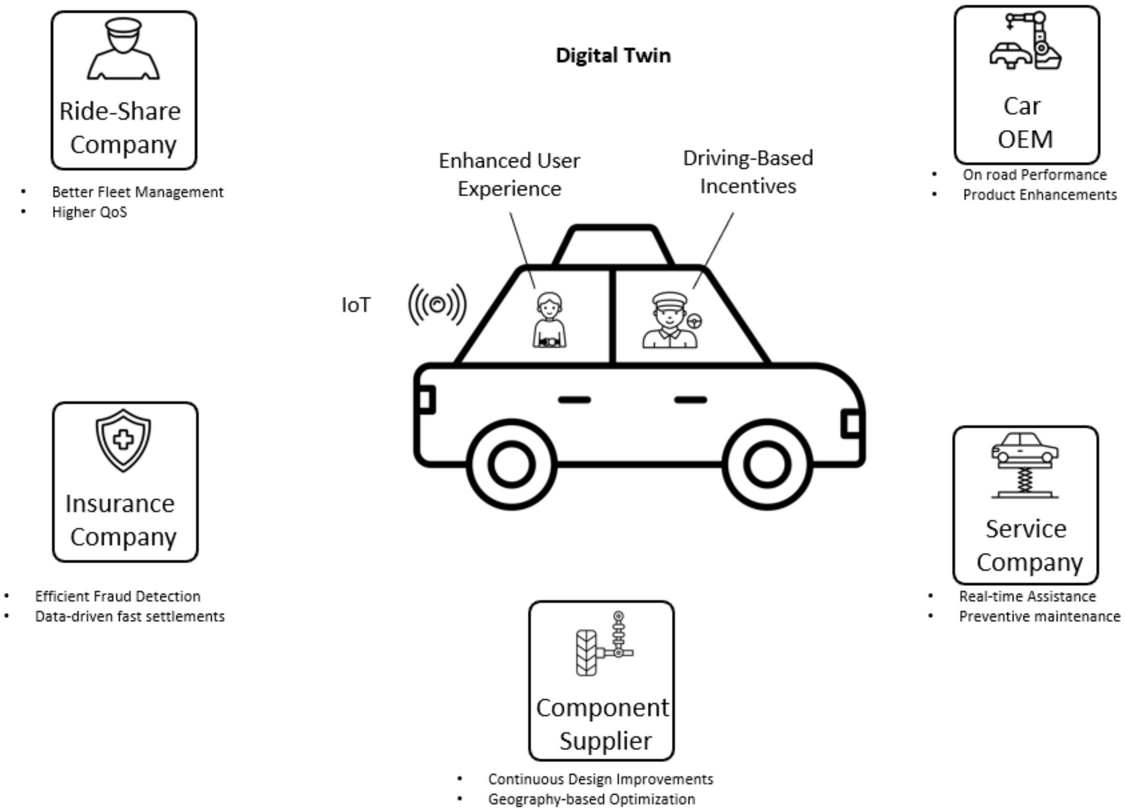
☑ Receive Shadow

Unit of Measure: Choose an option ▼

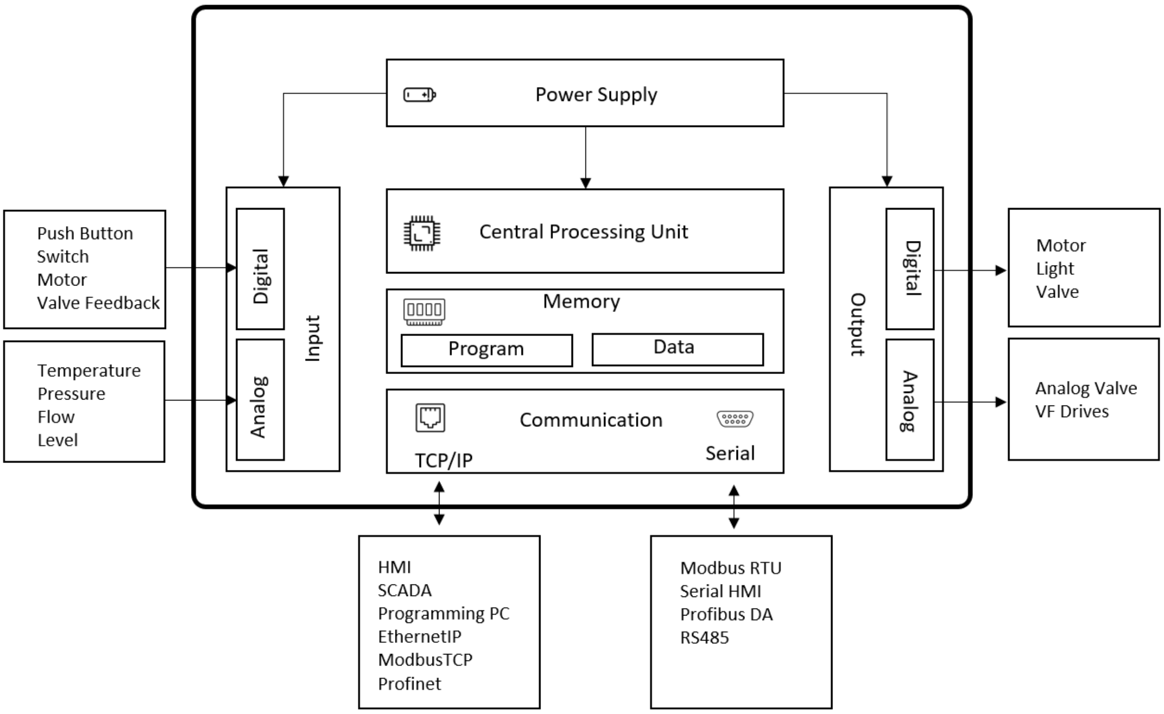


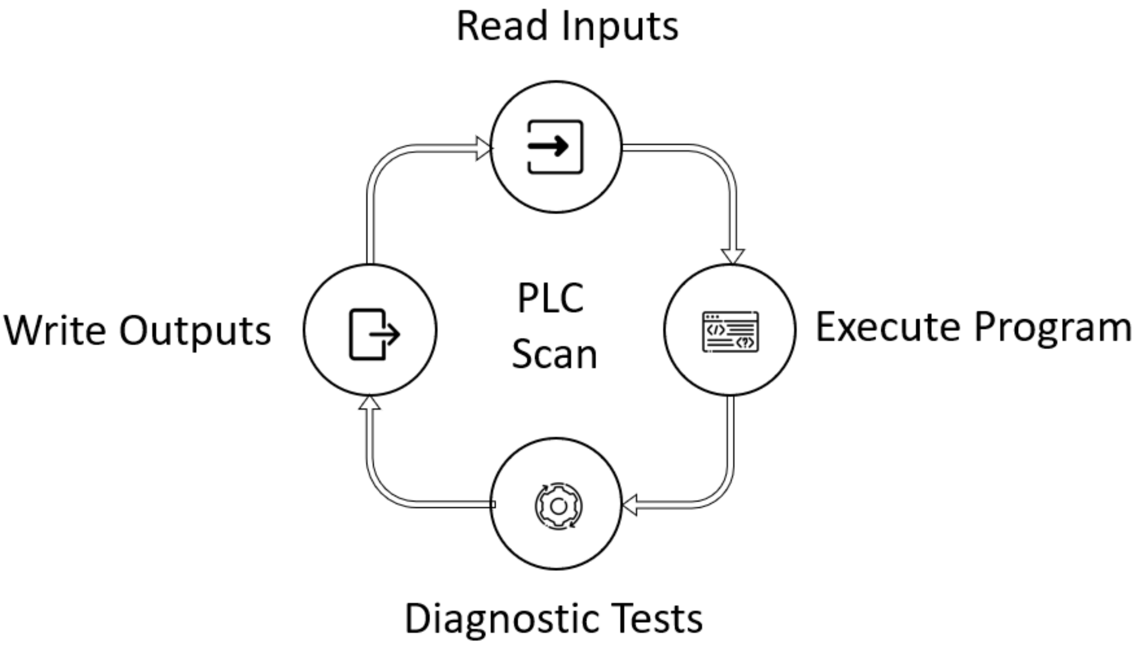

Data Transformation and Storage
Cloud Data Ingestion
IoT Core
Device Management
WAN or VPN
Edge Servers and Applications
RTU
AWS IoT
GreenGrass
Lambda
(collection)
Lambda
(collection)
Connector
Connector
Device Controllers and Systems
PLC
PLC
Sensors and Actuators

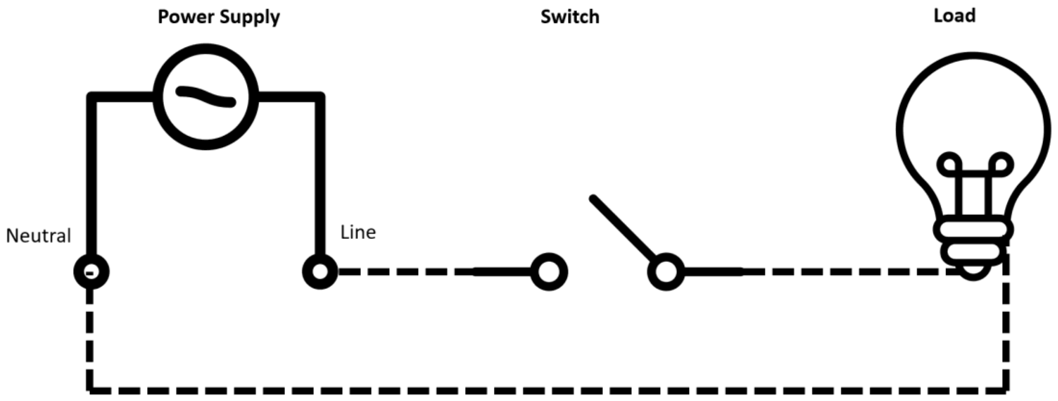

Data Transformation and Storage
Cloud Data Ingestion
MQTT
IoT Core
Live Updates
Device Management
WAN or VPN
Edge Servers and Applications
x.509 IoT Certificate
AWS IoT GreenGrass
Lambda (aggragation)
ML Models
Lambda (collection)
Connector
docker
Local Applications
Device Controllers and Systems
PLC
Sensors and Actuators

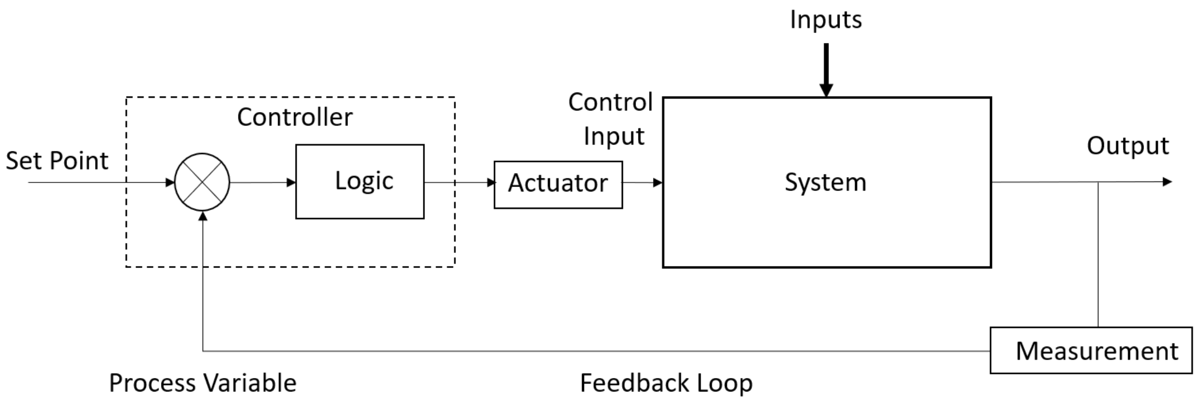

Digital Twin
Ride-Share Company
Better Fleet Management
Higher QoS
Car OEM
On road Performance
Product Enhancements
Enhanced User Experience
Driving-Based Incentives
IoT
Insurance Company
Efficient Fraud Detection
Data-driven fast settlements
Service Company
Real-time Assistance
Preventive maintenance
Component Supplier
Continuous Design Improvements
Geography-based Optimization

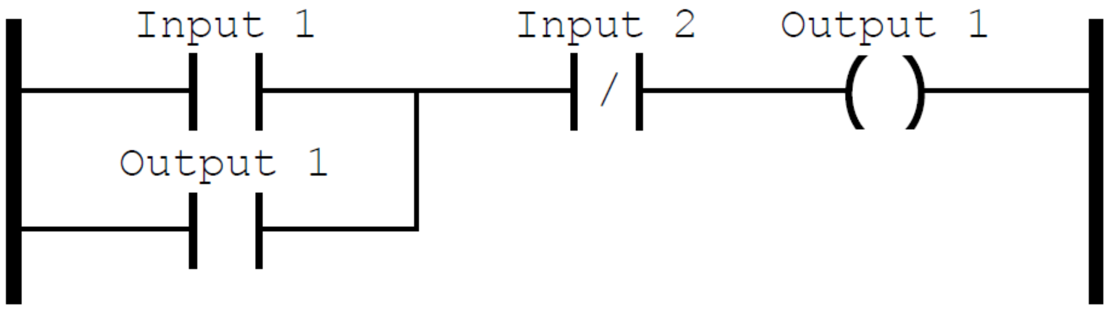
Chapter 7: PLC Data Acquisition and Analysis

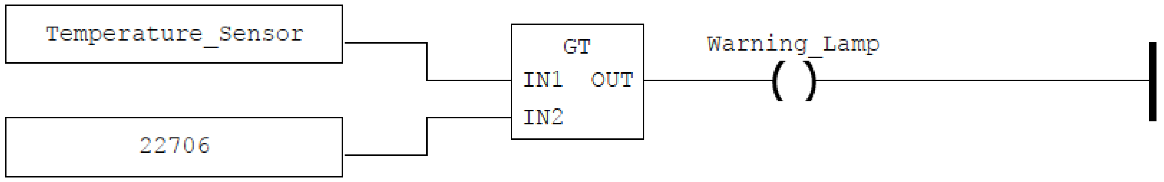


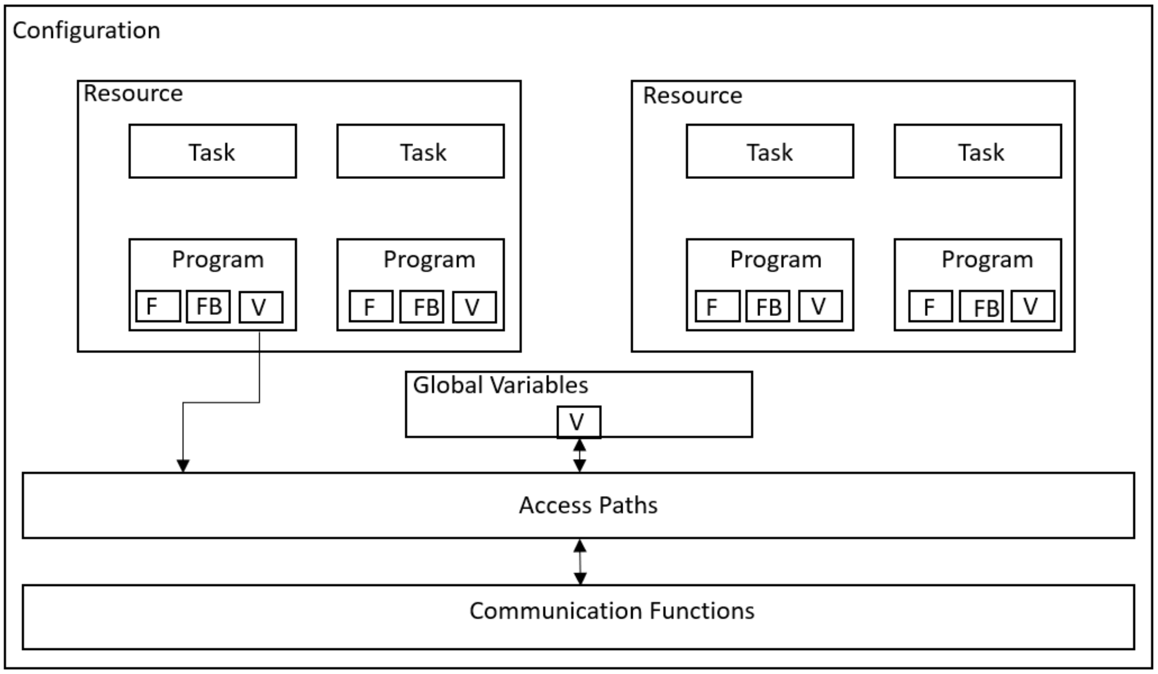


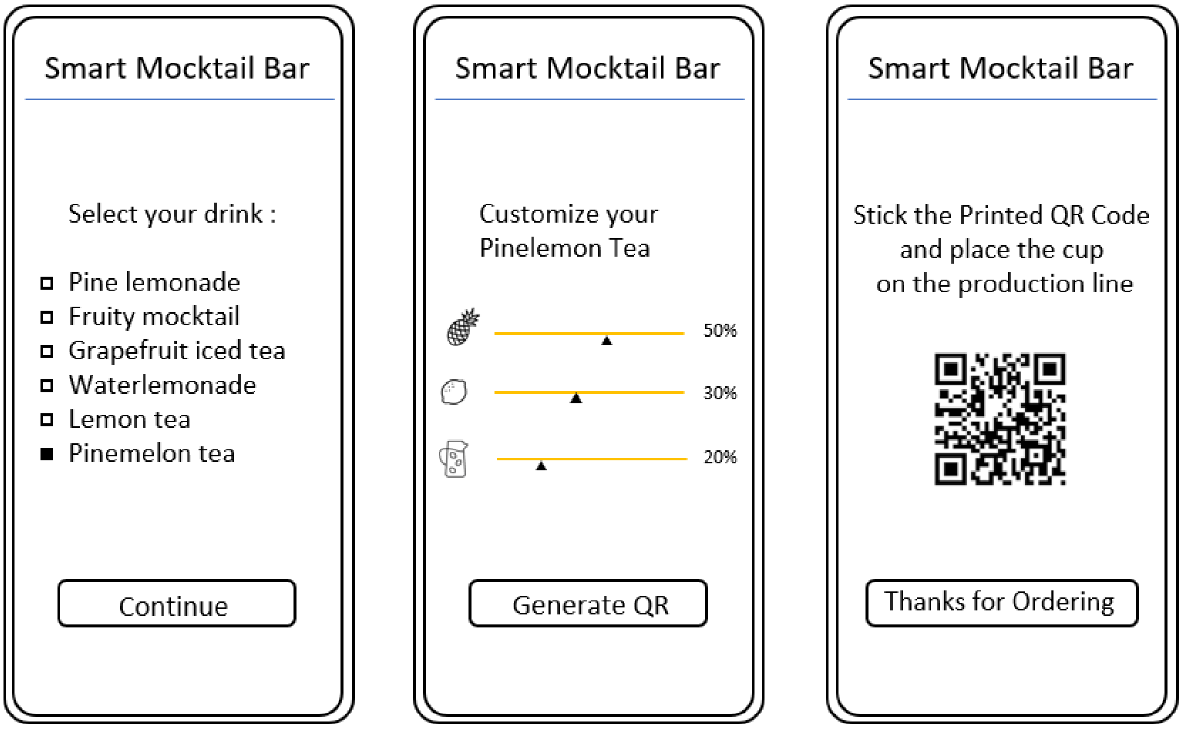

Power Supply
Switch
Load
Neutral
Line


Inputs
Controller
Control Input
Set Point
Logic
Actuator
System
Output
Measurement
Process Variable
Feedback Loop


Input 1
Input 2
Output 1
Output 1


Temperature_Sensor
22706
GT
IN1
OUT
IN2
Warning_Lamp




Smart Mocktail Bar

Select your drink :

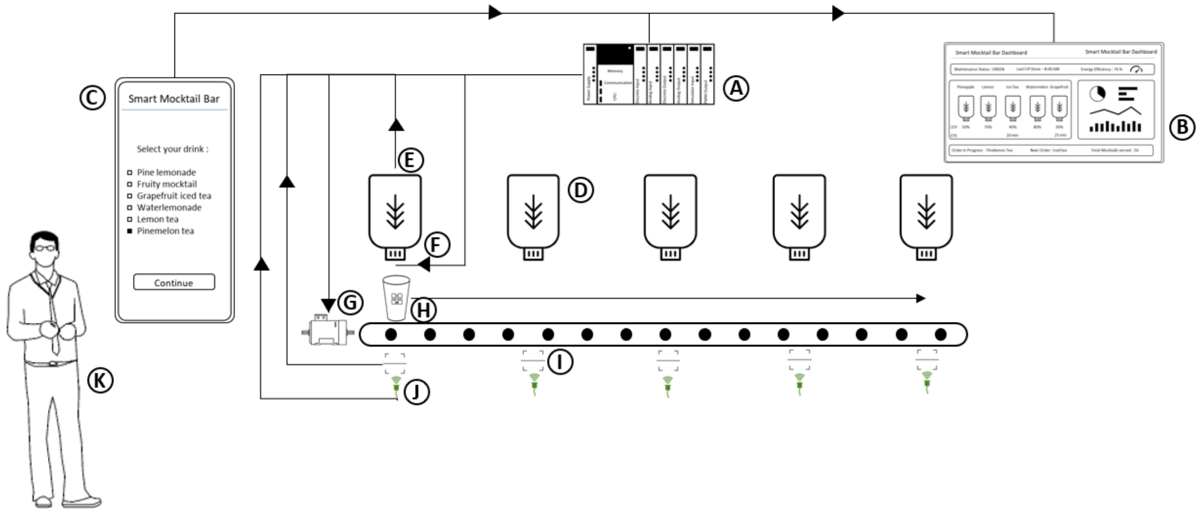
- □ Pine lemonade
- □ Fruity mocktail
- □ Grapefruit iced tea
- □ Waterlemonade
- □ Lemon tea
- ■ Pinemelon tea

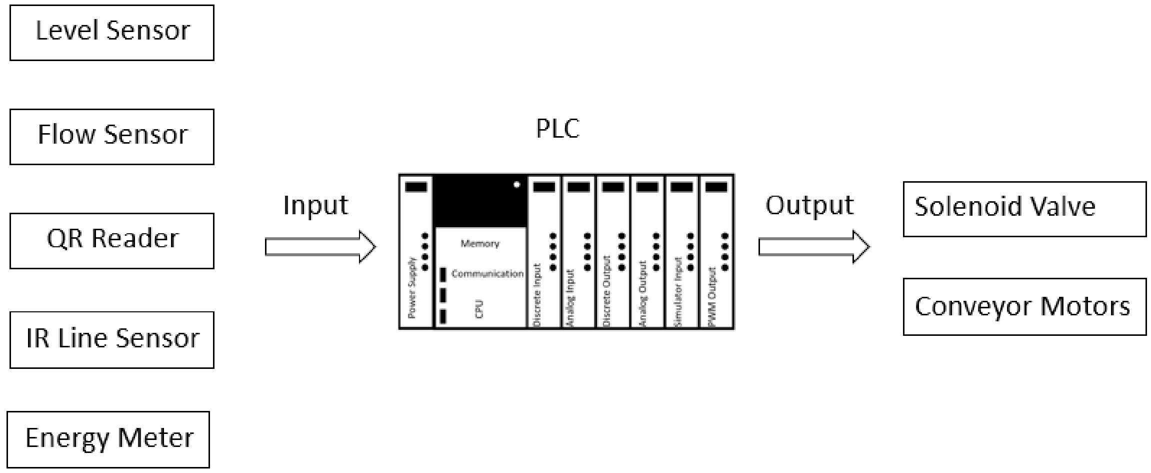
Smart Mocktail Bar

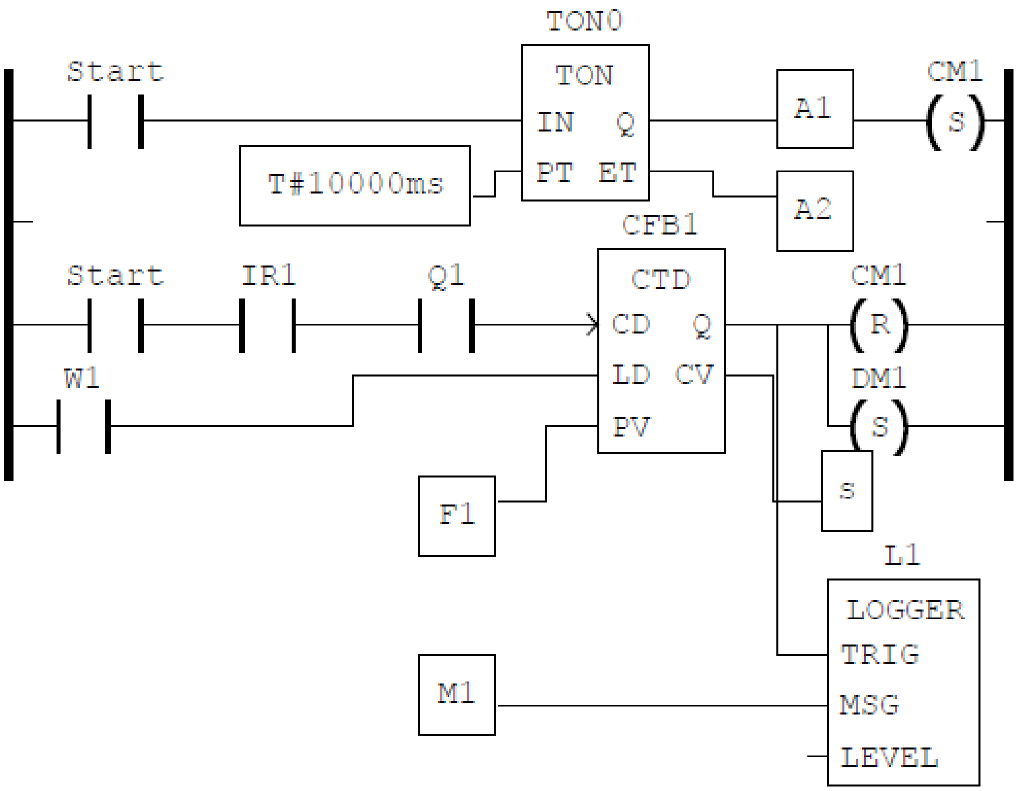
Customize your
Pinelemon Tea

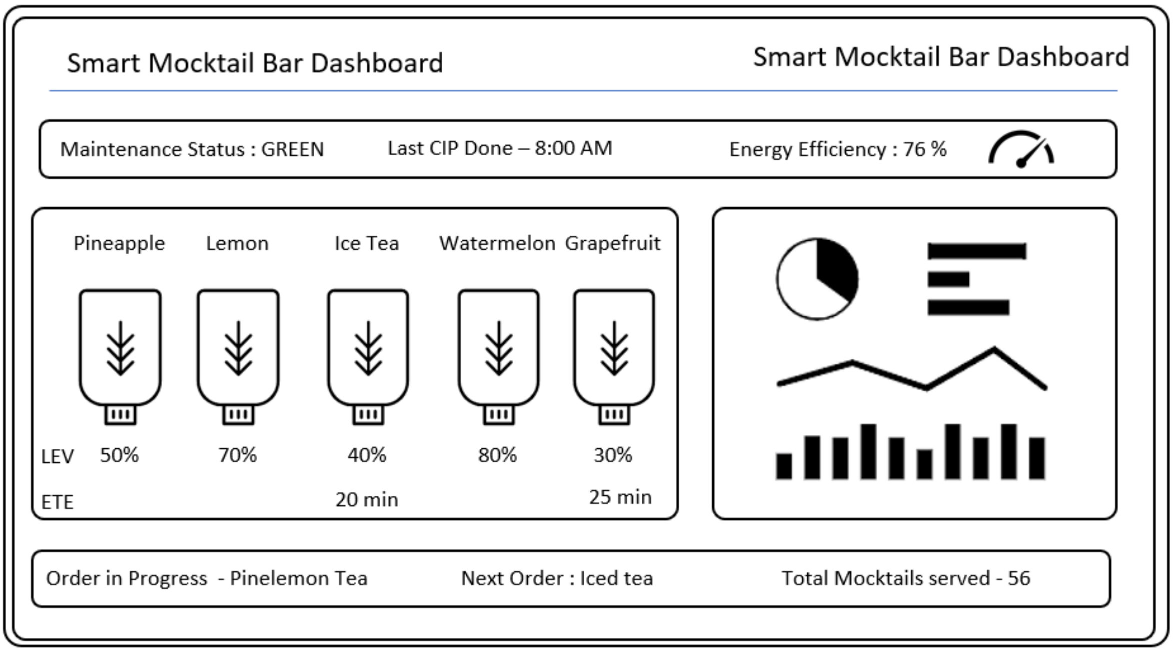
Smart Mocktail Bar

Stick the Printed QR Code
and place the cup
on the production line


A
B
C
D
E
F
G
H
I
J
K
Smart Mocktail Bar
Select your drink :
□ Pine lemonade
□ Fruity mocktail
□ Grapefruit iced tea
□ Waterlemonade
□ Lemon tea
■ Pinemelon tea
Continue
Smart Mocktail Bar Dashboard
Smart Mocktail Bar Dashboard
Maintenance Status : GREEN
Last CIP Done – 8:00 AM
Energy Efficiency : 76 %
Pineapple
Lemon
Ice Tea
Watermelon
Grapefruit
Order in Progress : Pinelemon Tea
Next Order : Iced tea
Total Mocktails served : 56
Memory
Communication
CPU


Level Sensor
Flow Sensor
QR Reader
IR Line Sensor
Energy Meter
Input
PLC
Power Supply
Memory
Communication
CPU
Discrete Input
Analog Input
Discrete Output
Analog Output
Simulator Input
PWM Output
Output
Solenoid Valve
Conveyor Motors





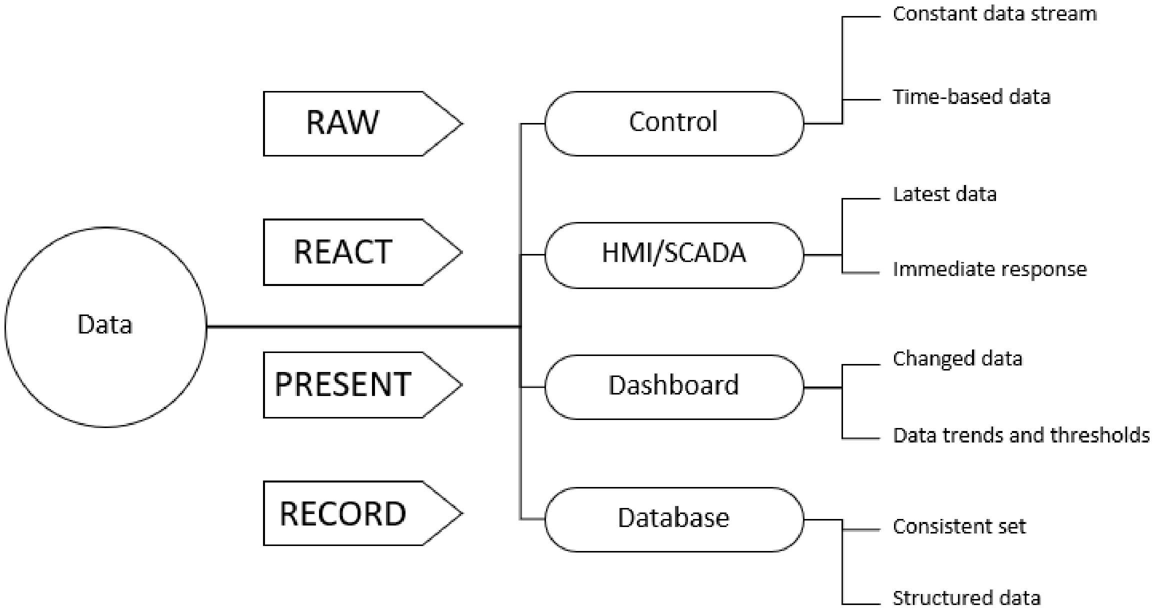
Smart Mocktail Bar Dashboard

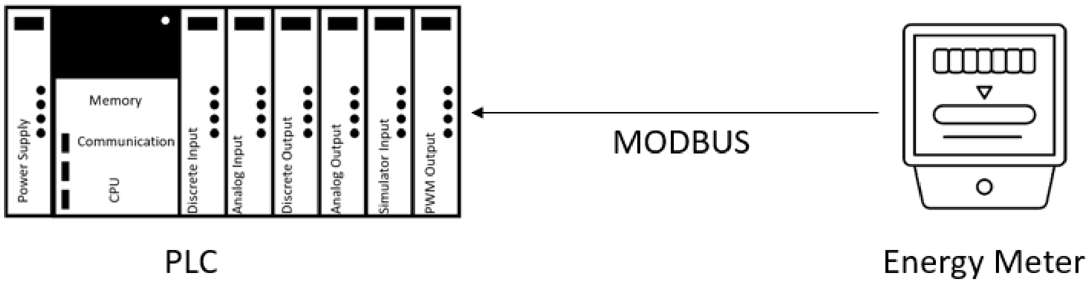
Smart Mocktail Bar Dashboard

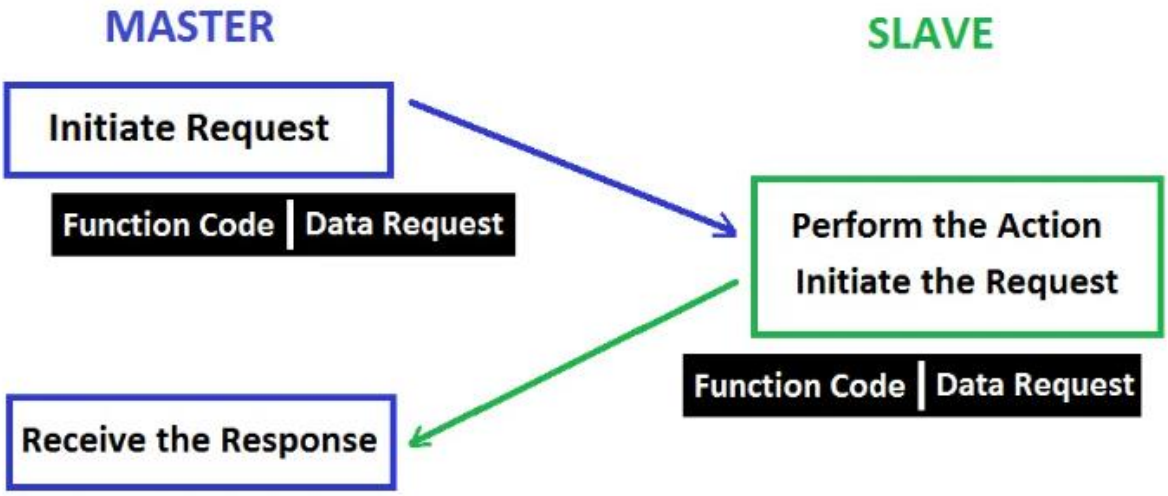
Maintenance Status : GREEN   Last CIP Done – 8:00 AM   Energy Efficiency : 76 %

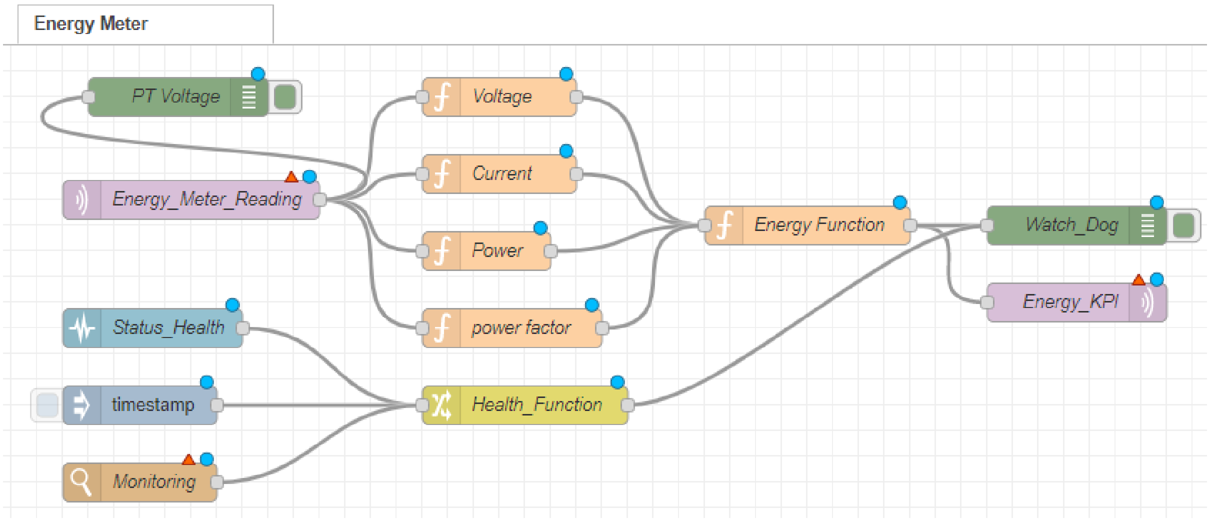
	Pineapple	Lemon	Ice Tea	Watermelon	Grapefruit
LEV	50%	70%	40%	80%	30%
ETE			20 min		25 min

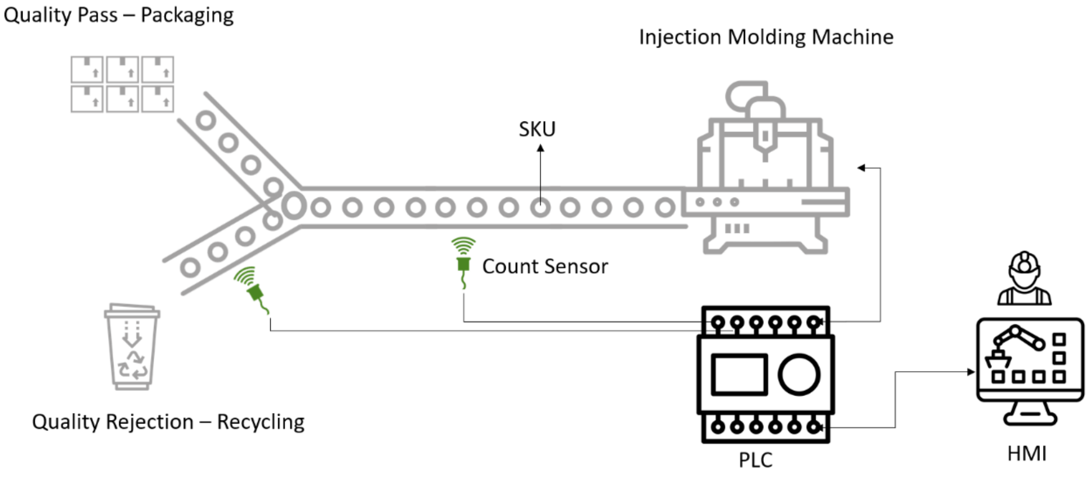
Order in Progress - Pinelemon Tea   Next Order : Iced tea   Total Mocktails served - 56

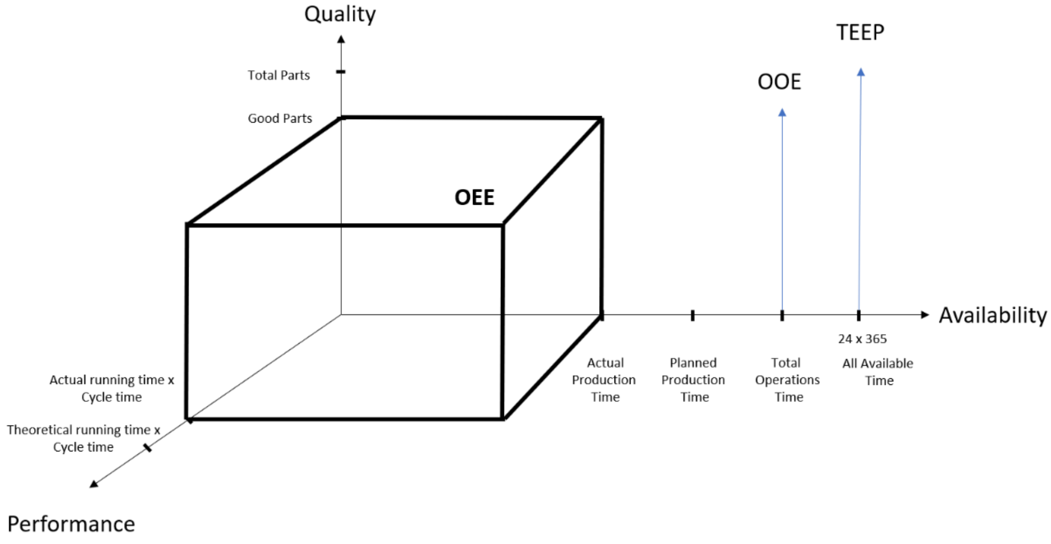

Data
RAW
REACT
PRESENT
RECORD
Control
HMI/SCADA
Dashboard
Database
Constant data stream
Time-based data
Latest data
Immediate response
Changed data
Data trends and thresholds
Consistent set
Structured data

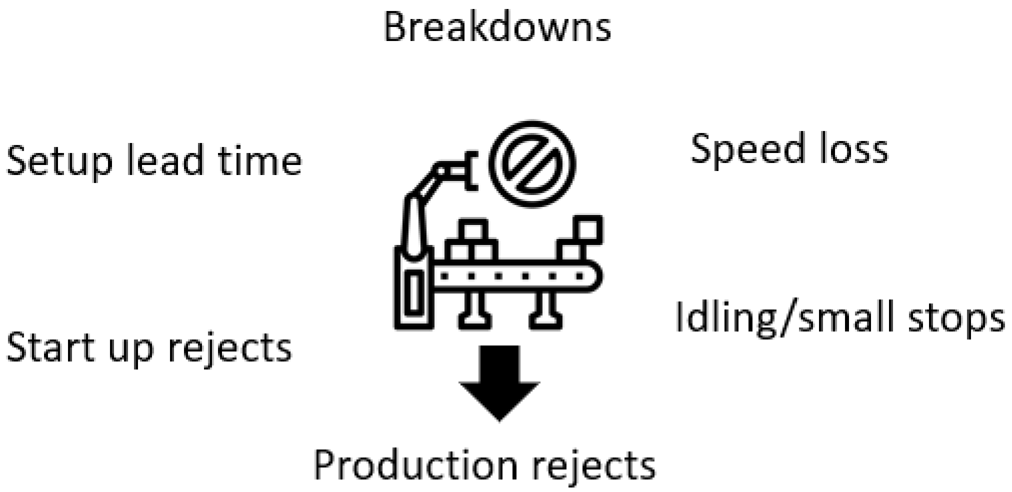

Power Supply
Memory
Communication
CPU
Discrete Input
Analog Input
Discrete Output
Analog Output
Simulator Input
PWM Output
MODBUS
PLC
Energy Meter


MASTER
SLAVE
Initiate Request
Function Code
Data Request
Perform the Action
Initiate the Request
Function Code
Data Request
Receive the Response


Energy Meter
PT Voltage
Energy_Meter_Reading
Voltage
Current
Power
power factor
Energy Function
Watch_Dog
Energy_KPI
Status_Health
timestamp
Monitoring
Health_Function


Quality Pass – Packaging
Injection Molding Machine
SKU
Count Sensor
Quality Rejection – Recycling
PLC
HMI


Quality
Total Parts
Good Parts
OEE
TEEP
OOE
Availability
24 x 365
Actual Production Time
Planned Production Time
Total Operations Time
All Available Time
Actual running time x Cycle time
Theoretical running time x Cycle time
Performance



Breakdowns

Setup lead time

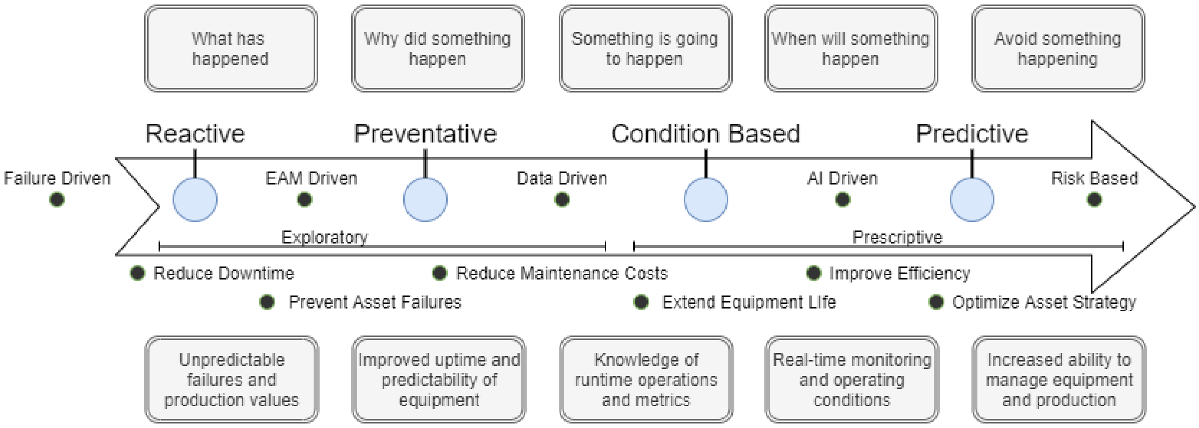
Speed loss

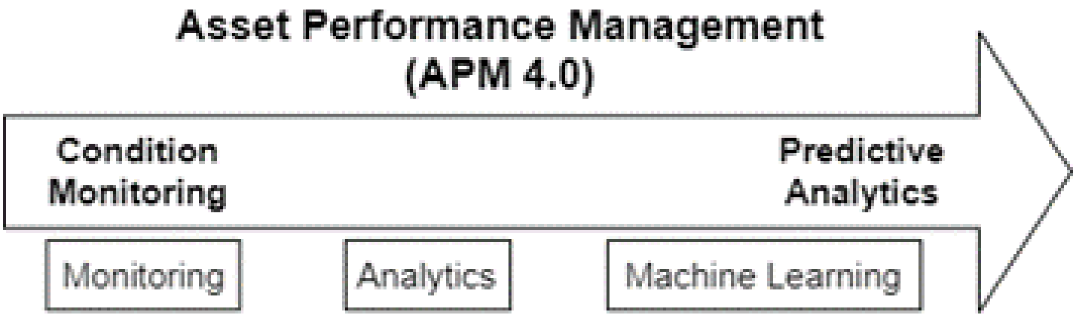
Start up rejects

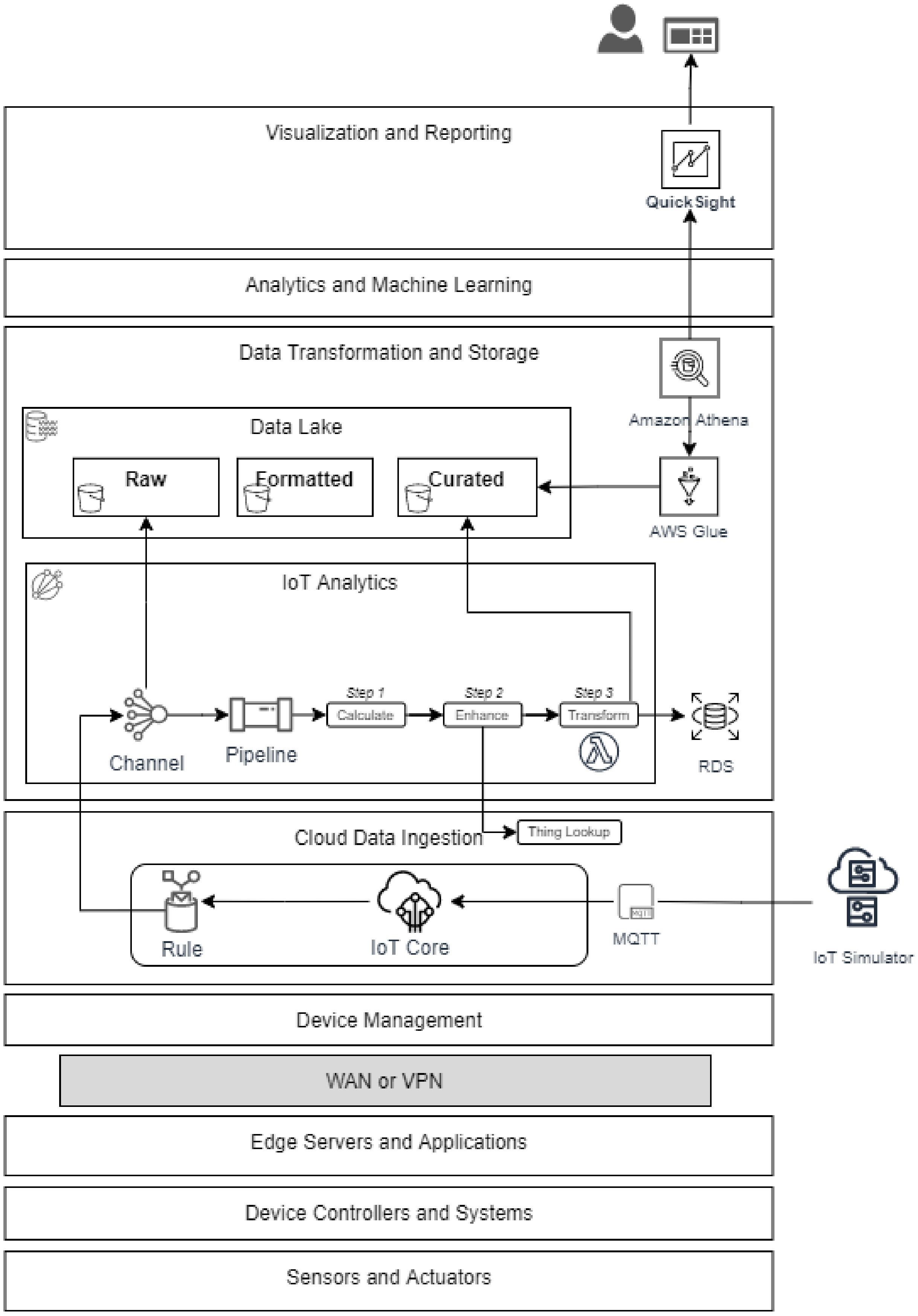
Idling/small stops

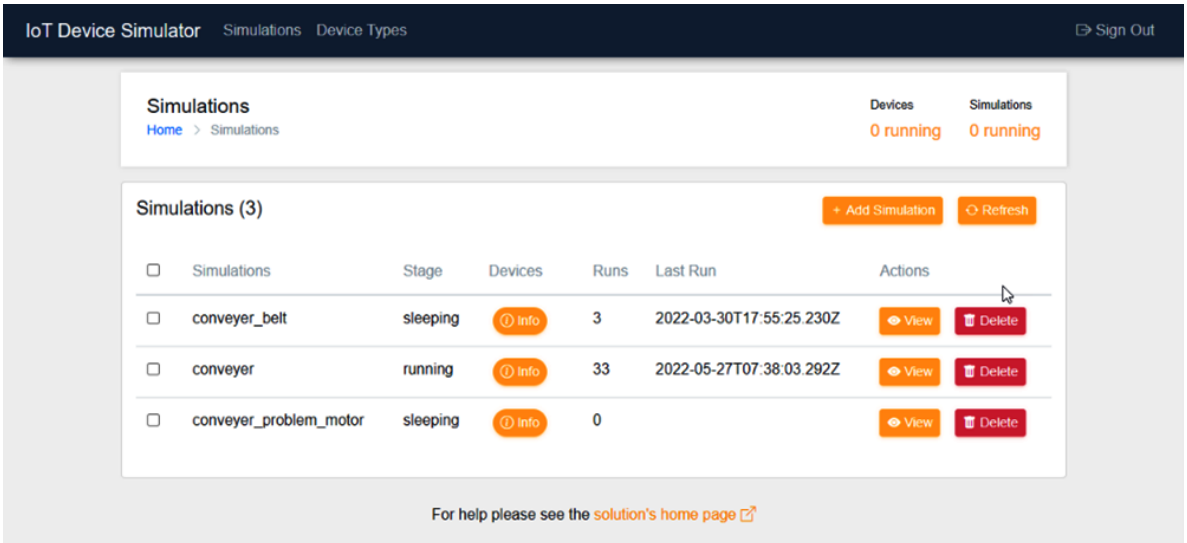
Production rejects

Chapter 8: Asset and Condition Monitoring


What has happened
Why did something happen
Something is going to happen
When will something happen
Avoid something happening
Reactive
Preventative
Condition Based
Predictive
Failure Driven
EAM Driven
Data Driven
AI Driven
Risk Based
Exploratory
Prescriptive
Reduce Downtime
Reduce Maintenance Costs
Improve Efficiency
Prevent Asset Failures
Extend Equipment Life
Optimize Asset Strategy
Unpredictable failures and production values
Improved uptime and predictability of equipment
Knowledge of runtime operations and metrics
Real-time monitoring and operating conditions
Increased ability to manage equipment and production




Visualization and Reporting
Quick Sight
Analytics and Machine Learning
Data Transformation and Storage
Amazon Athena
Data Lake
Raw
Formatted
Curated
AWS Glue
IoT Analytics
Step 1
Step 2
Step 3
Channel
Pipeline
Calculate
Enhance
Transform
RDS
Thing Lookup
Cloud Data Ingestion
Rule
IoT Core
MQTT
IoT Simulator
Device Management
WAN or VPN
Edge Servers and Applications
Device Controllers and Systems
Sensors and Actuators



Simulations

Home > Simulations

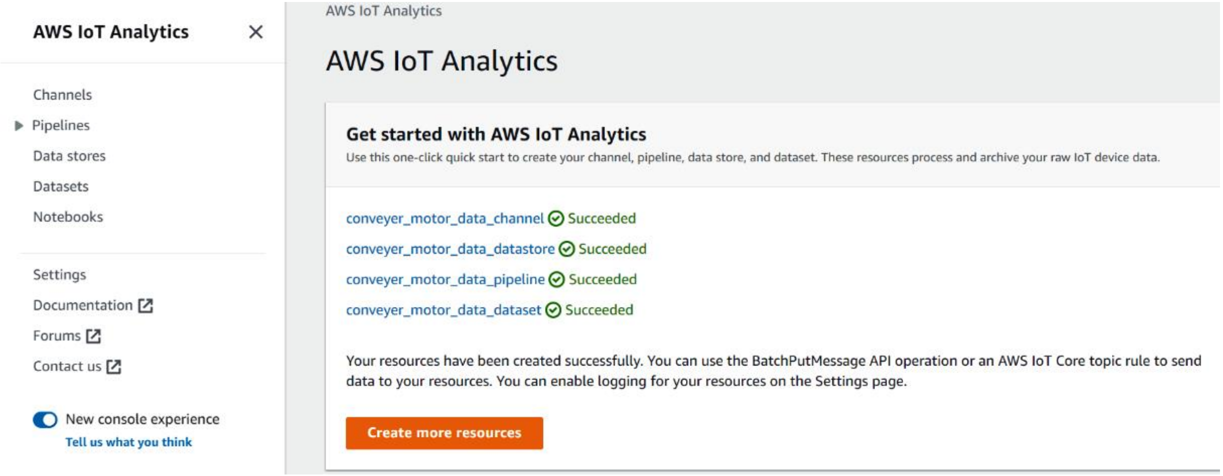
Devices 0 running

Simulations 0 running

Simulations (3)

□	Simulations	Stage	Devices	Runs	Last Run	Actions
□	conveyer_belt	sleeping	Info	3	2022-03-30T17:55:25.230Z	View Delete
□	conveyer	running	Info	33	2022-05-27T07:38:03.292Z	View Delete
□	conveyer_problem_motor	sleeping	Info	0		View Delete

For help please see the solution's home page



AWS IoT Analytics

AWS IoT Analytics

Get started with AWS IoT Analytics

Use this one-click quick start to create your channel, pipeline, data store, and dataset. These resources process and archive your raw IoT device data.

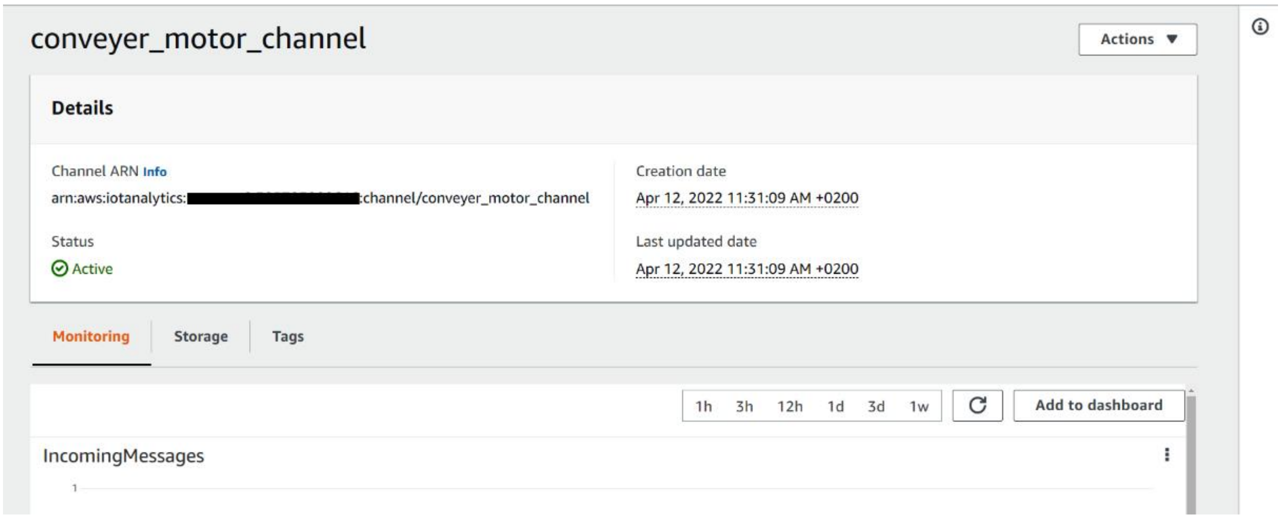
conveyer_motor_data_channel ✓ Succeeded

conveyer_motor_data_datastore ✓ Succeeded

conveyer_motor_data_pipeline ✓ Succeeded

conveyer_motor_data_dataset ✓ Succeeded

Your resources have been created successfully. You can use the BatchPutMessage API operation or an AWS IoT Core topic rule to send data to your resources. You can enable logging for your resources on the Settings page.



conveyer_motor_channel

Details

Channel ARN Info

arn:aws:iotanalytics:[redacted]:channel/conveyer_motor_channel

Status

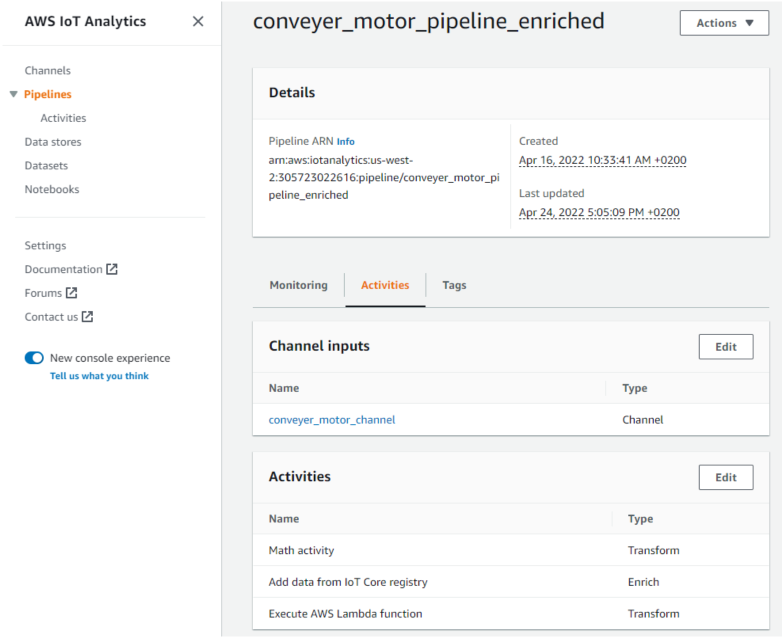
✓ Active

Creation date

Apr 12, 2022 11:31:09 AM +0200

Last updated date

Apr 12, 2022 11:31:09 AM +0200



conveyer_motor_pipeline_enriched

Details

Pipeline ARN Info
arn:aws:iotanalytics:us-west-2:305723022616:pipeline/conveyer_motor_pipeline_enriched

Created
Apr 16, 2022 10:33:41 AM +0200

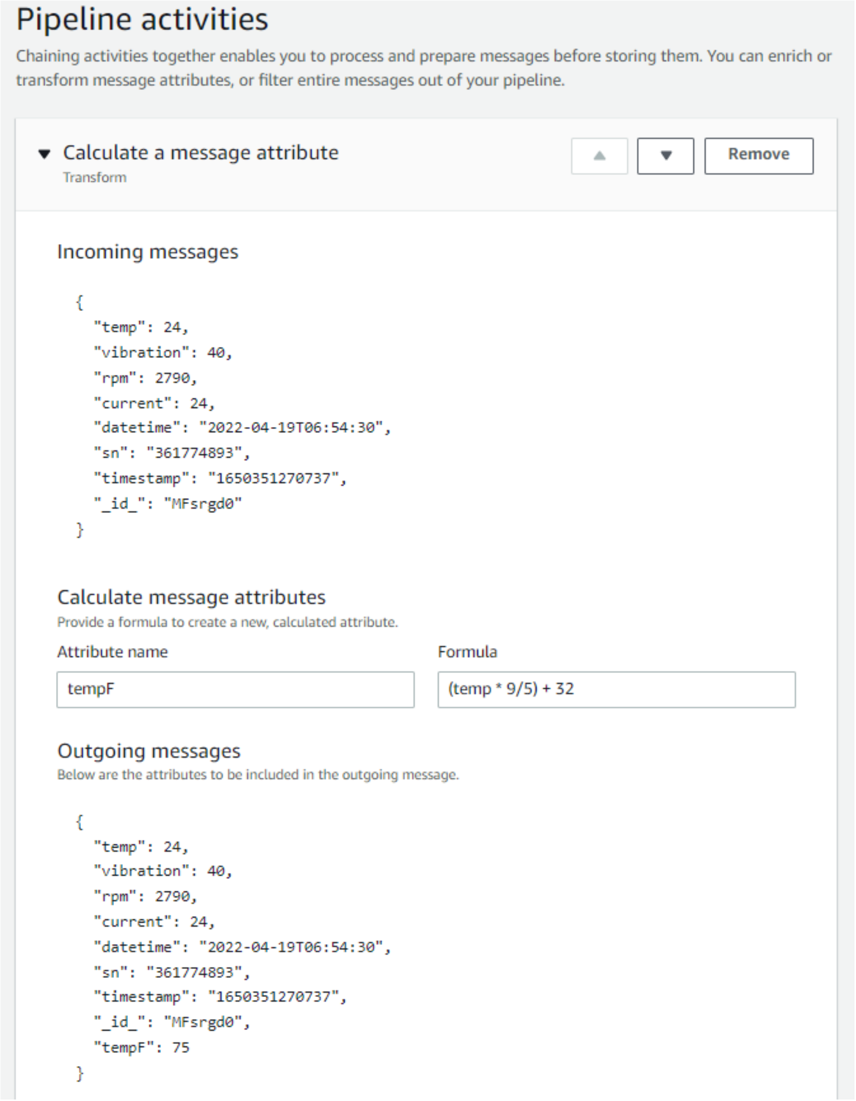
Last updated
Apr 24, 2022 5:05:09 PM +0200

Channel inputs

Name	Type
conveyer_motor_channel	Channel

Activities

Name	Type
Math activity	Transform
Add data from IoT Core registry	Enrich
Execute AWS Lambda function	Transform



Pipeline activities

Chaining activities together enables you to process and prepare messages before storing them. You can enrich or transform message attributes, or filter entire messages out of your pipeline.

Calculate a message attribute
Transform

Incoming messages

```
{
  "temp": 24,
  "vibration": 40,
  "rpm": 2790,
  "current": 24,
  "datetime": "2022-04-19T06:54:30",
  "sn": "361774893",
  "timestamp": "1650351270737",
  "_id_": "MFsrgd0"
}
```

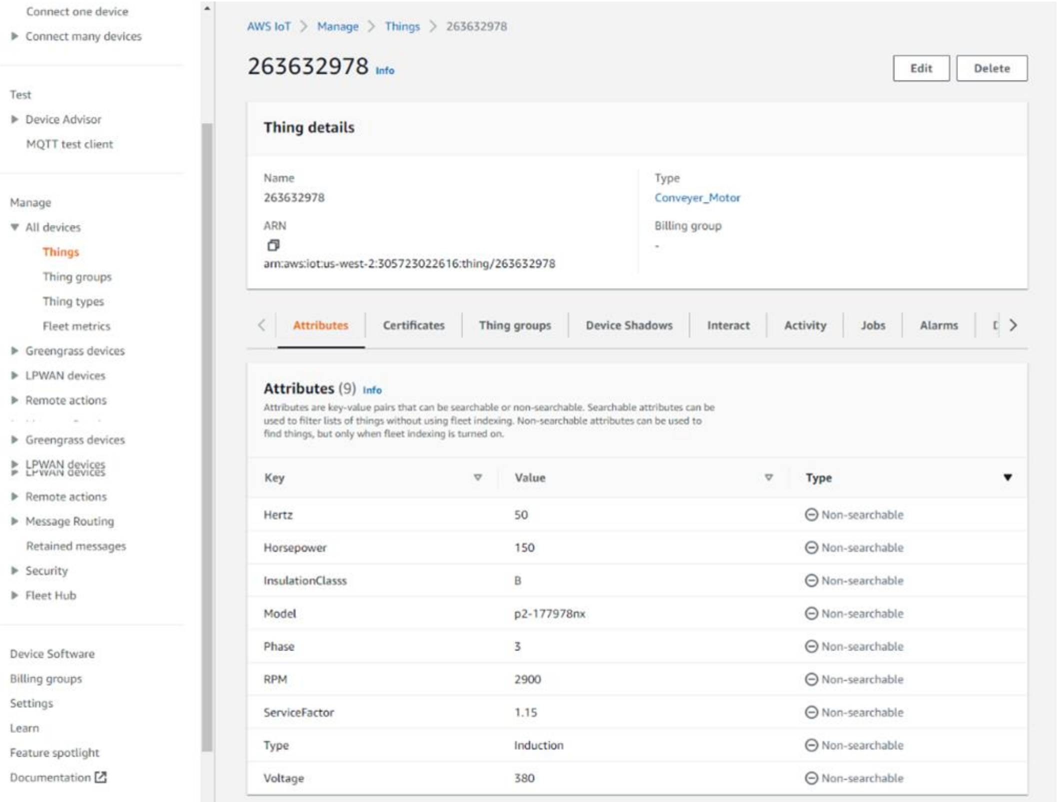
Calculate message attributes
Provide a formula to create a new, calculated attribute.

Attribute name: tempF

Formula: (temp * 9/5) + 32

Outgoing messages
Below are the attributes to be included in the outgoing message.

```
{
  "temp": 24,
  "vibration": 40,
  "rpm": 2790,
  "current": 24,
  "datetime": "2022-04-19T06:54:30",
  "sn": "361774893",
  "timestamp": "1650351270737",
  "_id_": "MFsrgd0",
  "tempF": 75
}
```



263632978 Info

Thing details

Name
263632978

ARN
arn:aws:iot:us-west-2:305723022616:thing/263632978

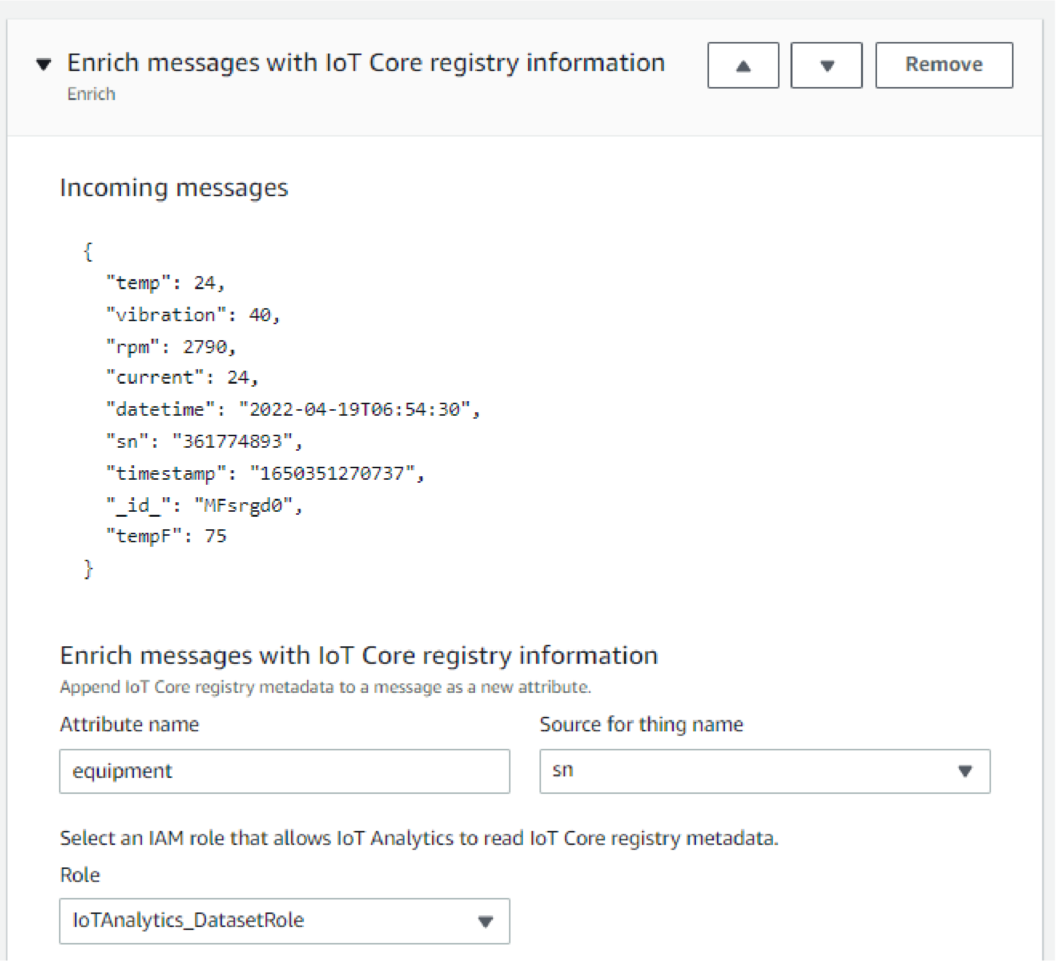
Type
Conveyer_Motor

Billing group
-

Attributes (9) Info

Attributes are key-value pairs that can be searchable or non-searchable. Searchable attributes can be used to filter lists of things without using fleet indexing. Non-searchable attributes can be used to find things, but only when fleet indexing is turned on.

Key	Value	Type
Hertz	50	⊖ Non-searchable
Horsepower	150	⊖ Non-searchable
InsulationClasss	B	⊖ Non-searchable
Model	p2-177978nx	⊖ Non-searchable
Phase	3	⊖ Non-searchable
RPM	2900	⊖ Non-searchable
ServiceFactor	1.15	⊖ Non-searchable
Type	Induction	⊖ Non-searchable
Voltage	380	⊖ Non-searchable



Enrich messages with IoT Core registry information

Enrich

Incoming messages

```
{
  "temp": 24,
  "vibration": 40,
  "rpm": 2790,
  "current": 24,
  "datetime": "2022-04-19T06:54:30",
  "sn": "361774893",
  "timestamp": "1650351270737",
  "_id_": "MFsrgd0",
  "tempF": 75
}
```

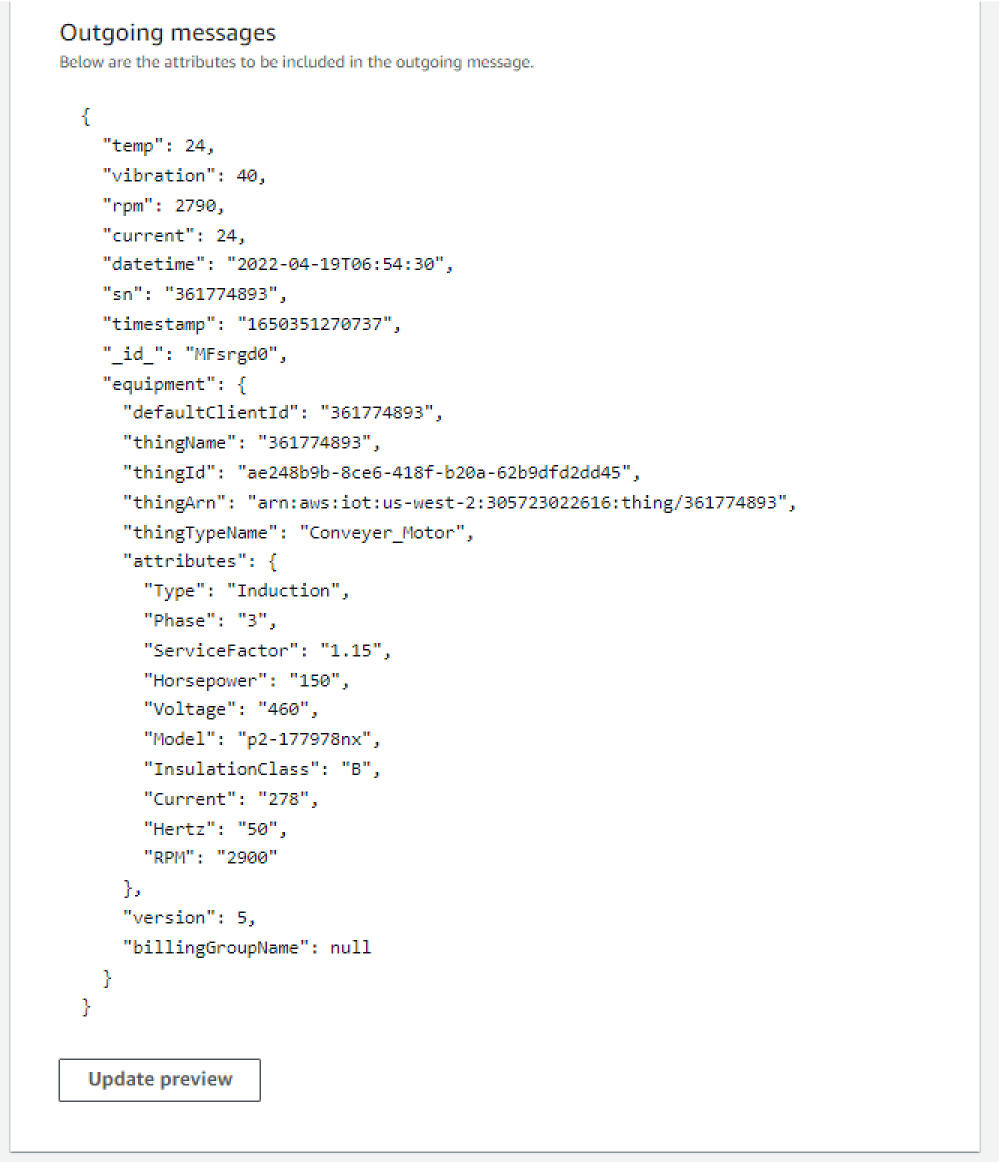
Enrich messages with IoT Core registry information

Append IoT Core registry metadata to a message as a new attribute.

Attribute name	Source for thing name
equipment	sn

Select an IAM role that allows IoT Analytics to read IoT Core registry metadata.

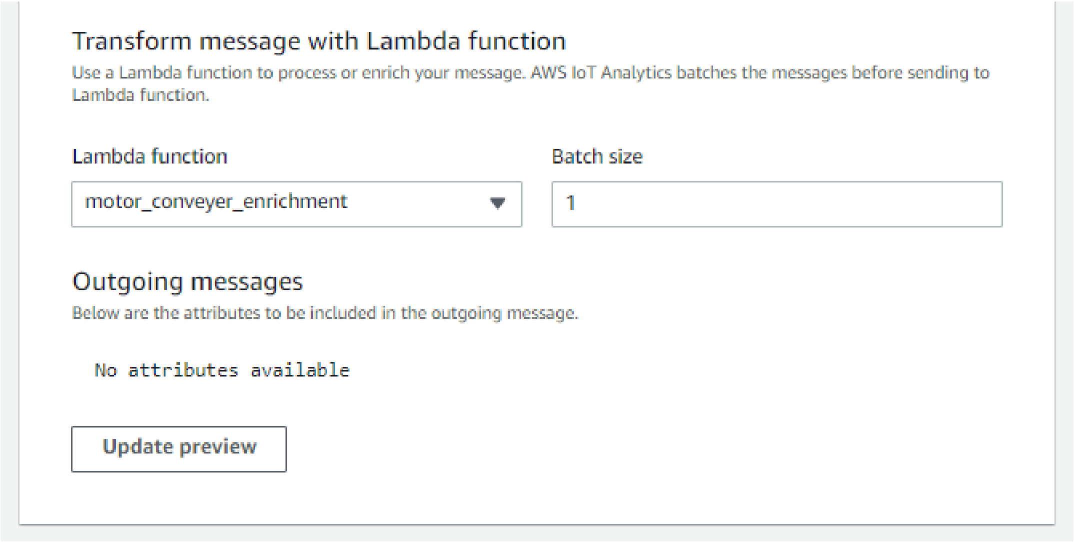
Role
IoTAnalytics_DatasetRole



Outgoing messages

Below are the attributes to be included in the outgoing message.

```
{
  "temp": 24,
  "vibration": 40,
  "rpm": 2790,
  "current": 24,
  "datetime": "2022-04-19T06:54:30",
  "sn": "361774893",
  "timestamp": "1650351270737",
  "_id_": "MFsrgd0",
  "equipment": {
    "defaultClientId": "361774893",
    "thingName": "361774893",
    "thingId": "ae248b9b-8ce6-418f-b20a-62b9dfd2dd45",
    "thingArn": "arn:aws:iot:us-west-2:305723022616:thing/361774893",
    "thingTypeName": "Conveyer_Motor",
    "attributes": {
      "Type": "Induction",
      "Phase": "3",
      "ServiceFactor": "1.15",
      "Horsepower": "150",
      "Voltage": "460",
      "Model": "p2-177978nx",
      "InsulationClass": "B",
      "Current": "278",
      "Hertz": "50",
      "RPM": "2900"
    },
    "version": 5,
    "billingGroupName": null
  }
}
```



Transform message with Lambda function

Use a Lambda function to process or enrich your message. AWS IoT Analytics batches the messages before sending to Lambda function.

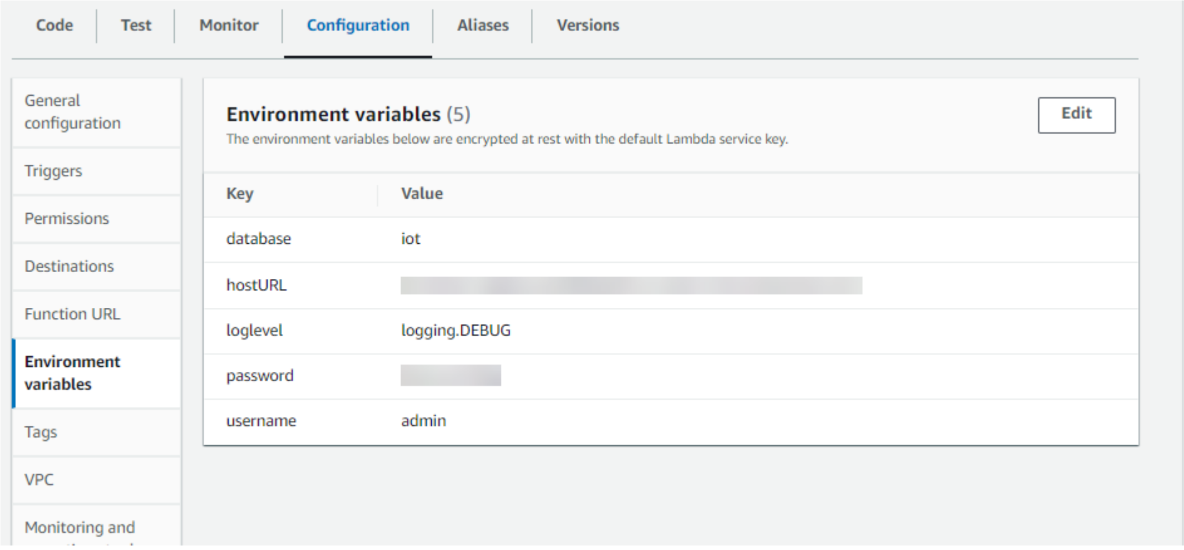
Lambda function: motor_conveyer_enrichment

Batch size: 1

Outgoing messages

Below are the attributes to be included in the outgoing message.

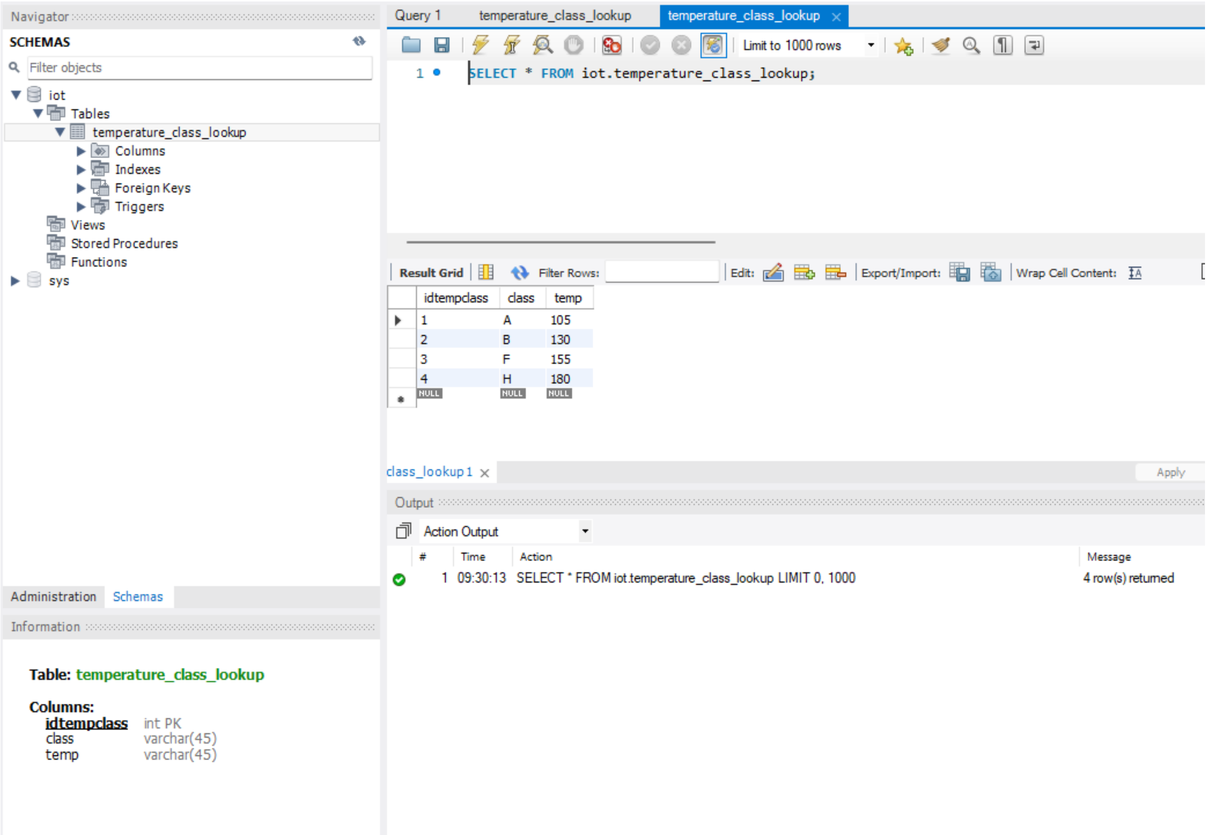
```
No attributes available
```



Environment variables (5)

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
database	iot
hostURL	[redacted]
loglevel	logging.DEBUG
password	[redacted]
username	admin



```
SELECT * FROM iot.temperature_class_lookup;
```

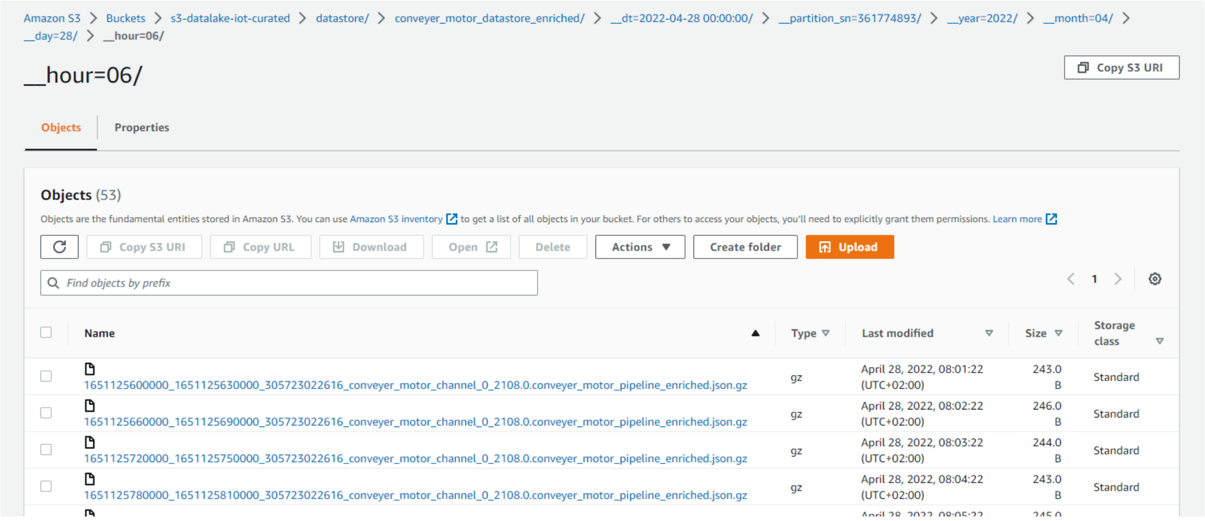
idtempclass	class	temp
1	A	105
2	B	130
3	F	155
4	H	180
NULL	NULL	NULL

#	Time	Action	Message
1	09:30:13	SELECT * FROM iot.temperature_class_lookup LIMIT 0, 1000	4 row(s) returned

Table: temperature_class_lookup

Columns:

idtempclass	int PK
class	varchar(45)
temp	varchar(45)



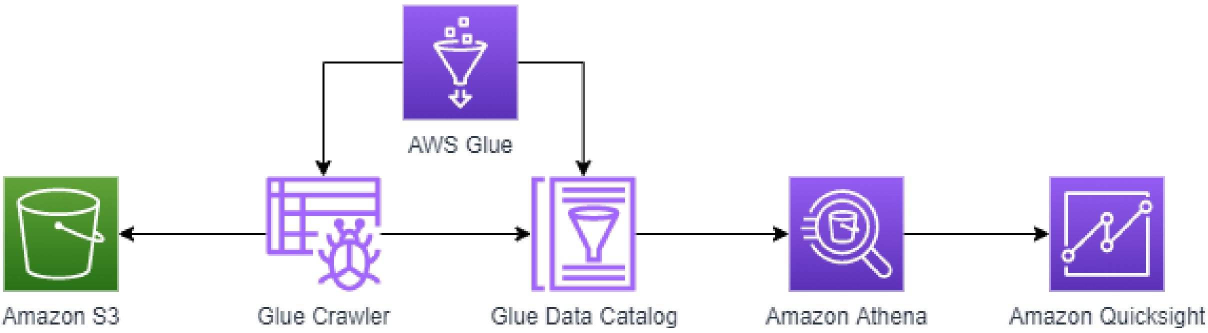
Amazon S3 > Buckets > s3-datalake-iot-curated > datastore/ > conveyer_motor_datastore_enriched/ > __dt=2022-04-28 00:00:00/ > __partition_sn=361774893/ > __year=2022/ > __month=04/ > __day=28/ > __hour=06/

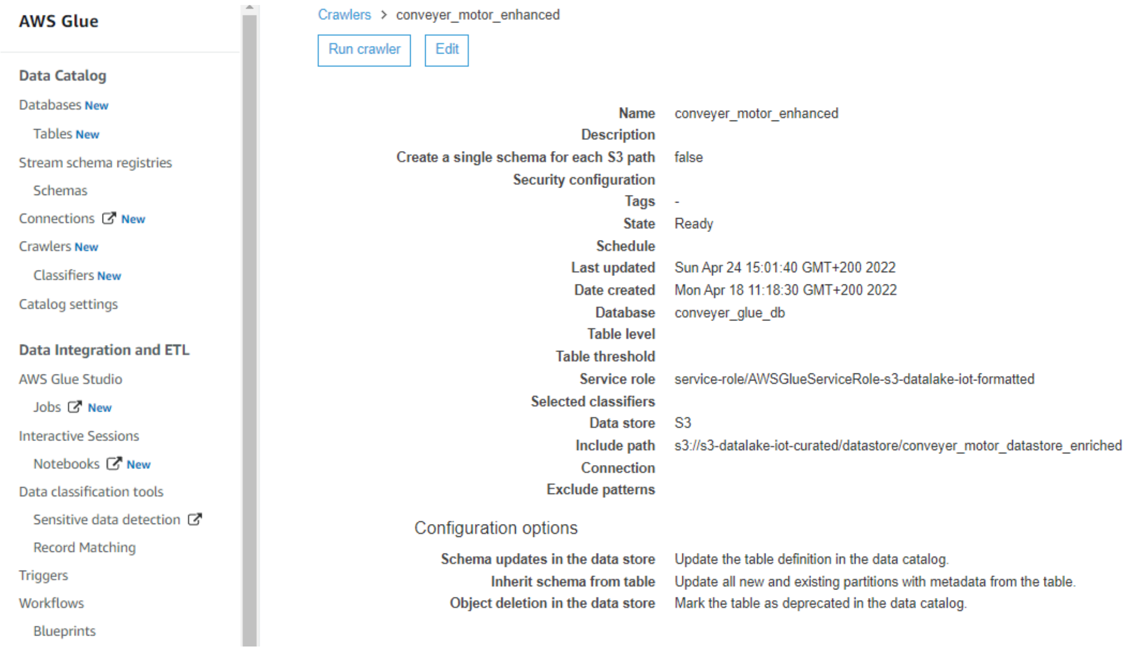
__hour=06/

Objects (53)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	1651125600000_1651125630000_305723022616_conveyer_motor_channel_0_2108.0.conveyer_motor_pipeline_enriched.json.gz	gz	April 28, 2022, 08:01:22 (UTC+02:00)	243.0 B	Standard
□	1651125660000_1651125690000_305723022616_conveyer_motor_channel_0_2108.0.conveyer_motor_pipeline_enriched.json.gz	gz	April 28, 2022, 08:02:22 (UTC+02:00)	246.0 B	Standard
□	1651125720000_1651125750000_305723022616_conveyer_motor_channel_0_2108.0.conveyer_motor_pipeline_enriched.json.gz	gz	April 28, 2022, 08:03:22 (UTC+02:00)	244.0 B	Standard
□	1651125780000_1651125810000_305723022616_conveyer_motor_channel_0_2108.0.conveyer_motor_pipeline_enriched.json.gz	gz	April 28, 2022, 08:04:22 (UTC+02:00)	243.0 B	Standard



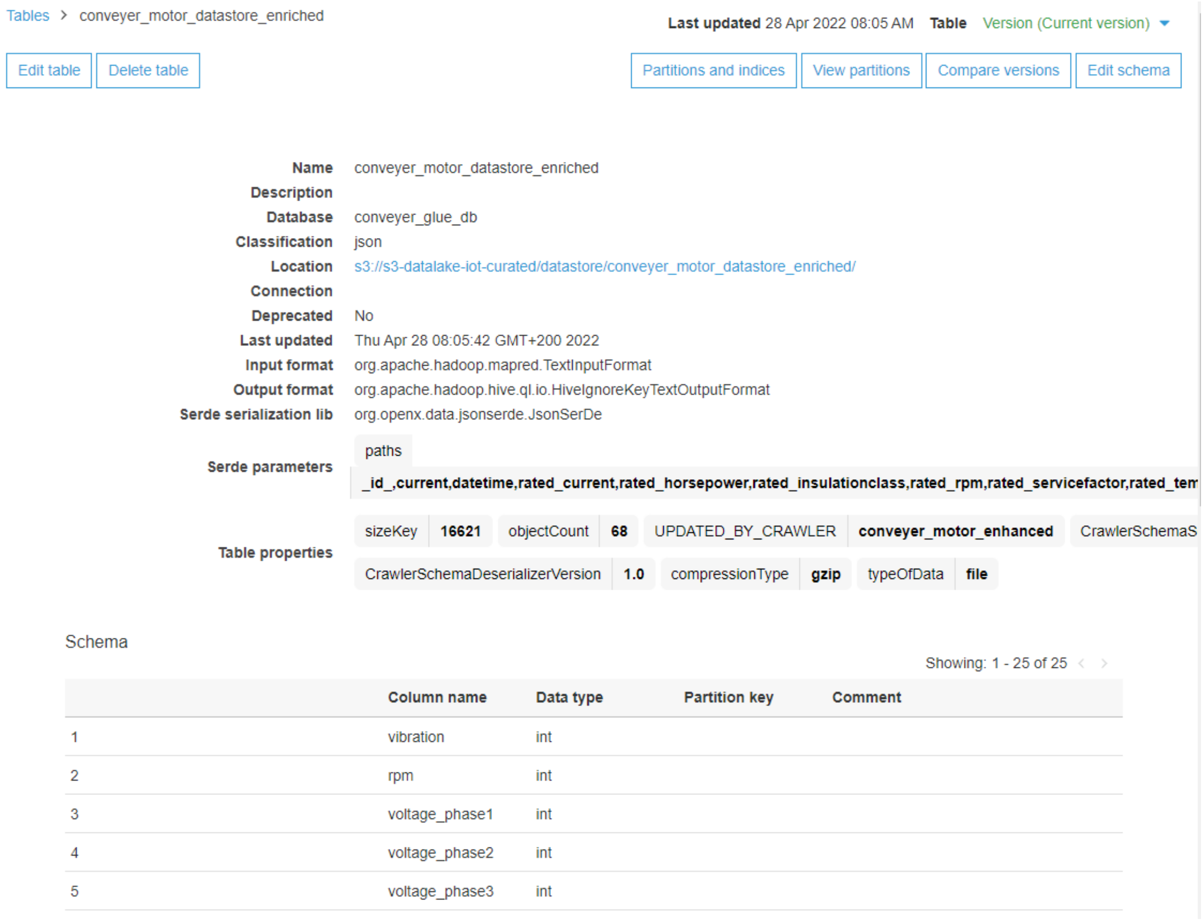


Crawlers > conveyer_motor_enhanced

Name	conveyer_motor_enhanced
Description	
Create a single schema for each S3 path	false
Security configuration	
Tags	-
State	Ready
Schedule	
Last updated	Sun Apr 24 15:01:40 GMT+200 2022
Date created	Mon Apr 18 11:18:30 GMT+200 2022
Database	conveyer_glue_db
Table level	
Table threshold	
Service role	service-role/AWSGlueServiceRole-s3-datalake-iot-formatted
Selected classifiers	
Data store	S3
Include path	s3://s3-datalake-iot-curated/datastore/conveyer_motor_datastore_enriched
Connection	
Exclude patterns	

Configuration options

Schema updates in the data store	Update the table definition in the data catalog.
Inherit schema from table	Update all new and existing partitions with metadata from the table.
Object deletion in the data store	Mark the table as deprecated in the data catalog.



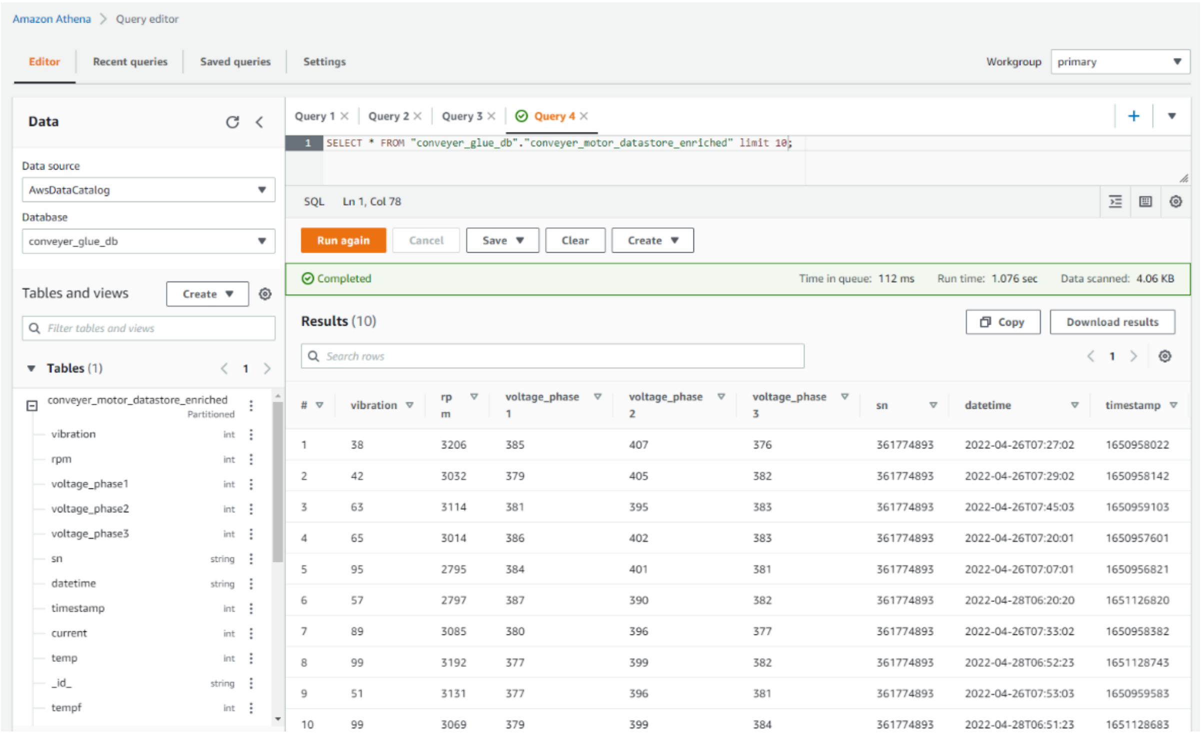
Last updated 28 Apr 2022 08:05 AM **Table** Version (Current version)

Name	conveyer_motor_datastore_enriched
Description	
Database	conveyer_glue_db
Classification	json
Location	s3://s3-datalake-iot-curated/datastore/conveyer_motor_datastore_enriched/
Connection	
Deprecated	No
Last updated	Thu Apr 28 08:05:42 GMT+200 2022
Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.openx.data.jsonserde.JsonSerDe
Serde parameters	paths **_id_,current,datetime,rated_current,rated_horsepower,rated_insulationclass,rated_rpm,rated_servicefactor,rated_ter**
Table properties	sizeKey **16621** objectCount **68** UPDATED_BY_CRAWLER **conveyer_motor_enhanced** CrawlerSchemaS CrawlerSchemaDeserializerVersion **1.0** compressionType **gzip** typeOfData **file**

Schema

Showing: 1 - 25 of 25

	Column name	Data type	Partition key	Comment
1	vibration	int		
2	rpm	int		
3	voltage_phase1	int		
4	voltage_phase2	int		
5	voltage_phase3	int		



Data source: AwsDataCatalog

Database: conveyer_glue_db

Tables and views

Tables (1)

conveyer_motor_datastore_enriched Partitioned

vibration	int
rpm	int
voltage_phase1	int
voltage_phase2	int
voltage_phase3	int
sn	string
datetime	string
timestamp	int
current	int
temp	int
_id_	string
tempf	int

```
SELECT * FROM "conveyer_glue_db"."conveyer_motor_datastore_enriched" limit 10;
```

SQL Ln 1, Col 78

Completed Time in queue: 112 ms Run time: 1.076 sec Data scanned: 4.06 KB

Results (10)

#	vibration	rpm	voltage_phase1	voltage_phase2	voltage_phase3	sn	datetime	timestamp
1	38	3206	385	407	376	361774893	2022-04-26T07:27:02	1650958022
2	42	3032	379	405	382	361774893	2022-04-26T07:29:02	1650958142
3	63	3114	381	395	383	361774893	2022-04-26T07:45:03	1650959103
4	65	3014	386	402	383	361774893	2022-04-26T07:20:01	1650957601
5	95	2795	384	401	381	361774893	2022-04-26T07:07:01	1650956821
6	57	2797	387	390	382	361774893	2022-04-28T06:20:20	1651126820
7	89	3085	380	396	377	361774893	2022-04-26T07:33:02	1650958382
8	99	3192	377	399	382	361774893	2022-04-28T06:52:23	1651128743
9	51	3131	377	396	381	361774893	2022-04-26T07:53:03	1650959583
10	99	3069	379	399	384	361774893	2022-04-28T06:51:23	1651128683

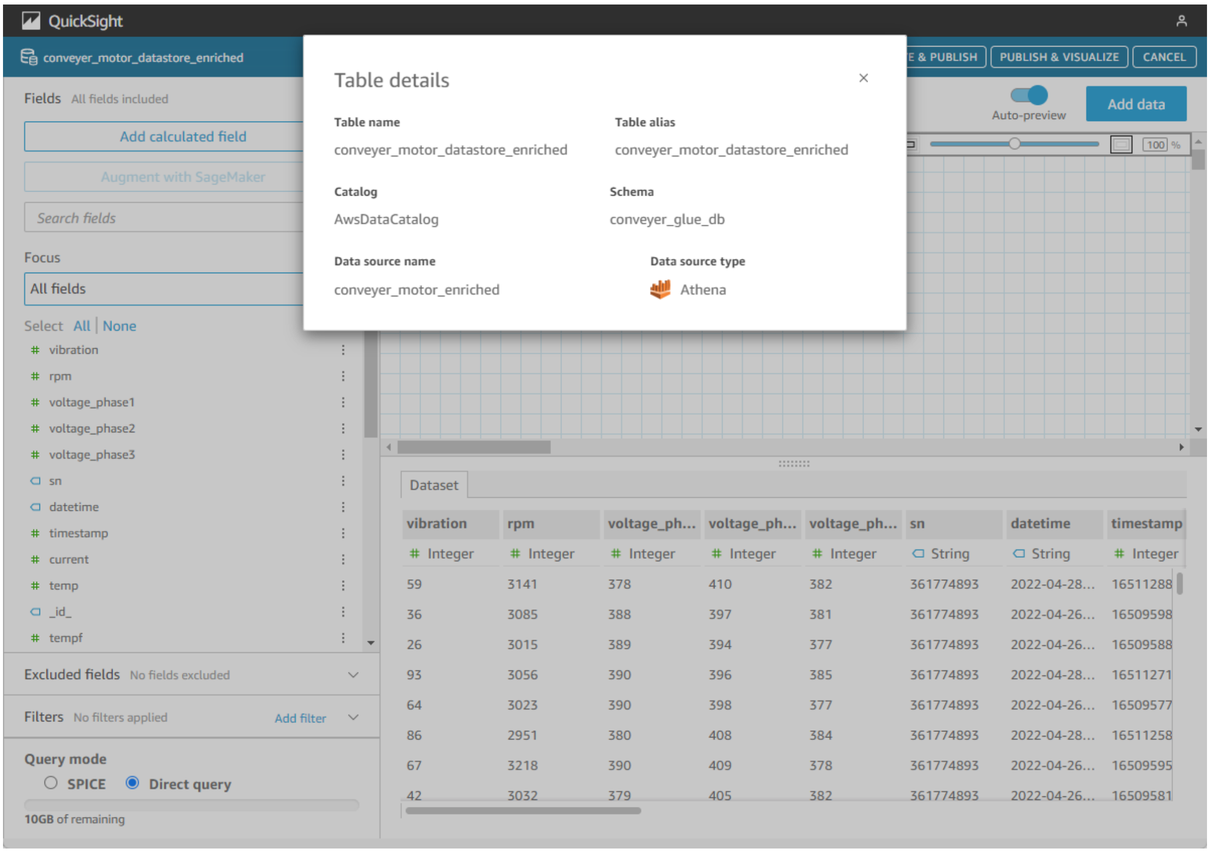


Table details

Table name
conveyer_motor_datastore_enriched

Table alias
conveyer_motor_datastore_enriched

Catalog
AwsDataCatalog

Schema
conveyer_glue_db

Data source name
conveyer_motor_enriched

Data source type
Athena

Fields All fields included

Focus
All fields

Select All | None

- # vibration
- # rpm
- # voltage_phase1
- # voltage_phase2
- # voltage_phase3
- sn
- datetime
- # timestamp
- # current
- # temp
- _id_
- # tempf

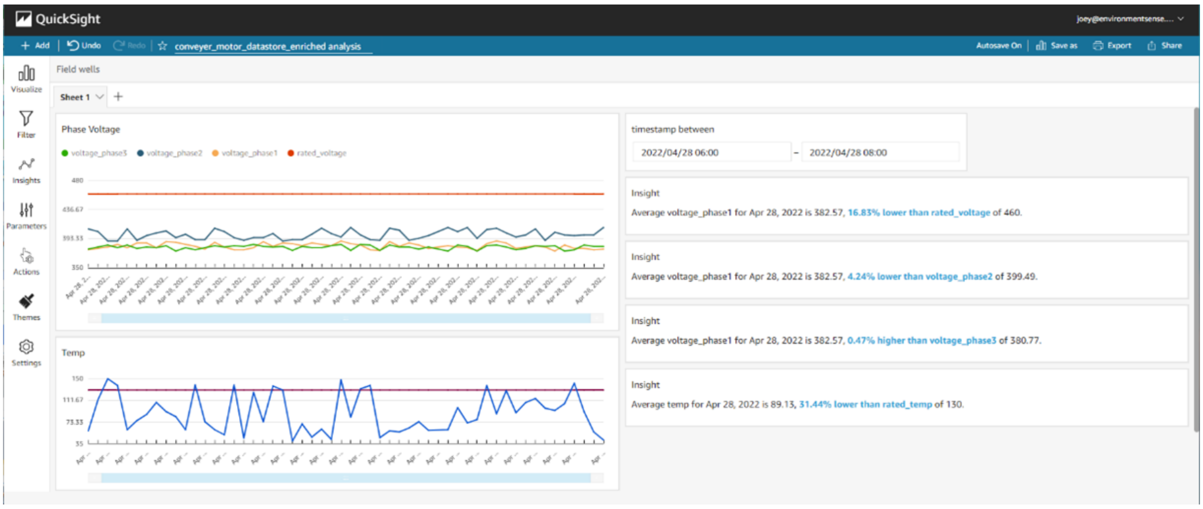
Excluded fields No fields excluded

Filters No filters applied

Query mode
○ SPICE ● Direct query

10GB of remaining

vibration	rpm	voltage_ph...	voltage_ph...	voltage_ph...	sn	datetime	timestamp
# Integer	# Integer	# Integer	# Integer	# Integer	String	String	# Integer
59	3141	378	410	382	361774893	2022-04-28...	16511288
36	3085	388	397	381	361774893	2022-04-26...	16509598
26	3015	389	394	377	361774893	2022-04-26...	16509588
93	3056	390	396	385	361774893	2022-04-28...	16511271
64	3023	390	398	377	361774893	2022-04-26...	16509577
86	2951	380	408	384	361774893	2022-04-28...	16511258
67	3218	390	409	378	361774893	2022-04-26...	16509595
42	3032	379	405	382	361774893	2022-04-26...	16509581



timestamp between
2022/04/28 06:00 – 2022/04/28 08:00

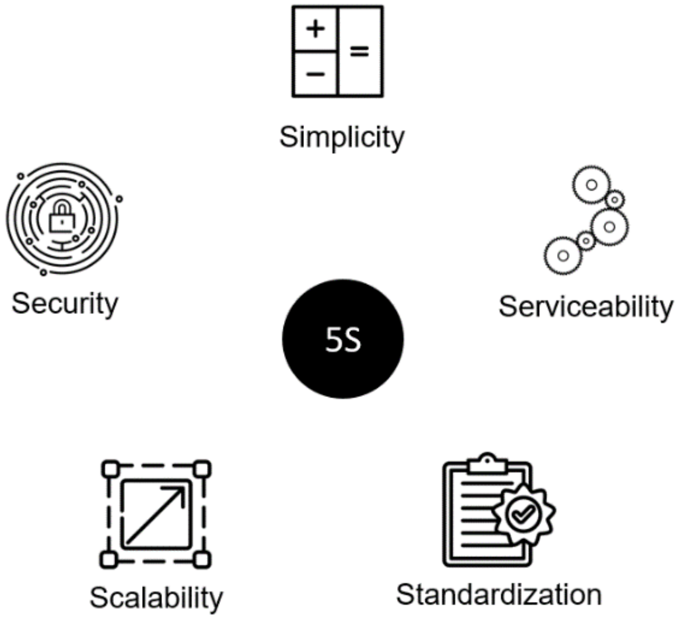
Insight
Average voltage_phase1 for Apr 28, 2022 is 382.57, 16.83% lower than rated_voltage of 460.

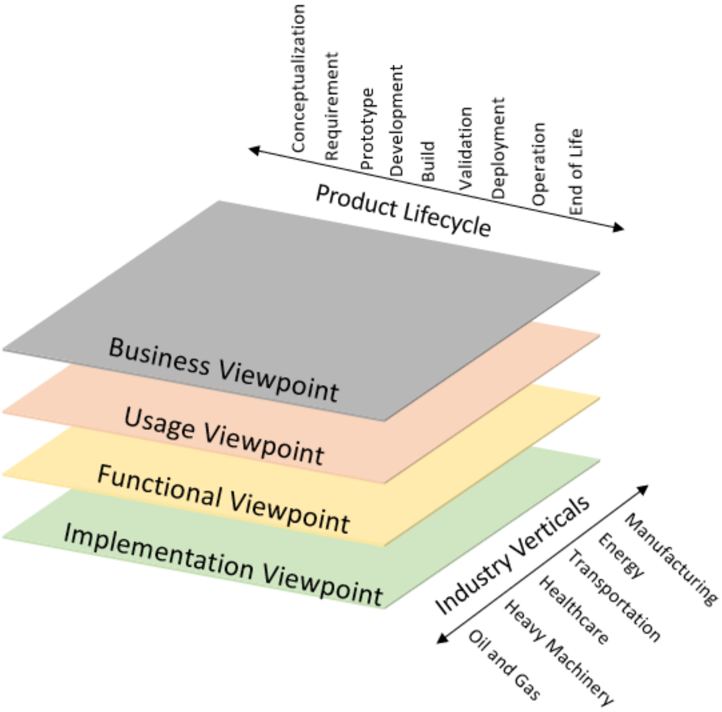
Insight
Average voltage_phase1 for Apr 28, 2022 is 382.57, 4.24% lower than voltage_phase2 of 399.49.

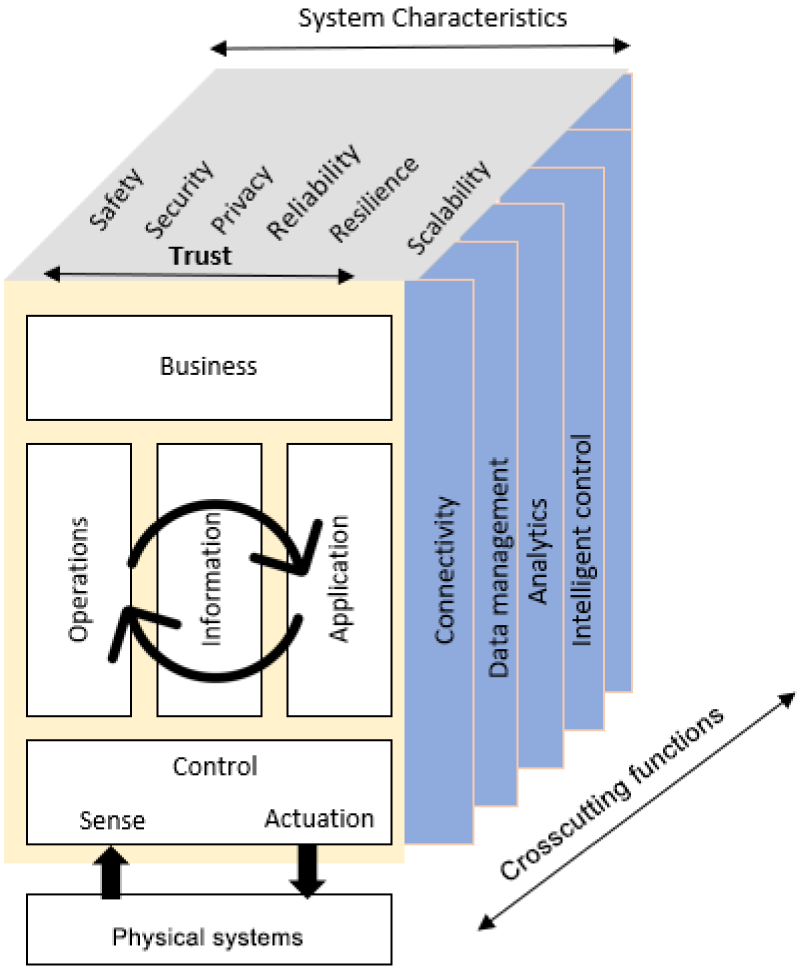
Insight
Average voltage_phase1 for Apr 28, 2022 is 382.57, 0.47% higher than voltage_phase3 of 380.77.

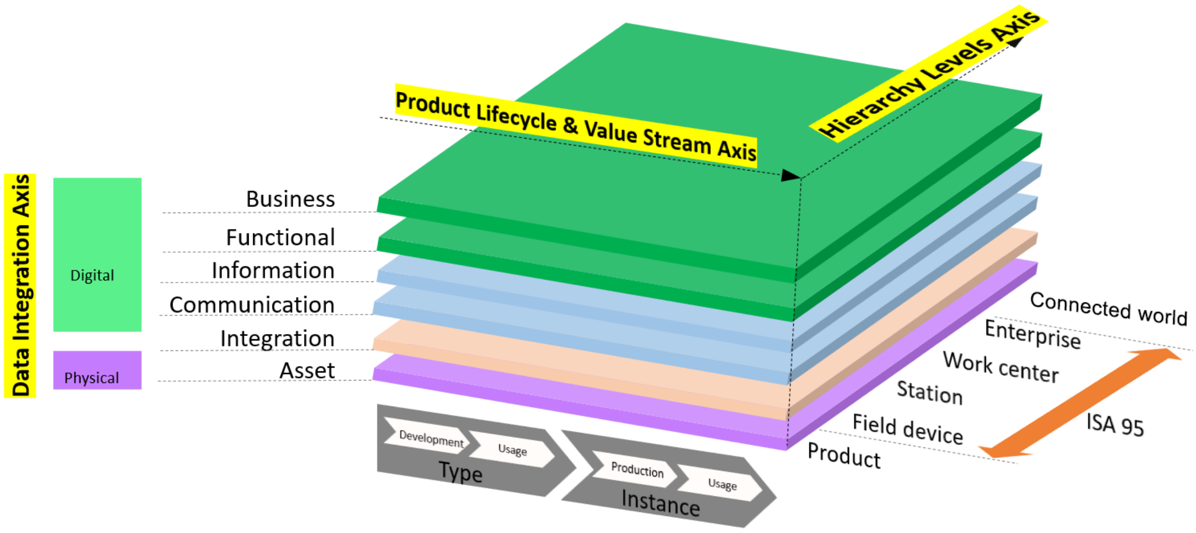
Insight
Average temp for Apr 28, 2022 is 89.13, 31.44% lower than rated_temp of 130.

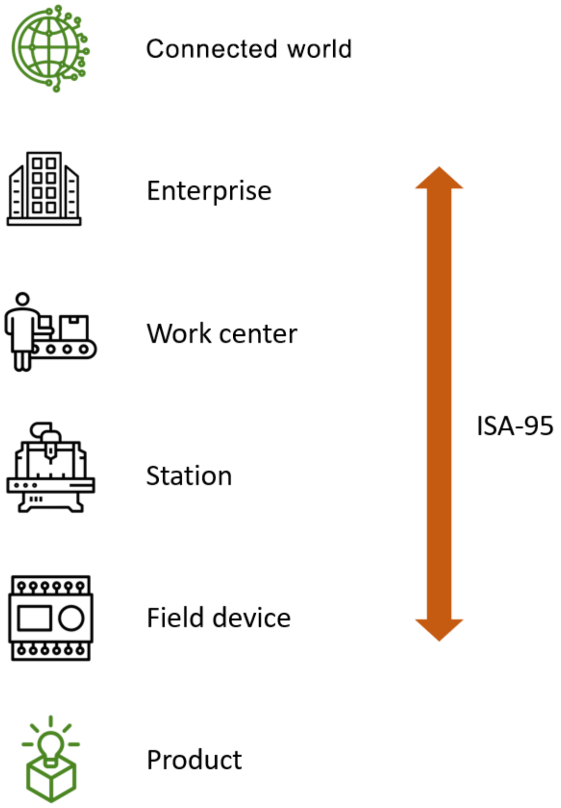
Chapter 9: Taking It Up a Notch – Scalable, Robust, and Secure Architectures

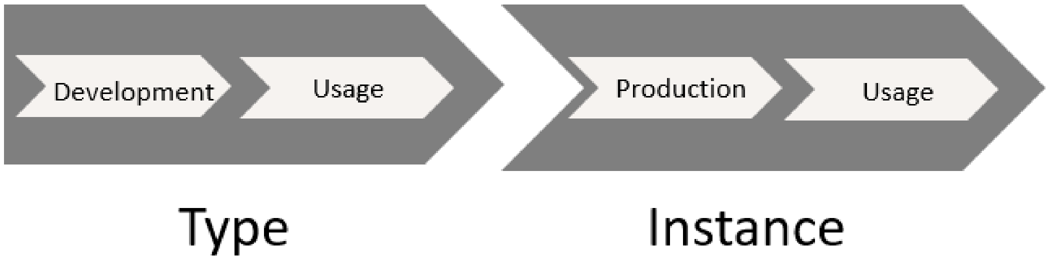


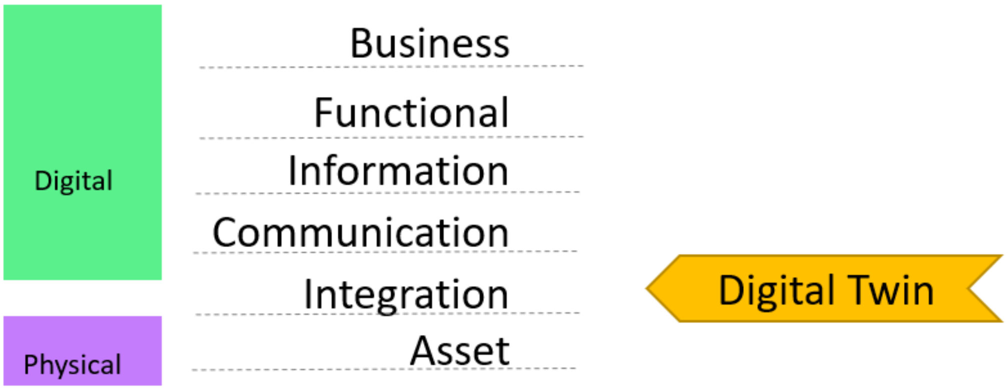


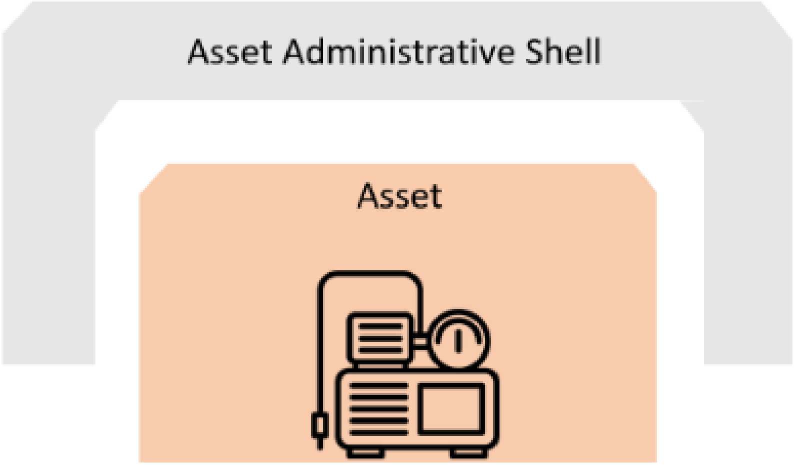

System Characteristics
Safety
Security
Privacy
Reliability
Resilience
Scalability
Trust
Business
Operations
Information
Application
Control
Sense
Actuation
Physical systems
Connectivity
Data management
Analytics
Intelligent control
Crosscutting functions

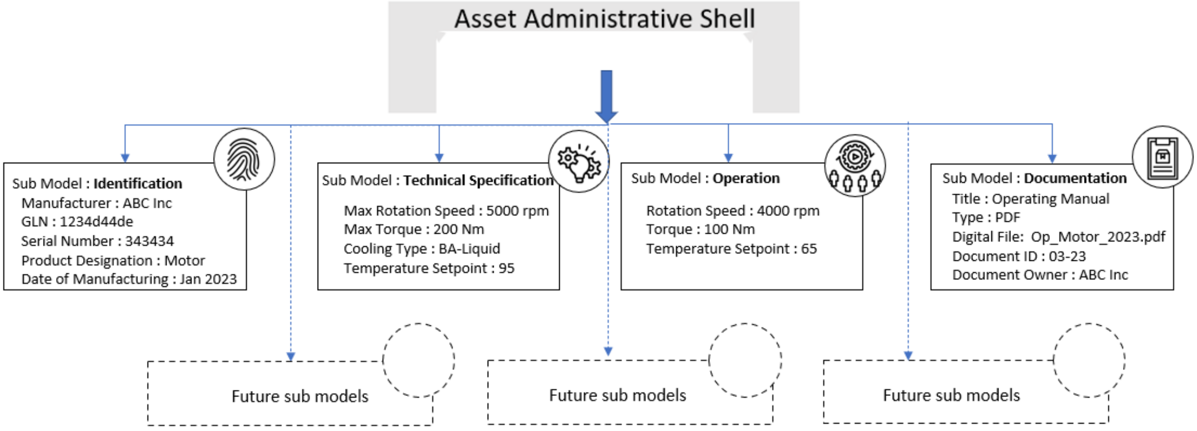

Data Integration Axis
Digital
Physical
Business
Functional
Information
Communication
Integration
Asset
Product Lifecycle & Value Stream Axis
Hierarchy Levels Axis
Development
Usage
Type
Production
Usage
Instance
Connected world
Enterprise
Work center
Station
Field device
Product
ISA 95


Connected world
Enterprise
Work center
Station
Field device
Product
ISA-95


Development
Usage
Production
Usage
Type
Instance


Digital
Physical
Business
Functional
Information
Communication
Integration
Asset
Digital Twin



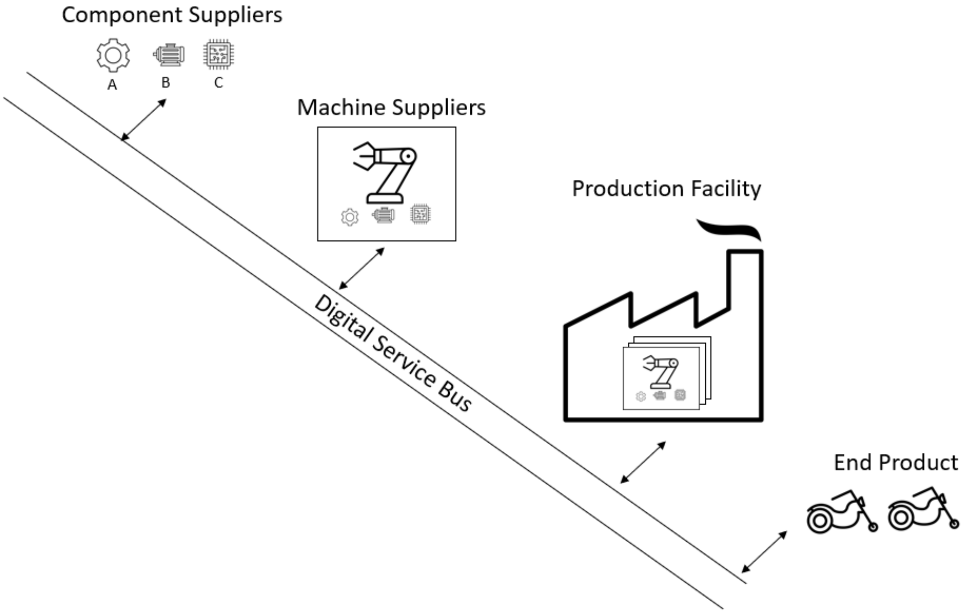


Sub Model : **Identification**
Manufacturer : ABC Inc
GLN : 1234d44de
Serial Number : 343434
Product Designation : Motor
Date of Manufacturing : Jan 2023

Sub Model : **Technical Specification**
Max Rotation Speed : 5000 rpm
Max Torque : 200 Nm
Cooling Type : BA-Liquid
Temperature Setpoint : 95

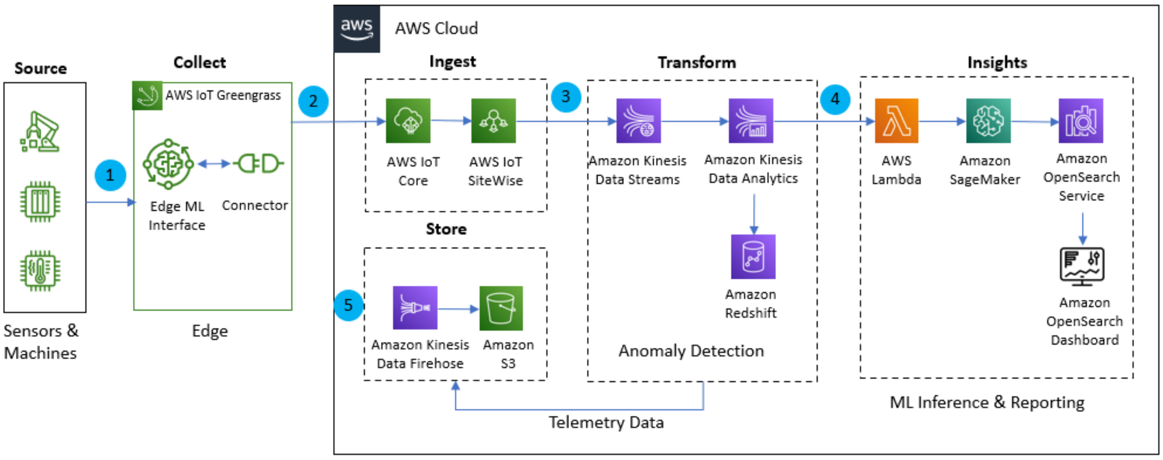
Sub Model : **Operation**
Rotation Speed : 4000 rpm
Torque : 100 Nm
Temperature Setpoint : 65

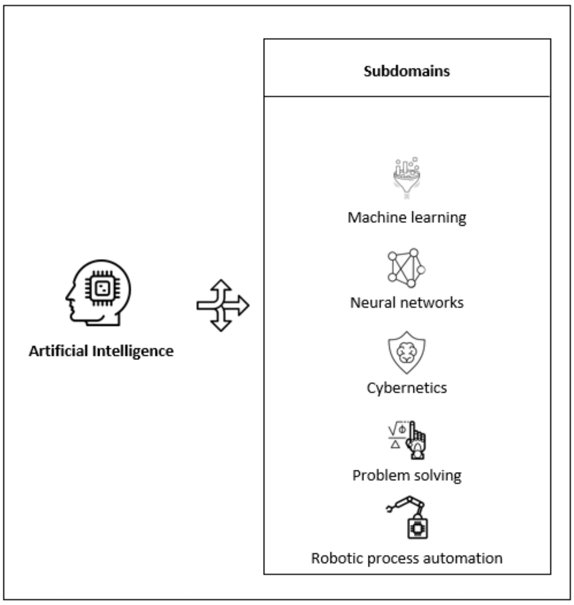
Sub Model : **Documentation**
Title : Operating Manual
Type : PDF
Digital File: Op_Motor_2023.pdf
Document ID : 03-23
Document Owner : ABC Inc



- Bandwidth/speed
- Data volume
- Network latency
- Privacy
- Security
- Autonomy
- Cost

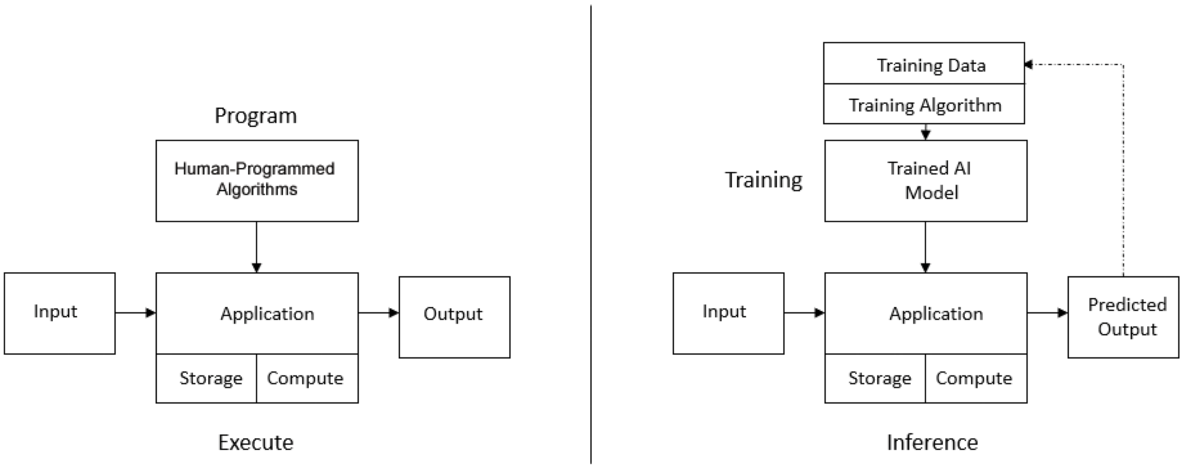
- Availability
- Enterprise integration
- Global plant network
- Geo-political boundaries
- Compute power
- Storage
- Audit and governance

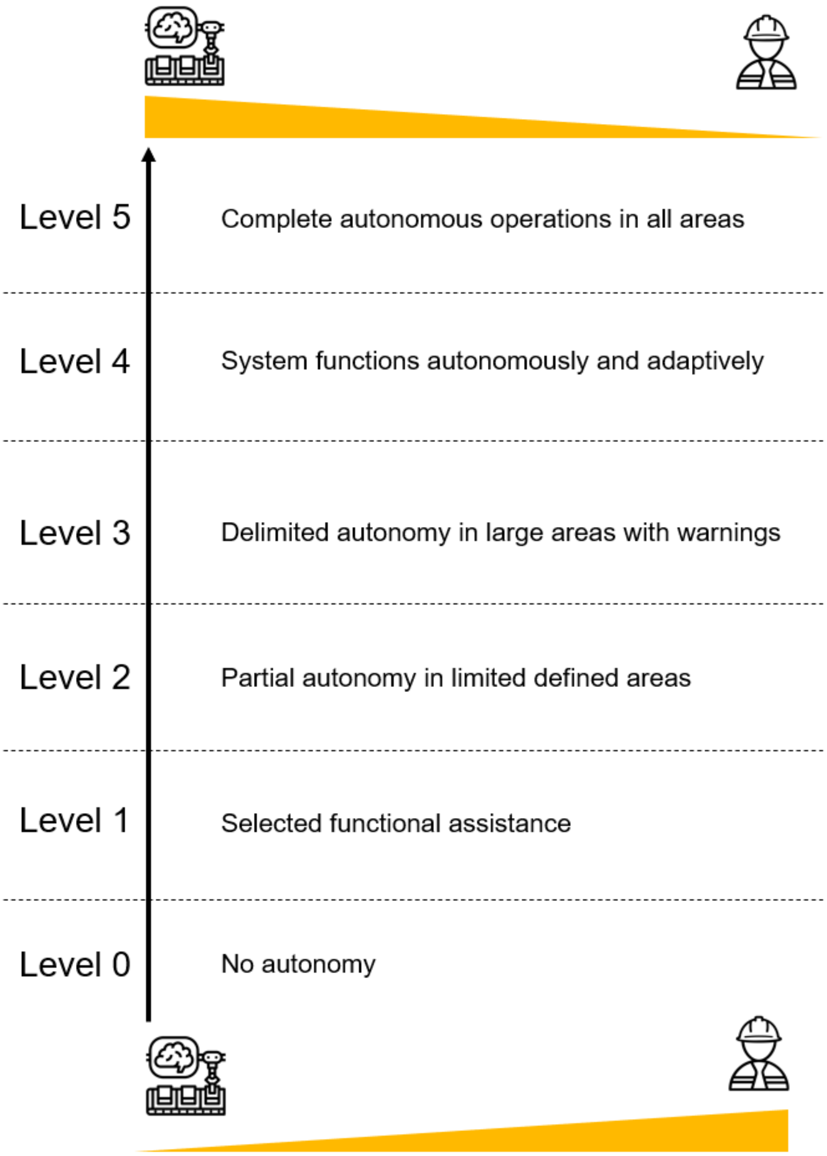




Non-Exhaustive Application List

- Complex data analytics
- Fault detection
- Predictive maintenance
- Visual quality inspection
- Advanced scheduling
- Workforce optimization
- Supply chain optimization
- Demand-predicted production
- Threat modeling and prevention
- Conversational chatbots





Level 5 Complete autonomous operations in all areas

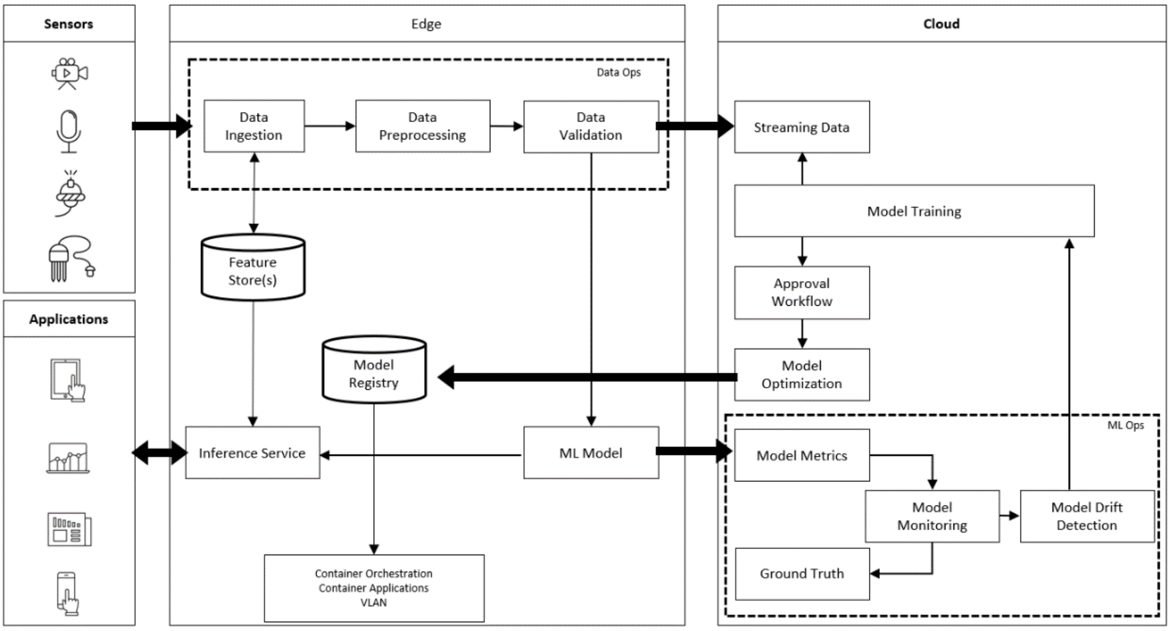
Level 4 System functions autonomously and adaptively

Level 3 Delimited autonomy in large areas with warnings

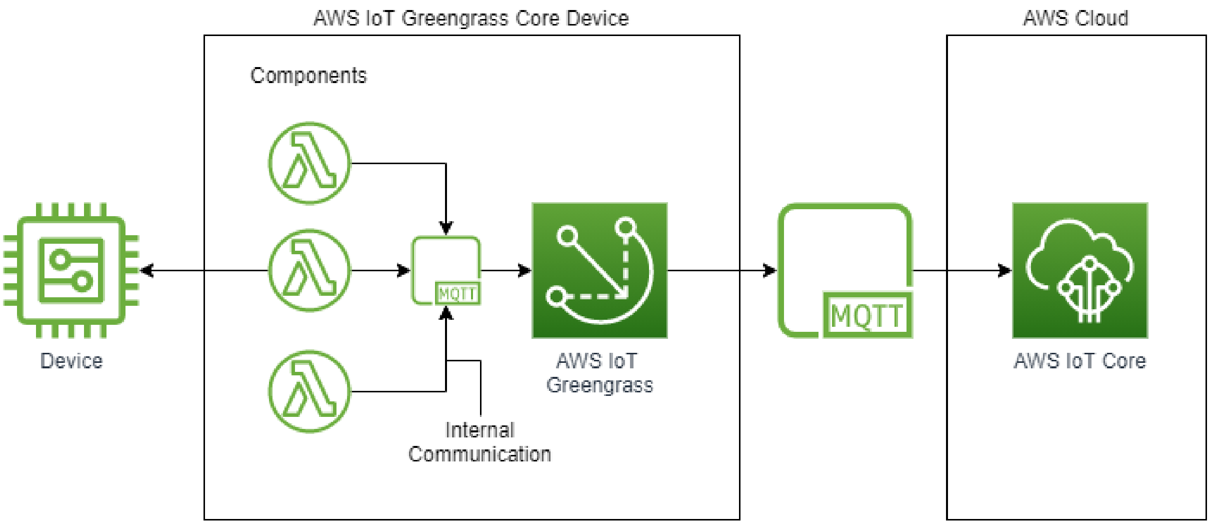
Level 2 Partial autonomy in limited defined areas

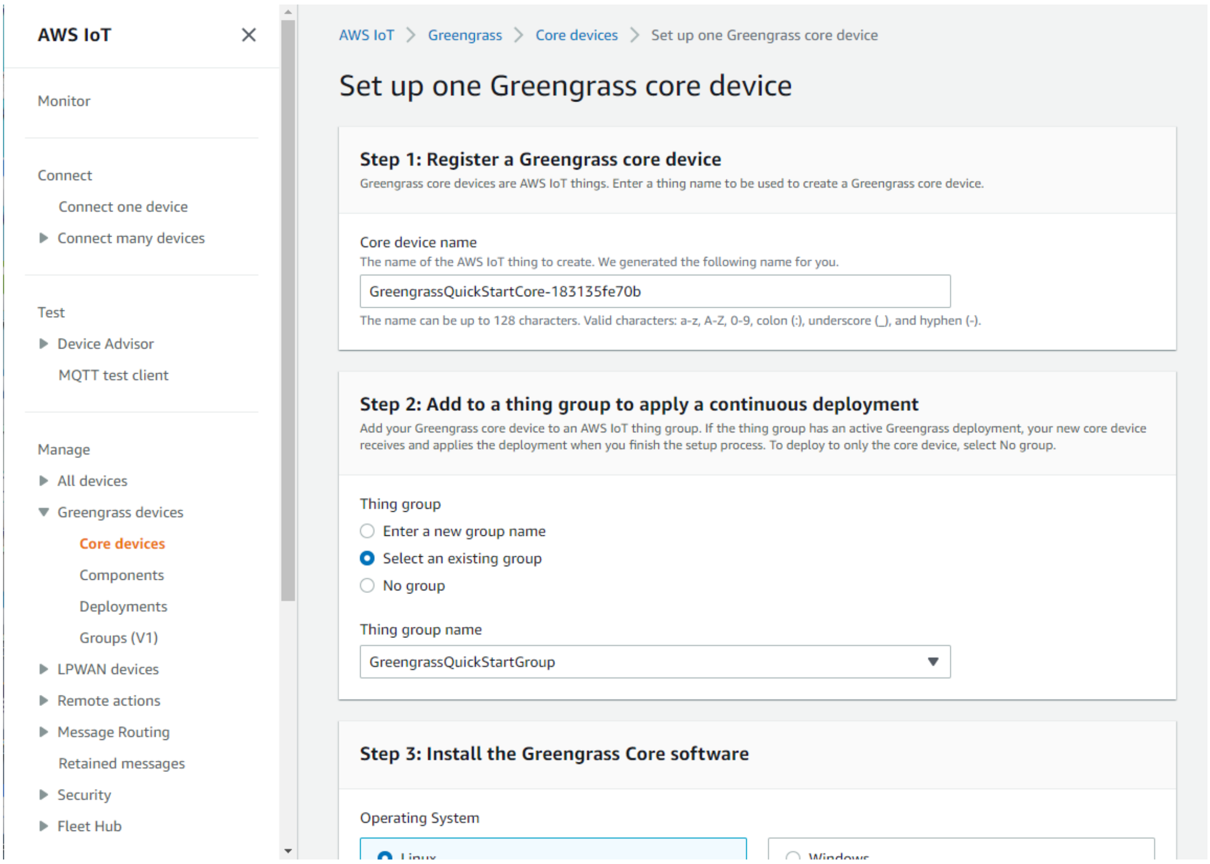
Level 1 Selected functional assistance

Level 0 No autonomy


Sensors
Edge
Cloud
Data Ops
Data Ingestion
Data Preprocessing
Data Validation
Streaming Data
Model Training
Feature Store(s)
Approval Workflow
Applications
Model Registry
Model Optimization
ML Ops
Inference Service
ML Model
Model Metrics
Model Monitoring
Model Drift Detection
Ground Truth
Container Orchestration
Container Applications
VLAN

Chapter 10: Intelligent Systems at the Edge





Set up one Greengrass core device

Step 1: Register a Greengrass core device

Greengrass core devices are AWS IoT things. Enter a thing name to be used to create a Greengrass core device.

Core device name

The name of the AWS IoT thing to create. We generated the following name for you.

GreengrassQuickStartCore-183135fe70b

The name can be up to 128 characters. Valid characters: a-z, A-Z, 0-9, colon (:), underscore (_), and hyphen (-).

Step 2: Add to a thing group to apply a continuous deployment

Add your Greengrass core device to an AWS IoT thing group. If the thing group has an active Greengrass deployment, your new core device receives and applies the deployment when you finish the setup process. To deploy to only the core device, select No group.

Thing group

○ Enter a new group name
● Select an existing group
○ No group

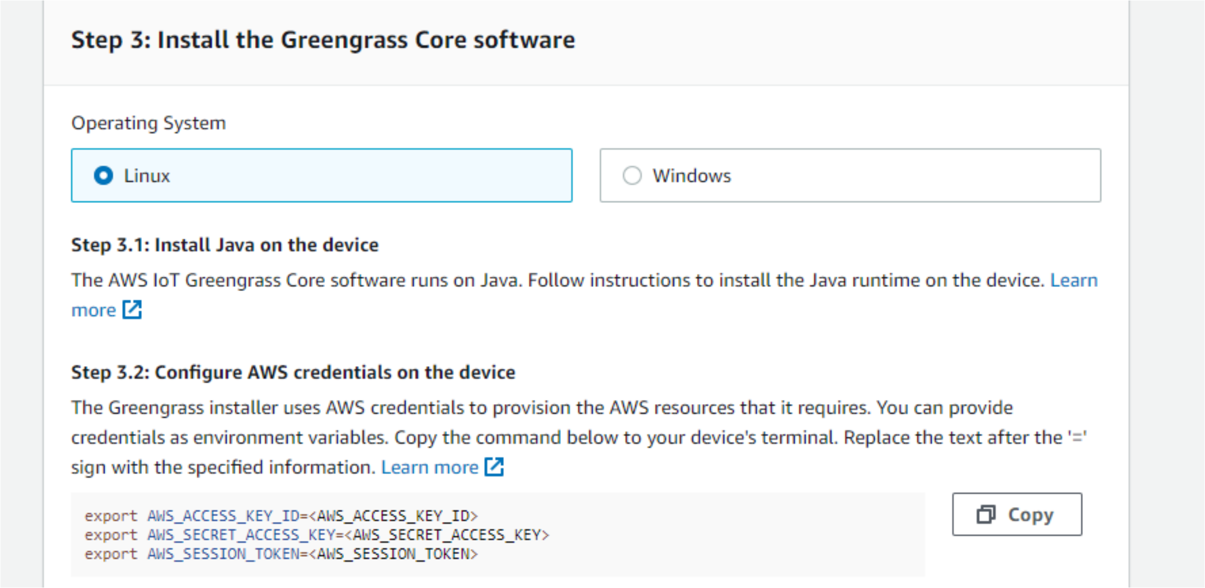
Thing group name

GreengrassQuickStartGroup ▼

Step 3: Install the Greengrass Core software

Operating System

● Linux ○ Windows



Step 3: Install the Greengrass Core software

Operating System

● Linux ○ Windows

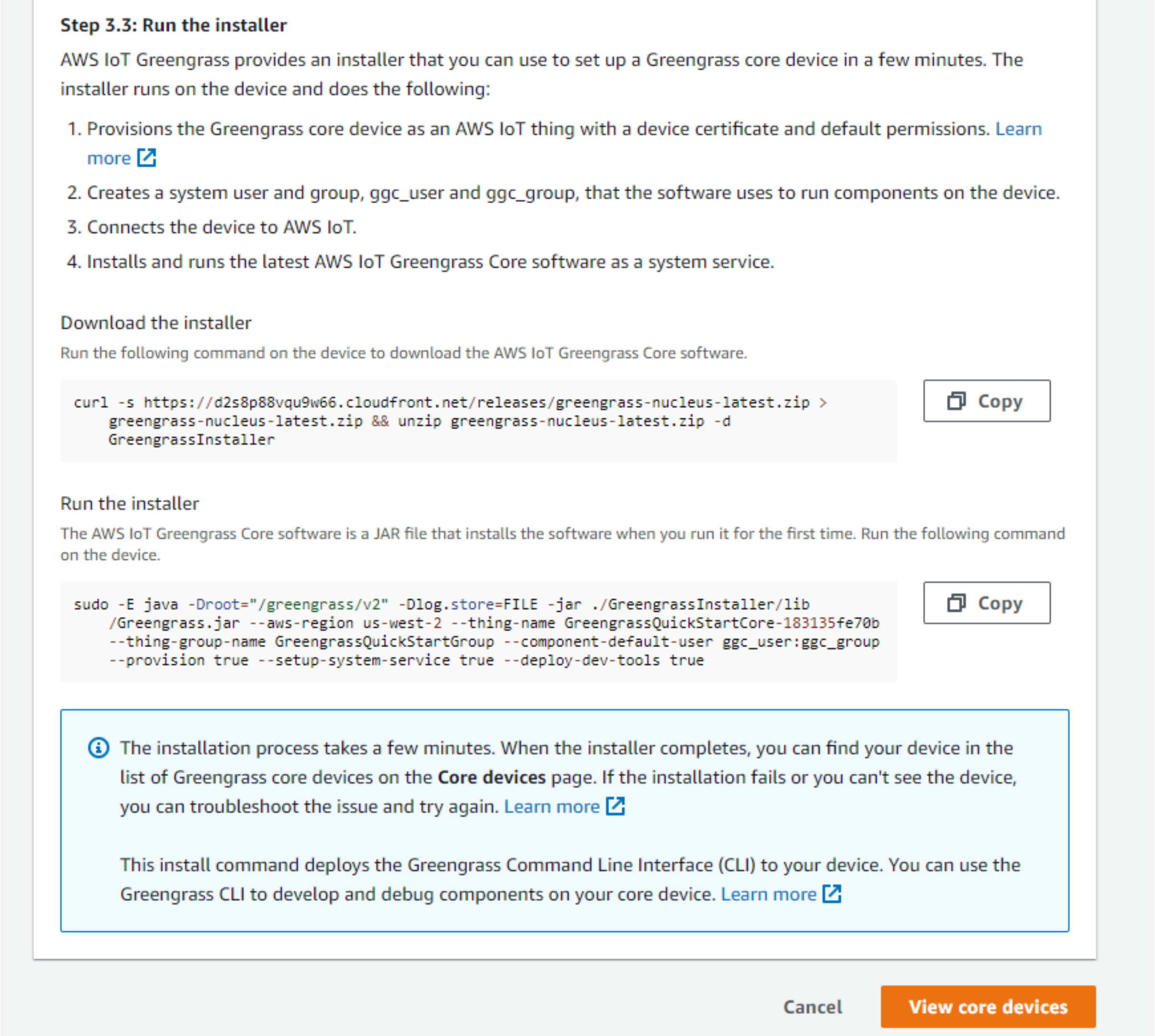
Step 3.1: Install Java on the device

The AWS IoT Greengrass Core software runs on Java. Follow instructions to install the Java runtime on the device. Learn more

Step 3.2: Configure AWS credentials on the device

The Greengrass installer uses AWS credentials to provision the AWS resources that it requires. You can provide credentials as environment variables. Copy the command below to your device's terminal. Replace the text after the '=' sign with the specified information. Learn more

```
export AWS_ACCESS_KEY_ID=<AWS_ACCESS_KEY_ID>
export AWS_SECRET_ACCESS_KEY=<AWS_SECRET_ACCESS_KEY>
export AWS_SESSION_TOKEN=<AWS_SESSION_TOKEN>
```



Step 3.3: Run the installer

AWS IoT Greengrass provides an installer that you can use to set up a Greengrass core device in a few minutes. The installer runs on the device and does the following:

1. Provisions the Greengrass core device as an AWS IoT thing with a device certificate and default permissions. Learn more
2. Creates a system user and group, ggc_user and ggc_group, that the software uses to run components on the device.
3. Connects the device to AWS IoT.
4. Installs and runs the latest AWS IoT Greengrass Core software as a system service.

Download the installer

Run the following command on the device to download the AWS IoT Greengrass Core software.

```
curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip >
    greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d
    GreengrassInstaller
```

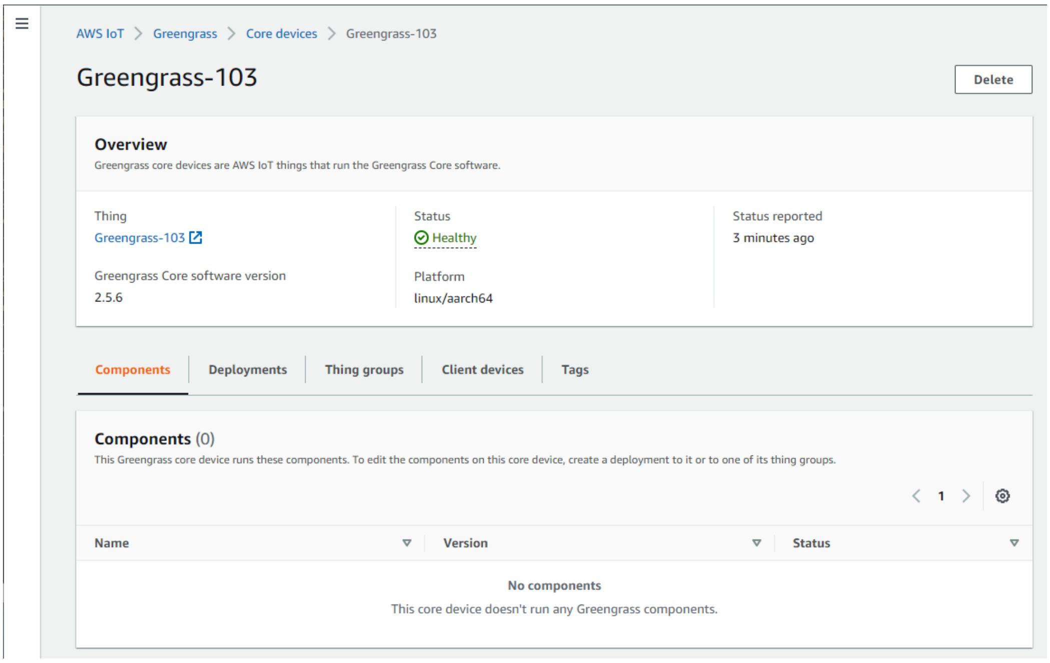
Run the installer

The AWS IoT Greengrass Core software is a JAR file that installs the software when you run it for the first time. Run the following command on the device.

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE -jar ./GreengrassInstaller/lib
    /Greengrass.jar --aws-region us-west-2 --thing-name GreengrassQuickStartCore-183135fe70b
    --thing-group-name GreengrassQuickStartGroup --component-default-user ggc_user:ggc_group
    --provision true --setup-system-service true --deploy-dev-tools true
```

The installation process takes a few minutes. When the installer completes, you can find your device in the list of Greengrass core devices on the **Core devices** page. If the installation fails or you can't see the device, you can troubleshoot the issue and try again. Learn more

This install command deploys the Greengrass Command Line Interface (CLI) to your device. You can use the Greengrass CLI to develop and debug components on your core device. Learn more



AWS IoT > Greengrass > Core devices > Greengrass-103

Greengrass-103

Delete

Overview

Greengrass core devices are AWS IoT things that run the Greengrass Core software.

Thing Greengrass-103	Status Healthy	Status reported 3 minutes ago
Greengrass Core software version 2.5.6	Platform linux/aarch64	

Components | Deployments | Thing groups | Client devices | Tags

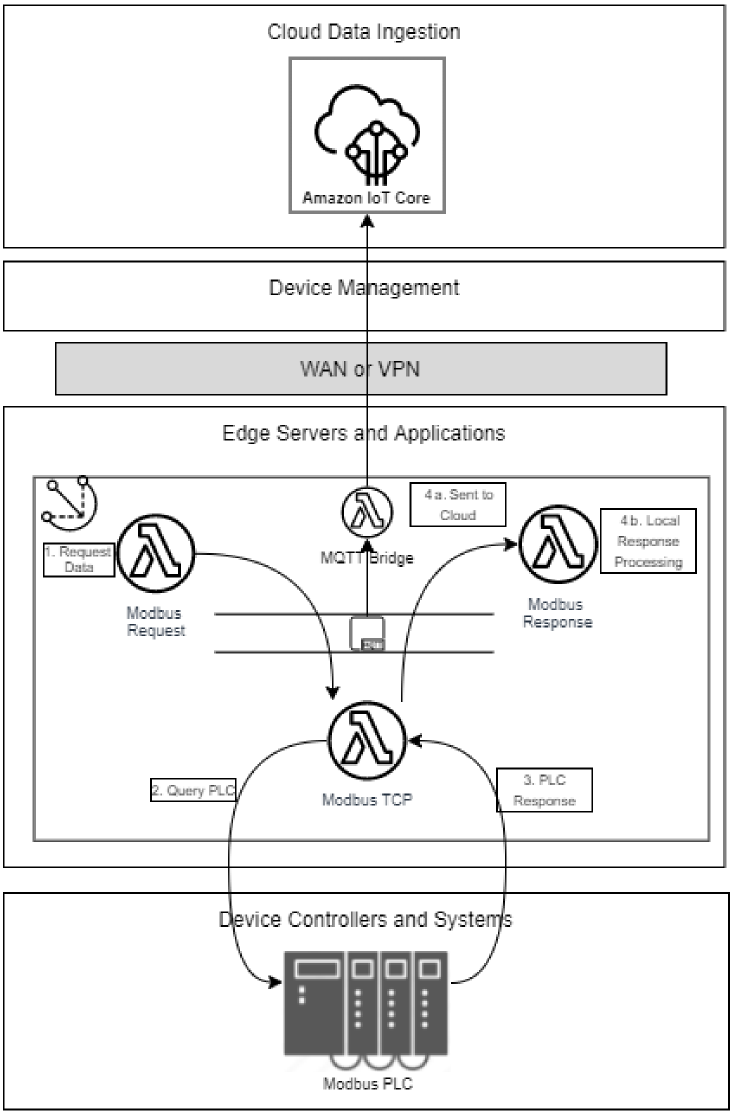
Components (0)

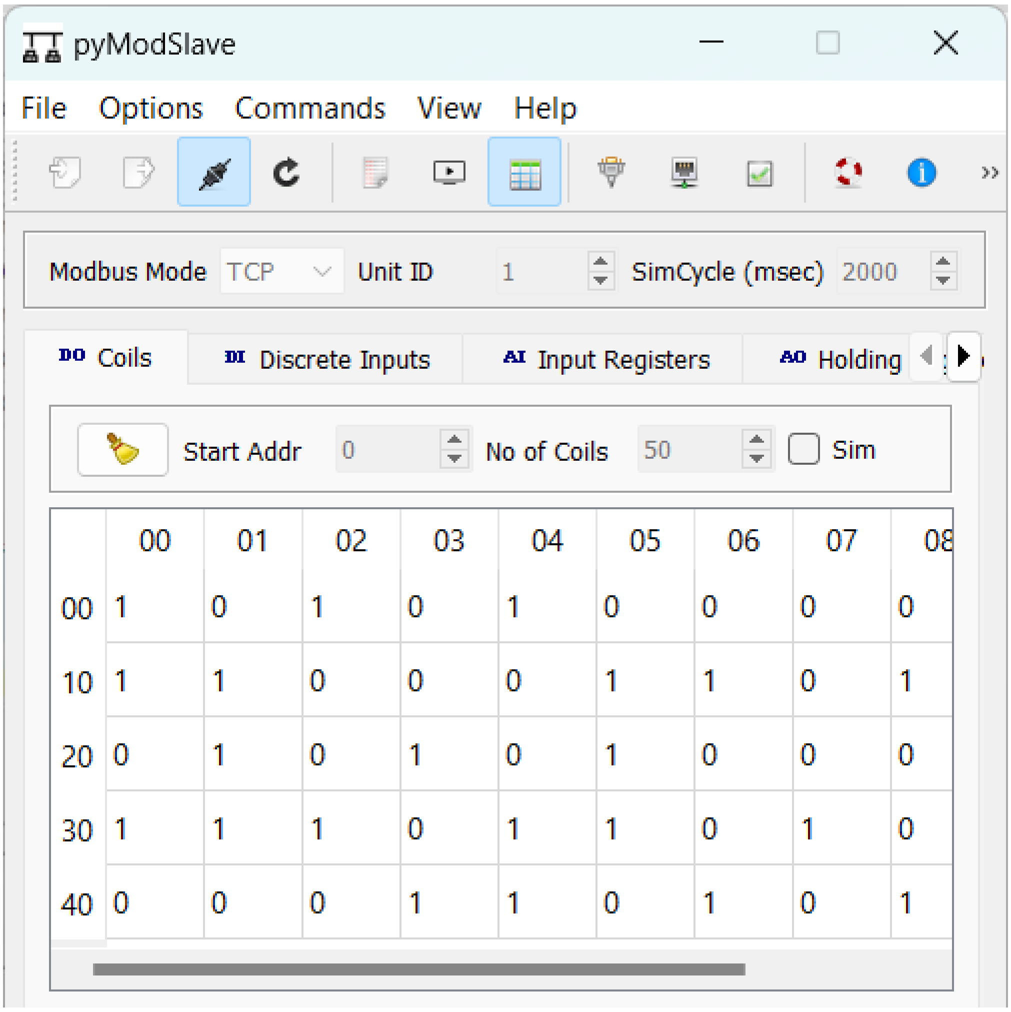
This Greengrass core device runs these components. To edit the components on this core device, create a deployment to it or to one of its thing groups.

Name	Version	Status

No components

This core device doesn't run any Greengrass components.





Modbus Mode TCP

Unit ID 1

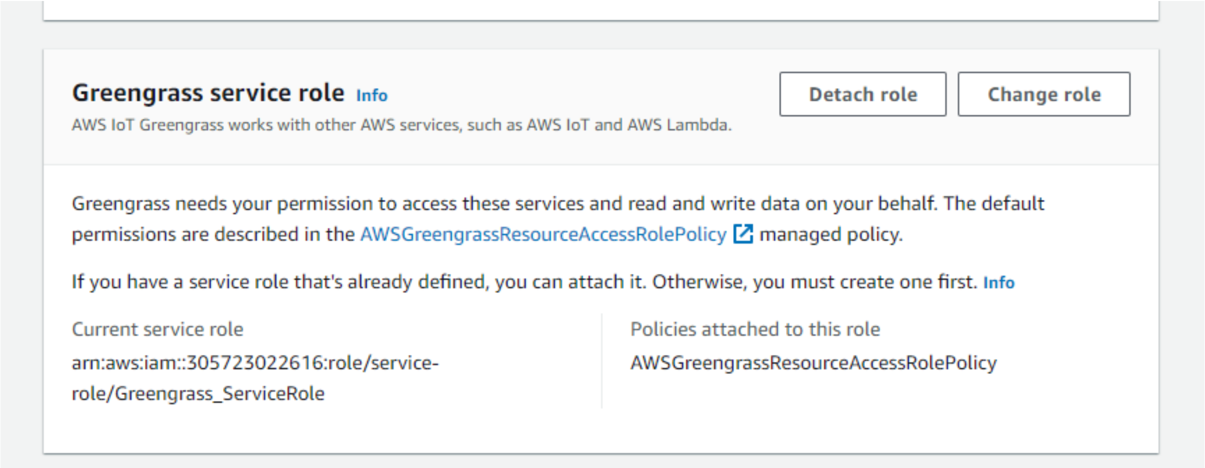
SimCycle (msec) 2000

Start Addr 0

No of Coils 50

□ Sim

	00	01	02	03	04	05	06	07	08
00	1	0	1	0	1	0	0	0	0
10	1	1	0	0	0	1	1	0	1
20	0	1	0	1	0	1	0	0	0
30	1	1	1	0	1	1	0	1	0
40	0	0	0	1	1	0	1	0	1



Greengrass service role

AWS IoT Greengrass works with other AWS services, such as AWS IoT and AWS Lambda.

Greengrass needs your permission to access these services and read and write data on your behalf. The default permissions are described in the AWSGreengrassResourceAccessRolePolicy managed policy.

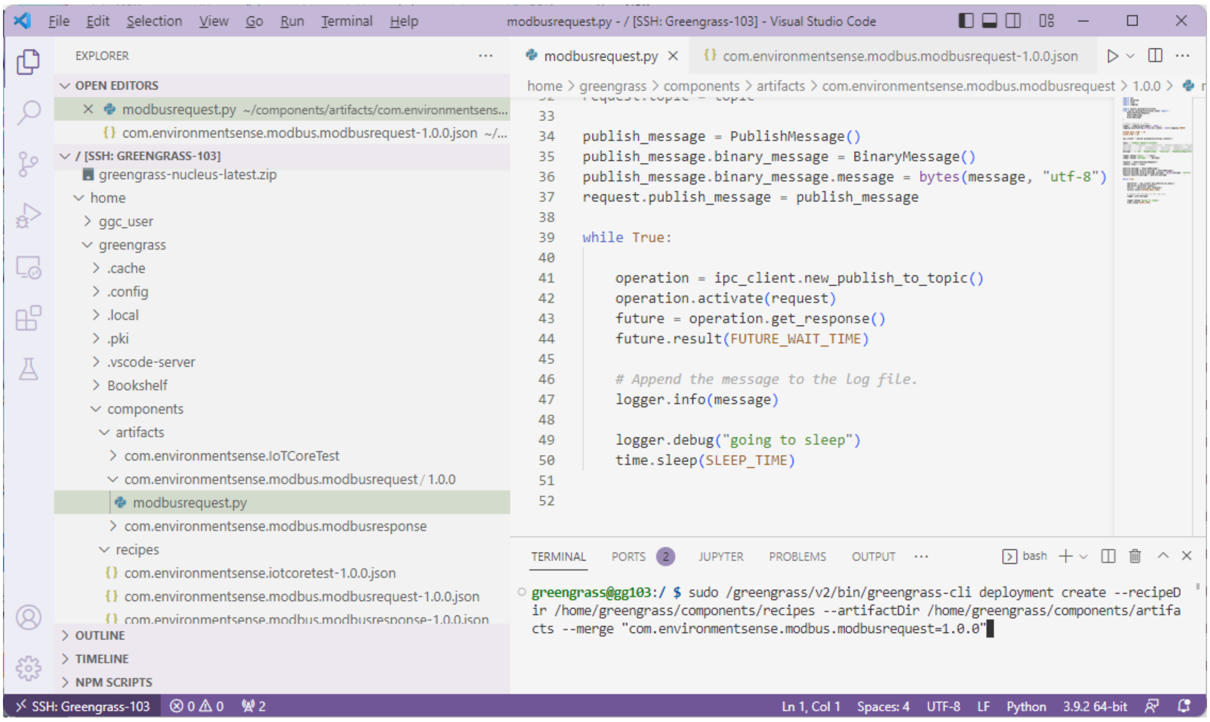
If you have a service role that's already defined, you can attach it. Otherwise, you must create one first. Info

Current service role

arn:aws:iam::305723022616:role/service-role/Greengrass_ServiceRole

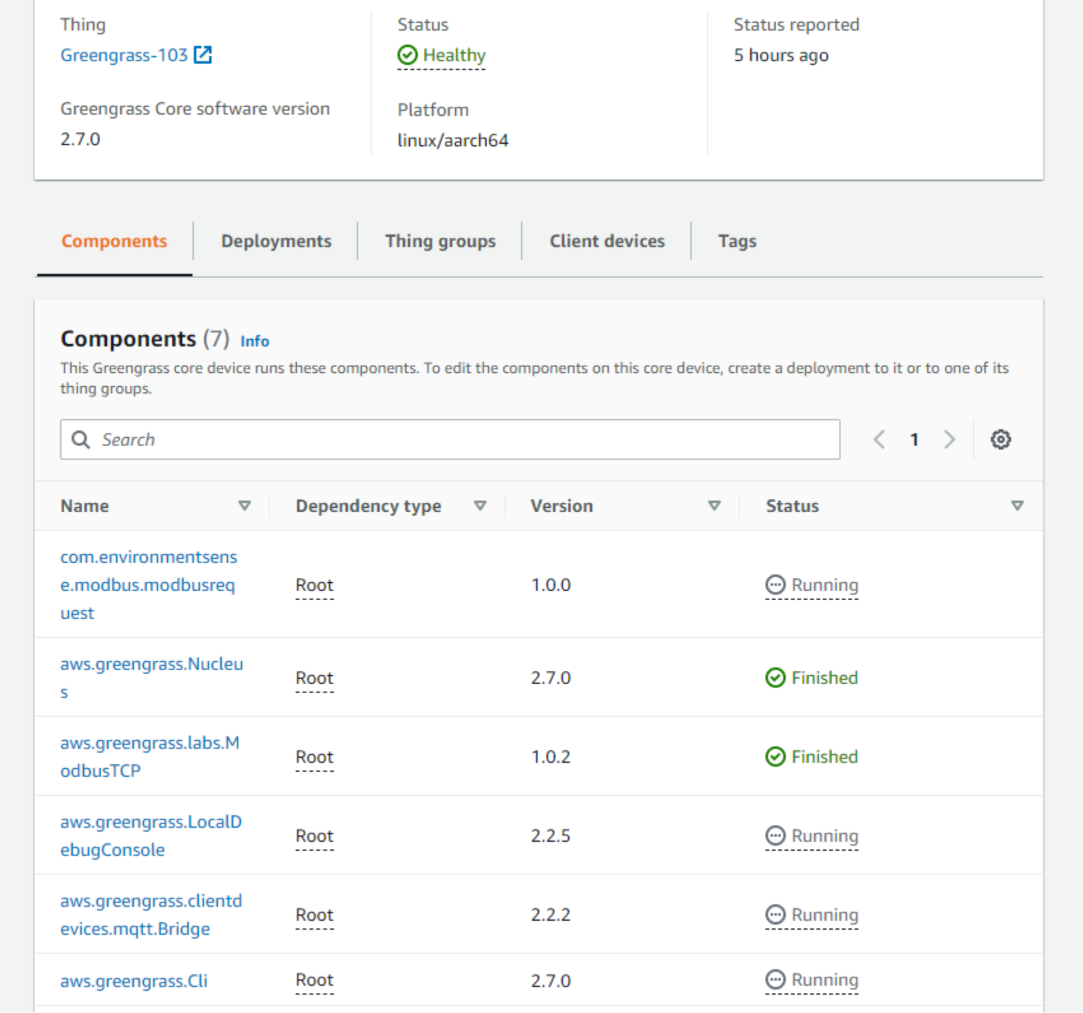
Policies attached to this role

AWSGreengrassResourceAccessRolePolicy


File
Edit
Selection
View
Go
Run
Terminal
Help
modbusrequest.py - / [SSH: Greengrass-103] - Visual Studio Code
EXPLORER
OPEN EDITORS
modbusrequest.py ~/components/artifacts/com.environmentsens...
com.environmentsense.modbus.modbusrequest-1.0.0.json ~/...
/ [SSH: GREENGRASS-103]
greengrass-nucleus-latest.zip
home
ggc_user
greengrass
.cache
.config
.local
.pki
.vscode-server
Bookshelf
components
artifacts
com.environmentsense.IoTCoreTest
com.environmentsense.modbus.modbusrequest / 1.0.0
modbusrequest.py
com.environmentsense.modbus.modbusresponse
recipes
com.environmentsense.iotcoretest-1.0.0.json
com.environmentsense.modbus.modbusrequest-1.0.0.json
com.environmentsense.modbus.modbusresponse-1.0.0.json
OUTLINE
TIMELINE
NPM SCRIPTS
modbusrequest.py
com.environmentsense.modbus.modbusrequest-1.0.0.json
home > greengrass > components > artifacts > com.environmentsense.modbus.modbusrequest > 1.0.0 >
TERMINAL
PORTS 2
JUPYTER
PROBLEMS
OUTPUT
bash
SSH: Greengrass-103
⊗ 0 ⚠ 0
2
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Python
3.9.2 64-bit

```python
publish_message = PublishMessage()
publish_message.binary_message = BinaryMessage()
publish_message.binary_message.message = bytes(message, "utf-8")
request.publish_message = publish_message

while True:

    operation = ipc_client.new_publish_to_topic()
    operation.activate(request)
    future = operation.get_response()
    future.result(FUTURE_WAIT_TIME)

    # Append the message to the log file.
    logger.info(message)

    logger.debug("going to sleep")
    time.sleep(SLEEP_TIME)
```

```
greengrass@gg103:/ $ sudo /greengrass/v2/bin/greengrass-cli deployment create --recipeD
ir /home/greengrass/components/recipes --artifactDir /home/greengrass/components/artifa
cts --merge "com.environmentsense.modbus.modbusrequest=1.0.0"
```



Thing
Greengrass-103

Status
Healthy

Status reported
5 hours ago

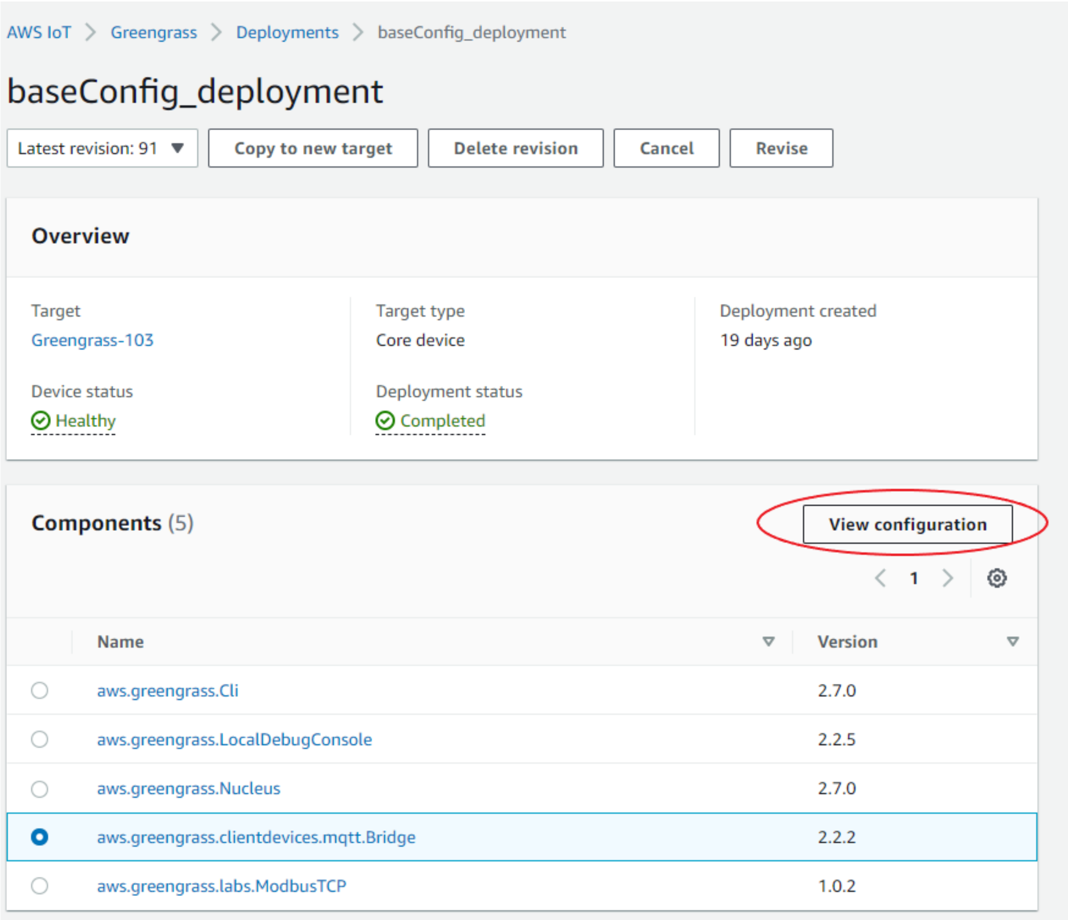
Greengrass Core software version
2.7.0

Platform
linux/aarch64

Components (7) Info

This Greengrass core device runs these components. To edit the components on this core device, create a deployment to it or to one of its thing groups.

Name	Dependency type	Version	Status
com.environmentsense.modbus.modbusrequest	Root	1.0.0	Running
aws.greengrass.Nucleus	Root	2.7.0	Finished
aws.greengrass.labs.ModbusTCP	Root	1.0.2	Finished
aws.greengrass.LocalDebugConsole	Root	2.2.5	Running
aws.greengrass.clientdevices.mqtt.Bridge	Root	2.2.2	Running
aws.greengrass.Cli	Root	2.7.0	Running



AWS IoT > Greengrass > Deployments > baseConfig_deployment

baseConfig_deployment

Overview

Target: Greengrass-103

Target type: Core device

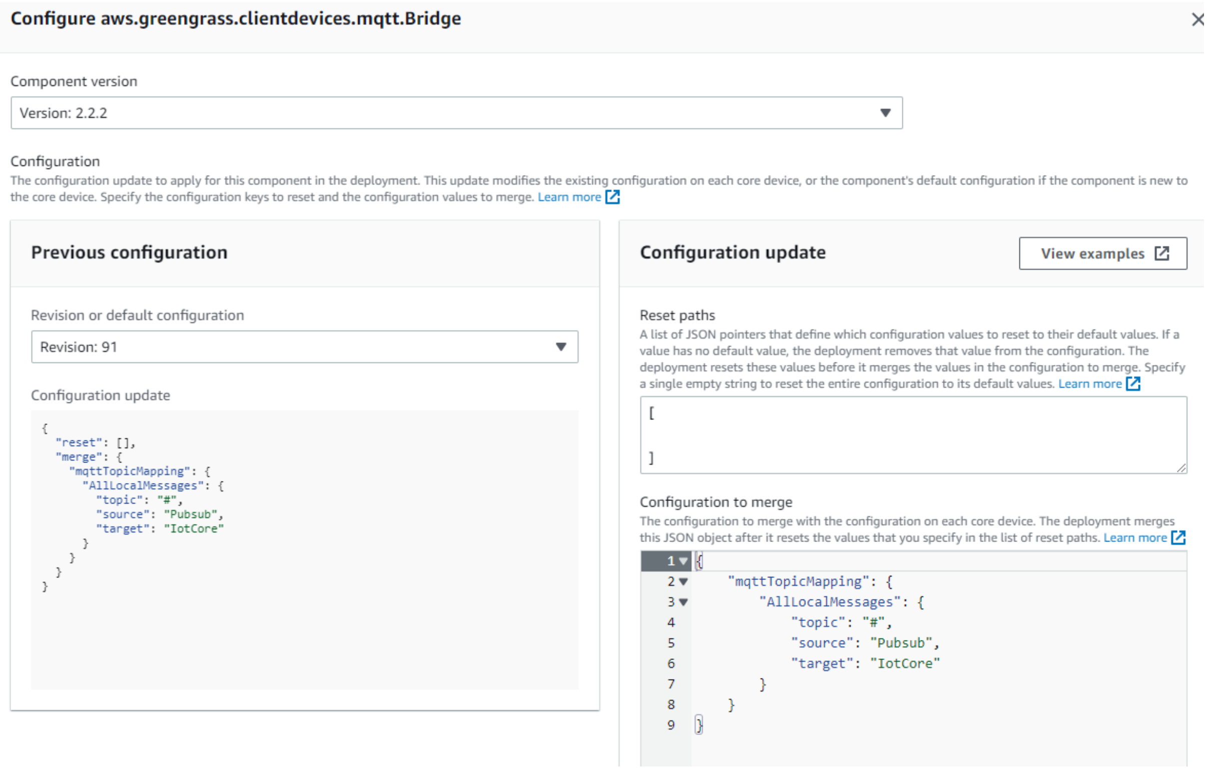
Deployment created: 19 days ago

Device status: ✓ Healthy

Deployment status: ✓ Completed

Components (5)

	Name	Version
○	aws.greengrass.Cli	2.7.0
○	aws.greengrass.LocalDebugConsole	2.2.5
○	aws.greengrass.Nucleus	2.7.0
●	aws.greengrass.clientdevices.mqtt.Bridge	2.2.2
○	aws.greengrass.labs.ModbusTCP	1.0.2



Configure aws.greengrass.clientdevices.mqtt.Bridge

Component version: Version: 2.2.2

Configuration

The configuration update to apply for this component in the deployment. This update modifies the existing configuration on each core device, or the component's default configuration if the component is new to the core device. Specify the configuration keys to reset and the configuration values to merge. Learn more

Previous configuration

Revision or default configuration: Revision: 91

Configuration update

```
{
  "reset": [],
  "merge": {
    "mqttTopicMapping": {
      "AllLocalMessages": {
        "topic": "#",
        "source": "Pubsub",
        "target": "IotCore"
      }
    }
  }
}
```

Configuration update

Reset paths

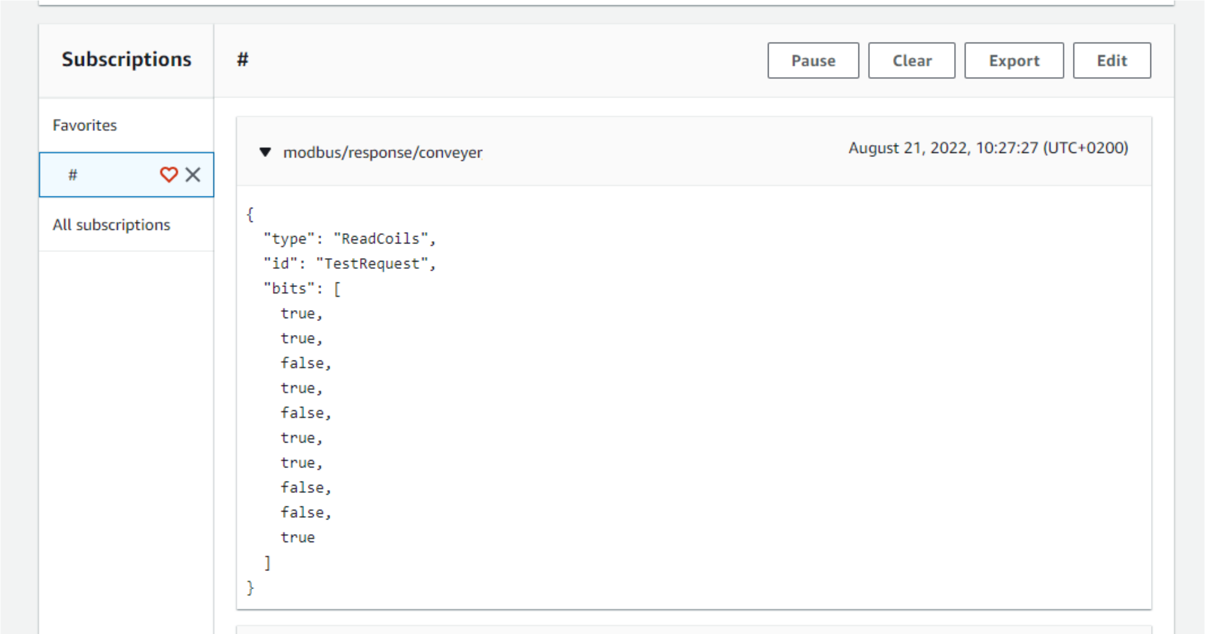
A list of JSON pointers that define which configuration values to reset to their default values. If a value has no default value, the deployment removes that value from the configuration. The deployment resets these values before it merges the values in the configuration to merge. Specify a single empty string to reset the entire configuration to its default values. Learn more

```
[

]
```

Configuration to merge

The configuration to merge with the configuration on each core device. The deployment merges this JSON object after it resets the values that you specify in the list of reset paths. Learn more

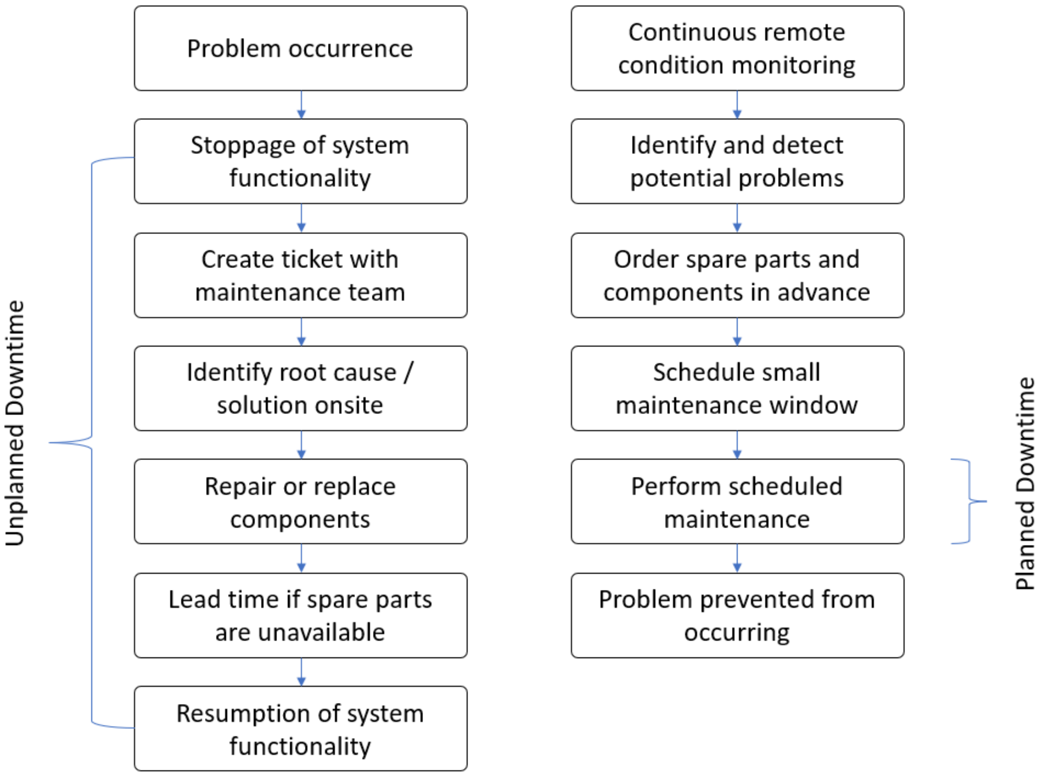
```
{
    "mqttTopicMapping": {
        "AllLocalMessages": {
            "topic": "#",
            "source": "Pubsub",
            "target": "IotCore"
        }
    }
}
```

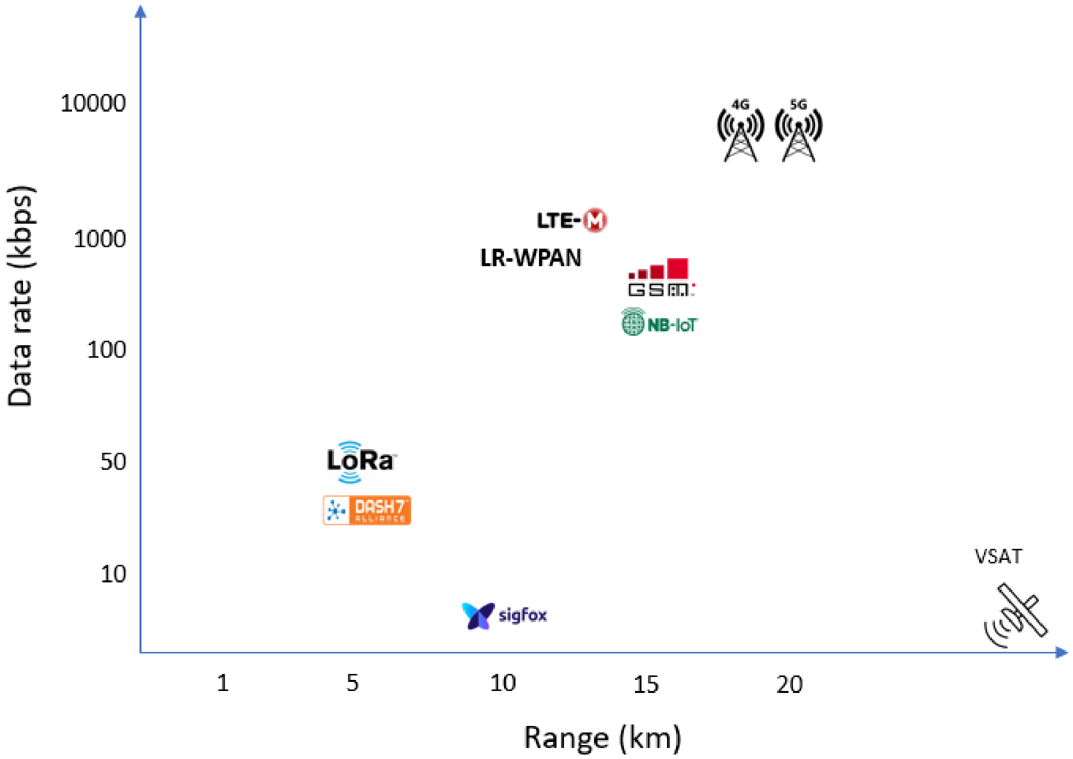


▼ modbus/response/conveyer August 21, 2022, 10:27:27 (UTC+0200)

```
{
  "type": "ReadCoils",
  "id": "TestRequest",
  "bits": [
    true,
    true,
    false,
    true,
    false,
    true,
    true,
    false,
    false,
    true
  ]
}
```

Chapter 11: Remote Monitoring Challenges


Problem occurrence
Stoppage of system functionality
Create ticket with maintenance team
Identify root cause / solution onsite
Repair or replace components
Lead time if spare parts are unavailable
Resumption of system functionality
Unplanned Downtime
Continuous remote condition monitoring
Identify and detect potential problems
Order spare parts and components in advance
Schedule small maintenance window
Perform scheduled maintenance
Problem prevented from occurring
Planned Downtime

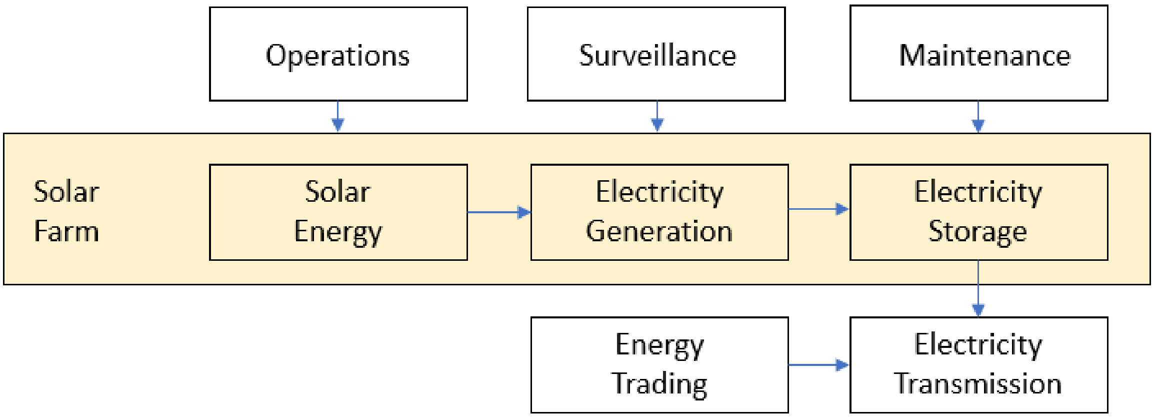


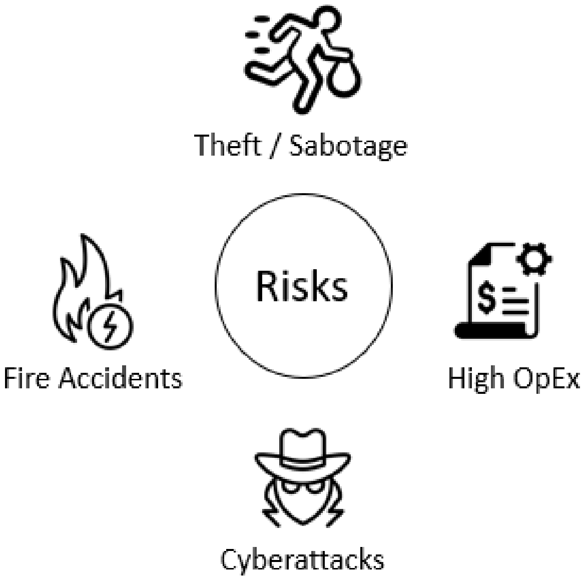
Edge

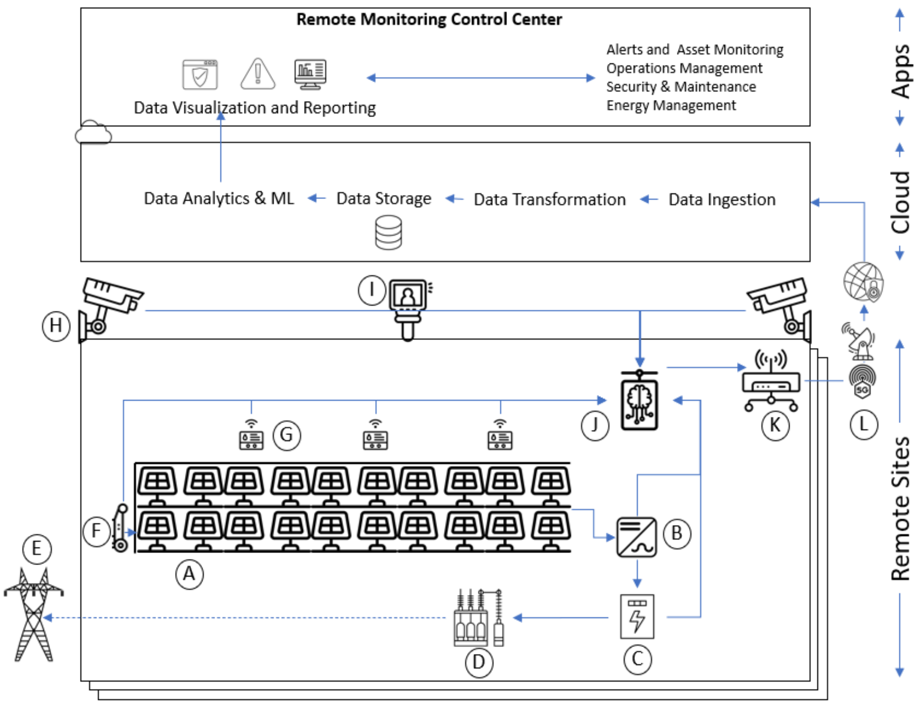
- Real-time decision making
- Control of systems
- Emergency first reaction
- Autonomous / Semi-autonomous systems

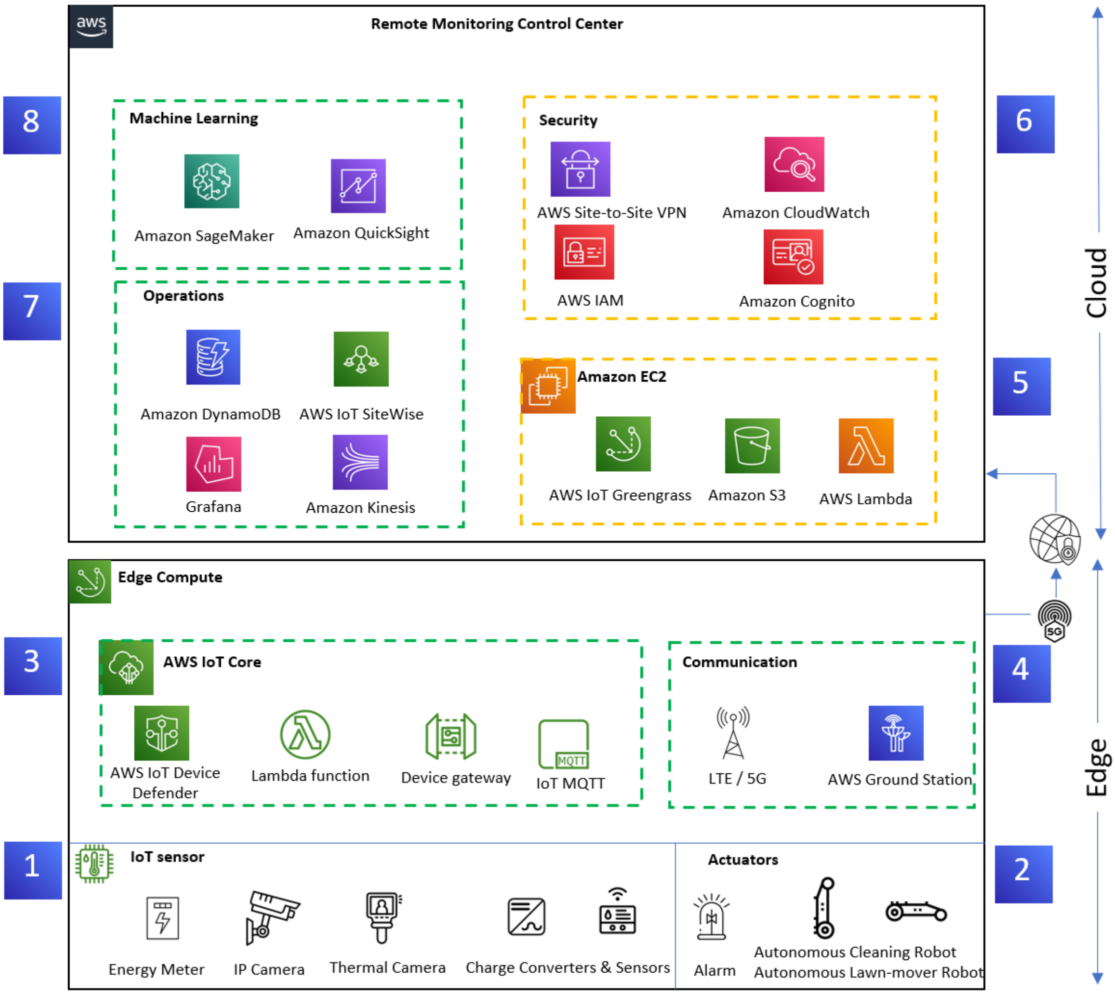
Cloud

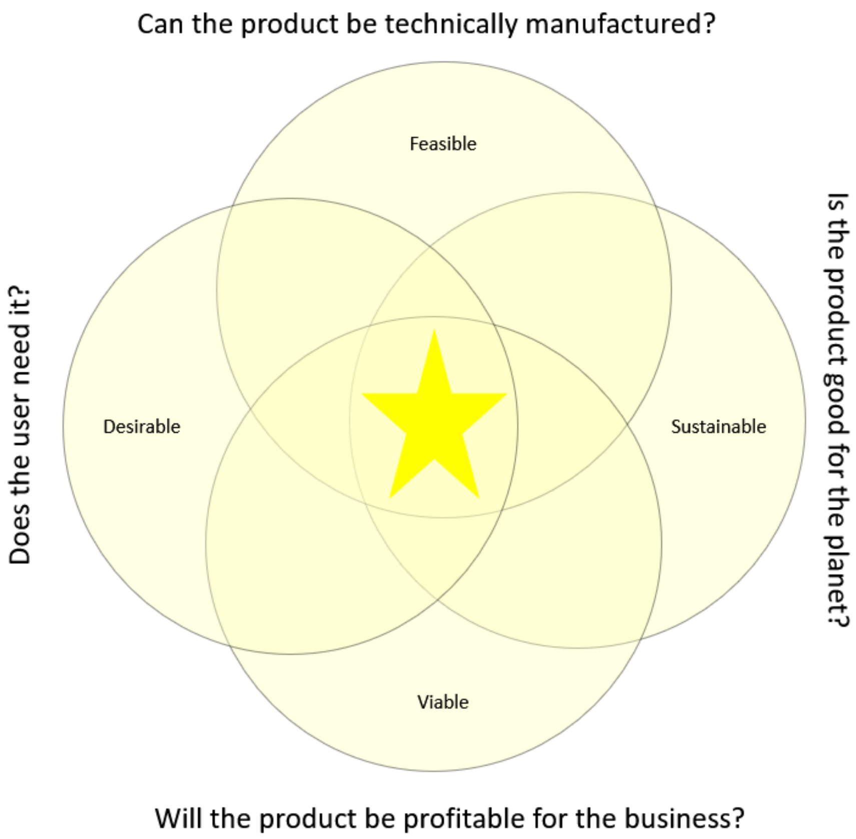
- Distributed systems control
- Expansive inferences through compute
- Forensics and audit
- Human-in-loop decision making
- Manual maintenance
- Supply chain integration

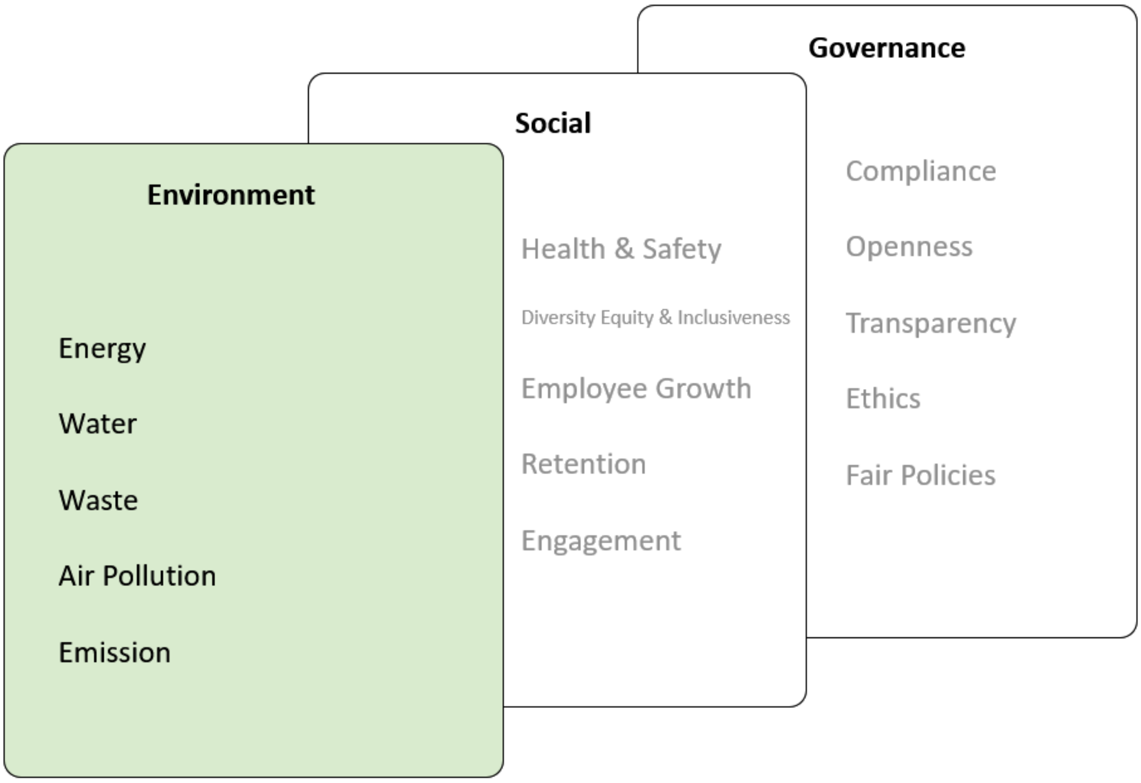





Remote Monitoring Control Center
Data Visualization and Reporting
Alerts and Asset Monitoring
Operations Management
Security & Maintenance
Energy Management
Data Analytics & ML
Data Storage
Data Transformation
Data Ingestion
Apps
Cloud
Remote Sites
A
B
C
D
E
F
G
H
I
J
K
L


aws
Remote Monitoring Control Center
8
Machine Learning
Amazon SageMaker
Amazon QuickSight
7
Operations
Amazon DynamoDB
AWS IoT SiteWise
Grafana
Amazon Kinesis
6
Security
AWS Site-to-Site VPN
Amazon CloudWatch
AWS IAM
Amazon Cognito
5
Amazon EC2
AWS IoT Greengrass
Amazon S3
AWS Lambda
Cloud
Edge Compute
3
AWS IoT Core
AWS IoT Device Defender
Lambda function
Device gateway
IoT MQTT
MQTT
4
Communication
LTE / 5G
AWS Ground Station
5G
Edge
1
IoT sensor
Energy Meter
IP Camera
Thermal Camera
Charge Converters & Sensors
2
Actuators
Alarm
Autonomous Cleaning Robot
Autonomous Lawn-mover Robot


Can the product be technically manufactured?
Feasible
Does the user need it?
Desirable
Sustainable
Is the product good for the planet?
Viable
Will the product be profitable for the business?



Governance

Compliance

Openness

Transparency

Ethics

Fair Policies

Social

Health & Safety

Diversity Equity & Inclusiveness

Employee Growth

Retention

Engagement

Environment

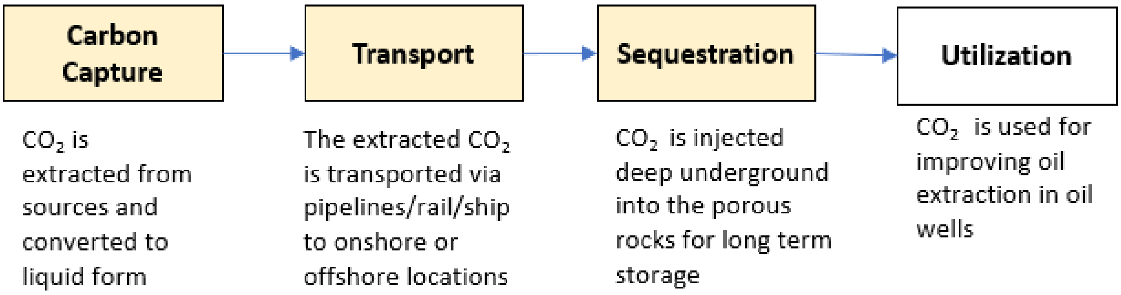
Energy

Water

Waste

Air Pollution

Emission

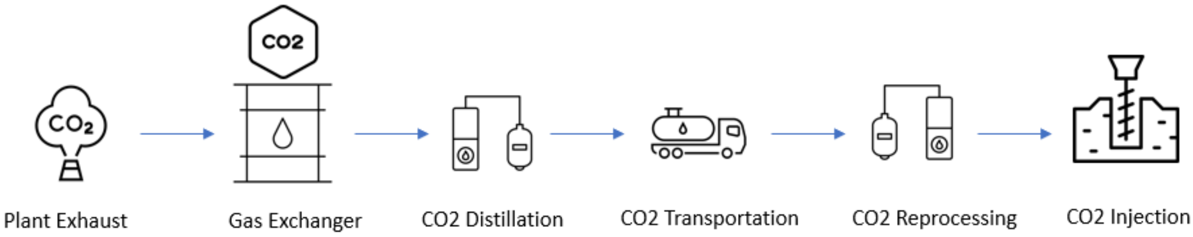


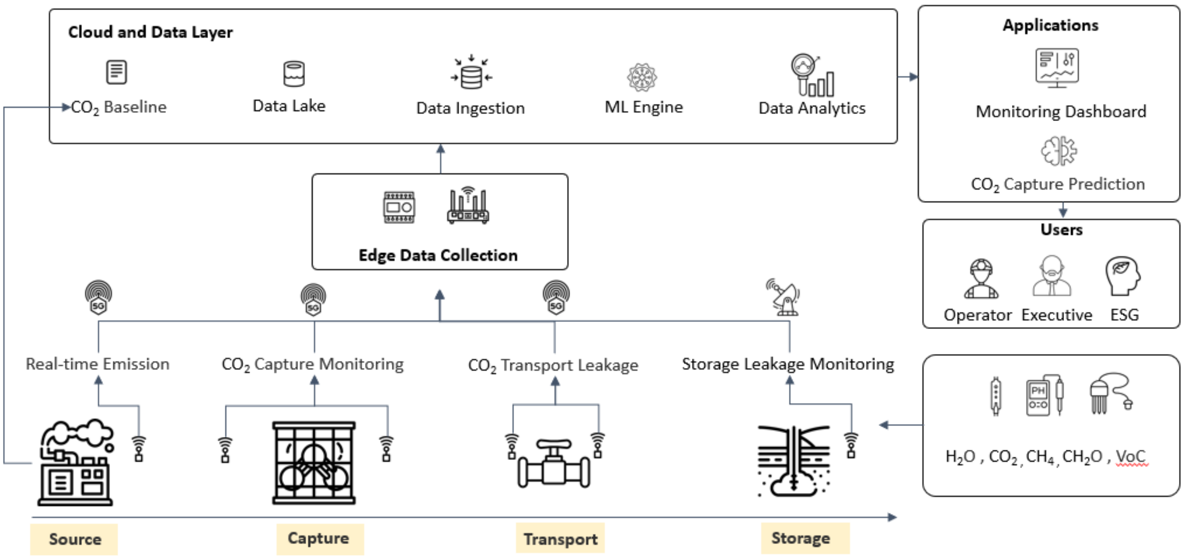
CO_2 is extracted from sources and converted to liquid form

The extracted CO_2 is transported via pipelines/rail/ship to onshore or offshore locations

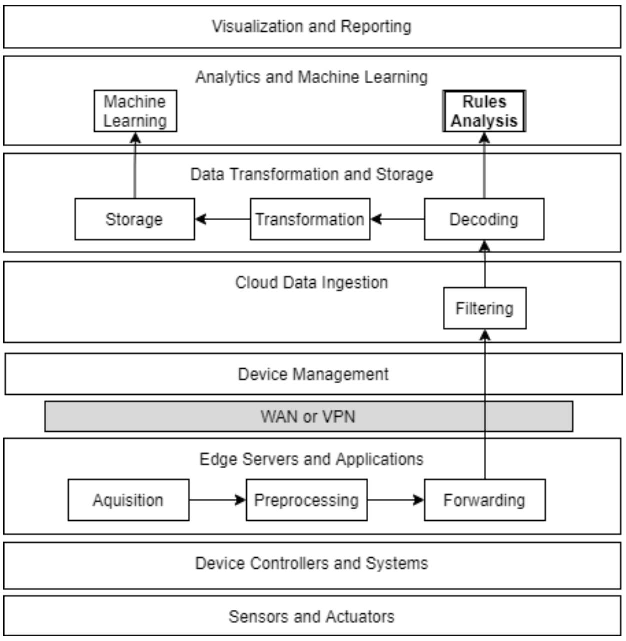
CO_2 is injected deep underground into the porous rocks for long term storage

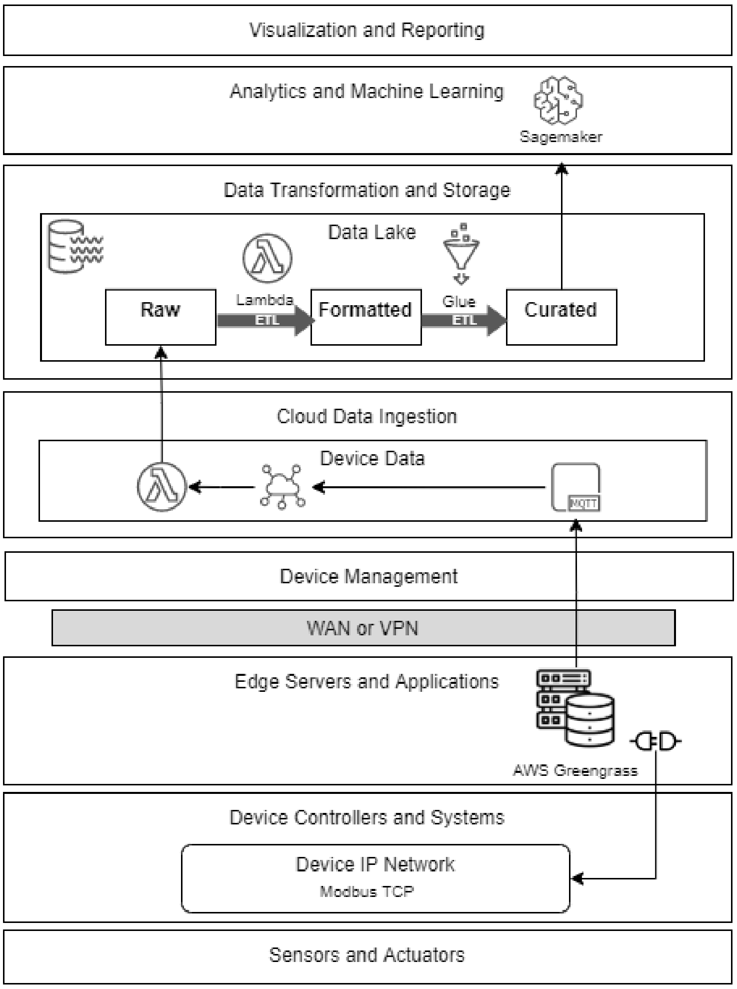
CO_2 is used for improving oil extraction in oil wells

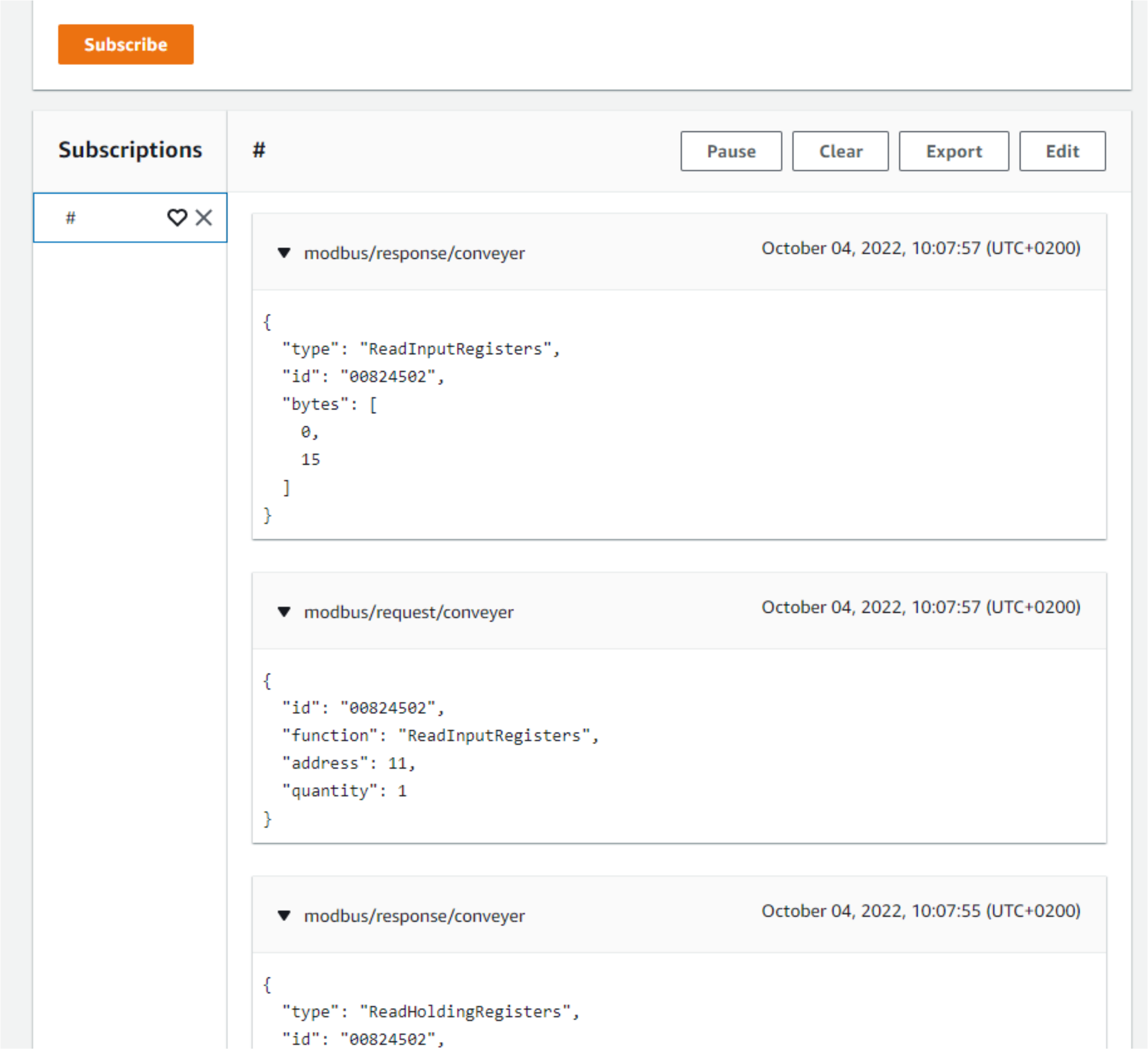




Chapter 12: Advanced Analytics and Machine Learning







modbus/response/conveyer — October 04, 2022, 10:07:57 (UTC+0200)

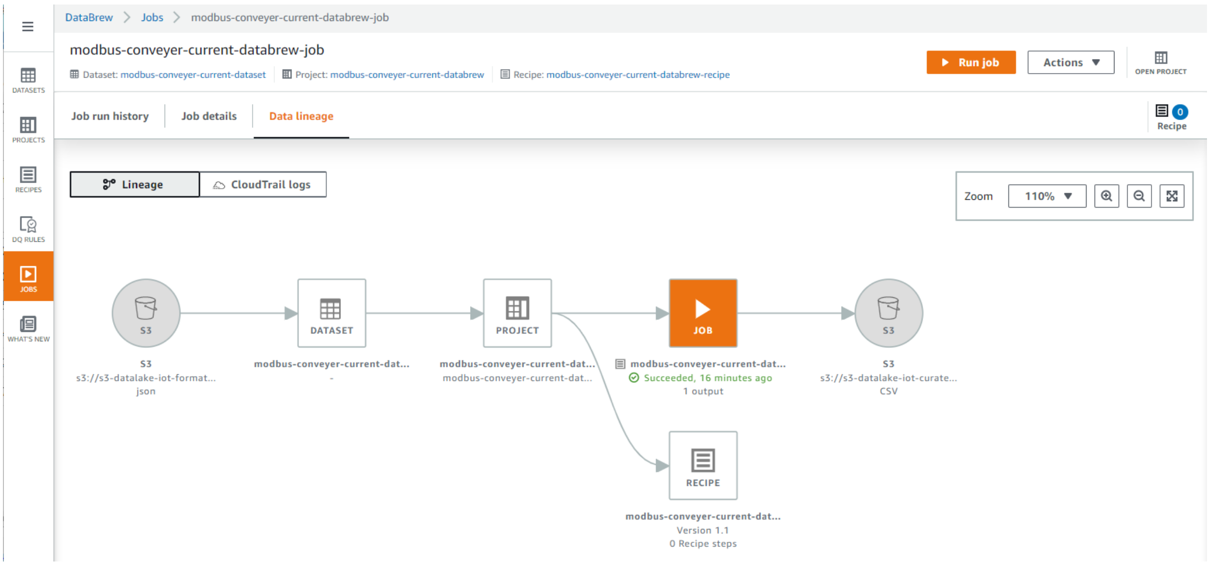
```
{
  "type": "ReadInputRegisters",
  "id": "00824502",
  "bytes": [
    0,
    15
  ]
}
```

modbus/request/conveyer — October 04, 2022, 10:07:57 (UTC+0200)

```
{
  "id": "00824502",
  "function": "ReadInputRegisters",
  "address": 11,
  "quantity": 1
}
```

modbus/response/conveyer — October 04, 2022, 10:07:55 (UTC+0200)

```
{
  "type": "ReadHoldingRegisters",
  "id": "00824502",
```



modbus-conveyer-current-databrew-job

Dataset: modbus-conveyer-current-dataset | Project: modbus-conveyer-current-databrew | Recipe: modbus-conveyer-current-databrew-recipe

S3
s3://s3-datalake-iot-format...
json

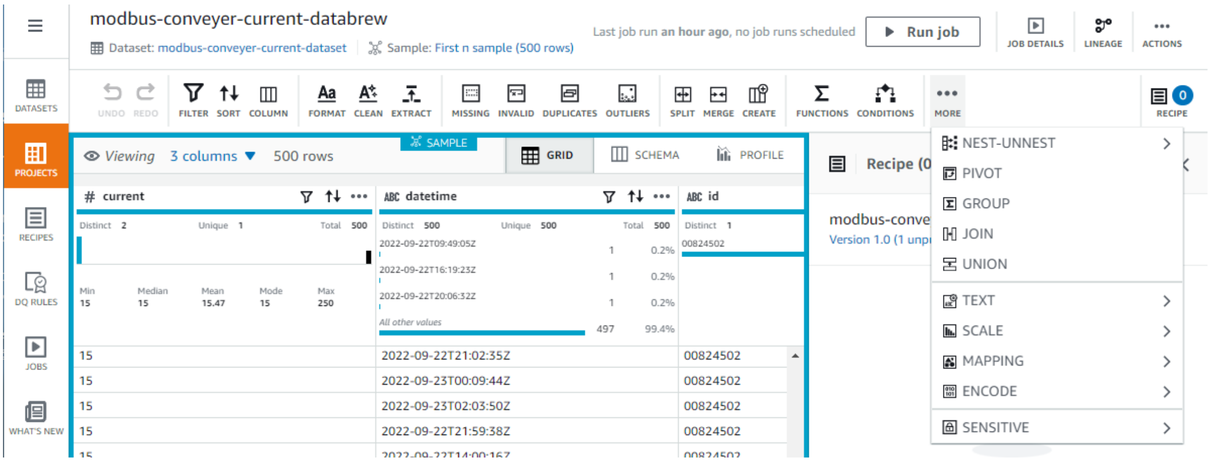
modbus-conveyer-current-dat...
-

modbus-conveyer-current-dat...
modbus-conveyer-current-dat...

modbus-conveyer-current-dat...
Succeeded, 16 minutes ago
1 output

S3
s3://s3-datalake-iot-curate...
CSV

modbus-conveyer-current-dat...
Version 1.1
0 Recipe steps



modbus-conveyer-current-databrew

Dataset: modbus-conveyer-current-dataset | Sample: First n sample (500 rows)

Last job run **an hour ago**, no job runs scheduled

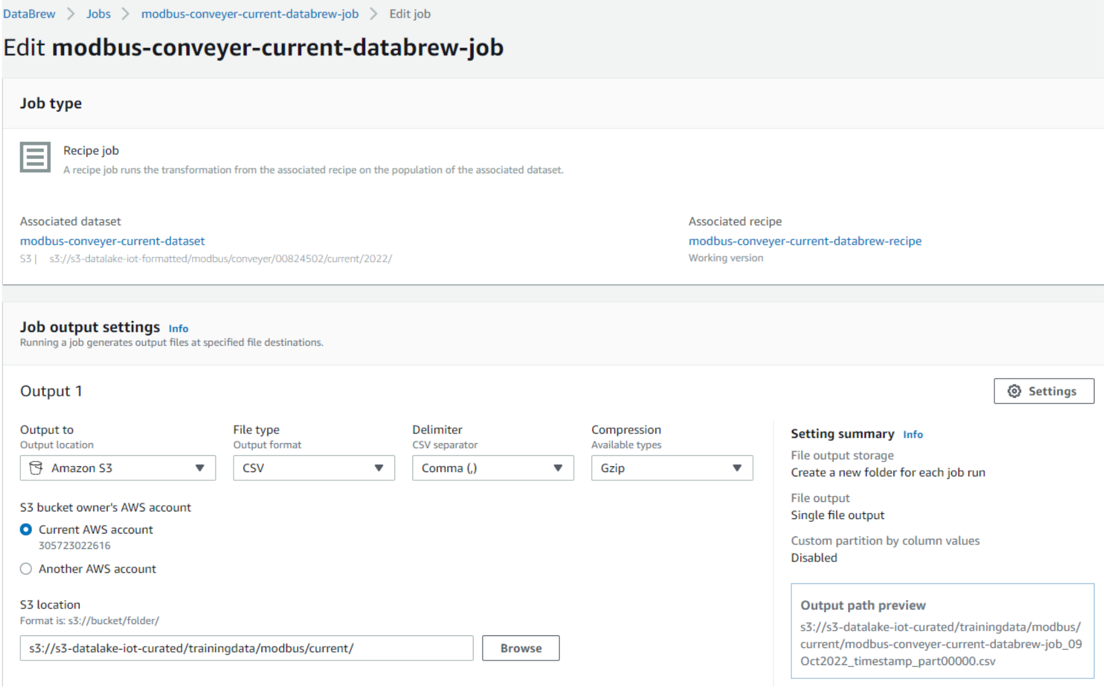
Viewing 3 columns ▼ 500 rows

Recipe (0

modbus-conve

Version 1.0 (1 unp

# current	ABC datetime	ABC id
Distinct 2 Unique 1 Total 500	Distinct 500 Unique 500 Total 500	Distinct 1
Min 15 Median 15 Mean 15.47 Mode 15 Max 250	2022-09-22T09:49:05Z 1 0.2% 2022-09-22T16:19:23Z 1 0.2% 2022-09-22T20:06:32Z 1 0.2% *All other values* 497 99.4%	00824502
15	2022-09-22T21:02:35Z	00824502
15	2022-09-23T00:09:44Z	00824502
15	2022-09-23T02:03:50Z	00824502
15	2022-09-22T21:59:38Z	00824502
15	2022-09-22T14:00:16Z	00824502



DataBrew > Jobs > modbus-conveyer-current-databrew-job > Edit job

Edit **modbus-conveyer-current-databrew-job**

Job type

Recipe job

A recipe job runs the transformation from the associated recipe on the population of the associated dataset.

Associated dataset

modbus-conveyer-current-dataset

S3 | s3://s3-datalake-iot-formatted/modbus/conveyer/00824502/current/2022/

Associated recipe

modbus-conveyer-current-databrew-recipe

Working version

Job output settings Info

Running a job generates output files at specified file destinations.

Output 1

Output to: Output location — Amazon S3

File type: Output format — CSV

Delimiter: CSV separator — Comma (,)

Compression: Available types — Gzip

S3 bucket owner's AWS account

● Current AWS account 305723022616

○ Another AWS account

S3 location

Format is: s3://bucket/folder/

s3://s3-datalake-iot-curated/trainingdata/modbus/current/

Setting summary Info

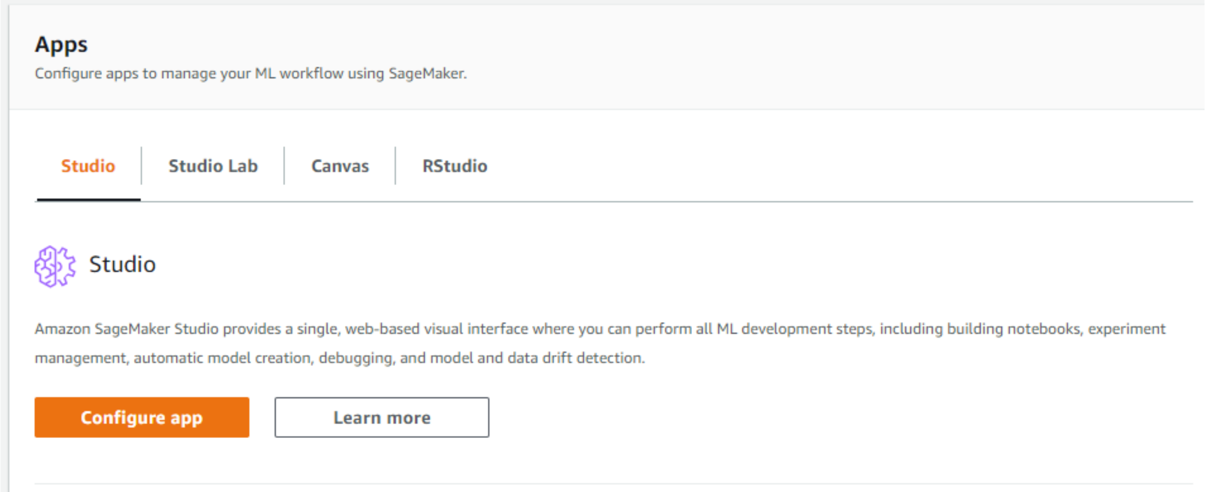
File output storage
Create a new folder for each job run

File output
Single file output

Custom partition by column values
Disabled

Output path preview

s3://s3-datalake-iot-curated/trainingdata/modbus/current/modbus-conveyer-current-databrew-job_09Oct2022_timestamp_part00000.csv

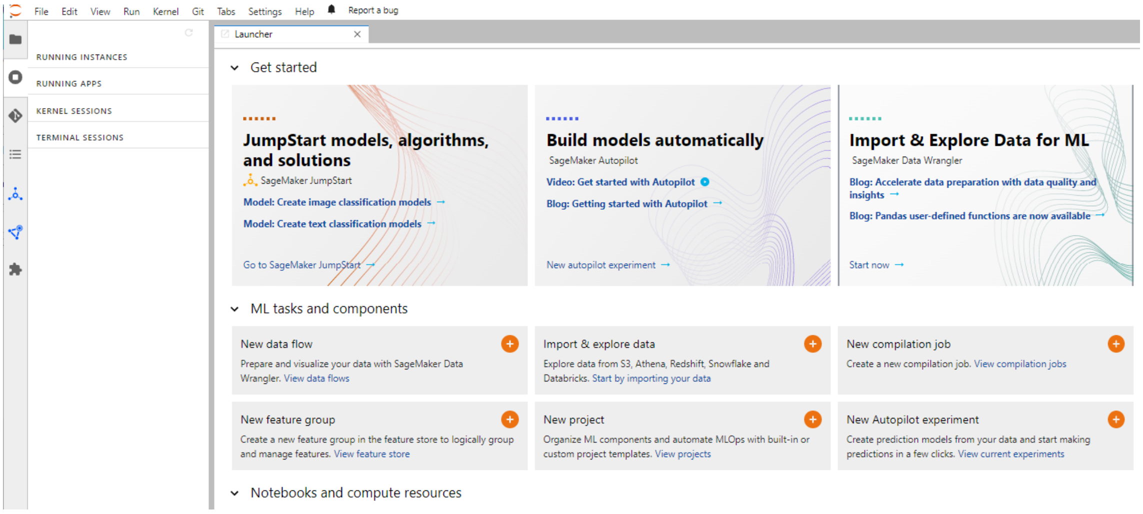


Apps

Configure apps to manage your ML workflow using SageMaker.

Studio

Amazon SageMaker Studio provides a single, web-based visual interface where you can perform all ML development steps, including building notebooks, experiment management, automatic model creation, debugging, and model and data drift detection.



Get started

JumpStart models, algorithms, and solutions

SageMaker JumpStart

Model: Create image classification models →

Model: Create text classification models →

Go to SageMaker JumpStart →

Build models automatically

SageMaker Autopilot

Video: Get started with Autopilot

Blog: Getting started with Autopilot →

New autopilot experiment →

Import & Explore Data for ML

SageMaker Data Wrangler

Blog: Accelerate data preparation with data quality and insights →

Blog: Pandas user-defined functions are now available →

Start now →

ML tasks and components

New data flow

Prepare and visualize your data with SageMaker Data Wrangler. View data flows

Import & explore data

Explore data from S3, Athena, Redshift, Snowflake and Databricks. Start by importing your data

New compilation job

Create a new compilation job. View compilation jobs

New feature group

Create a new feature group in the feature store to logically group and manage features. View feature store

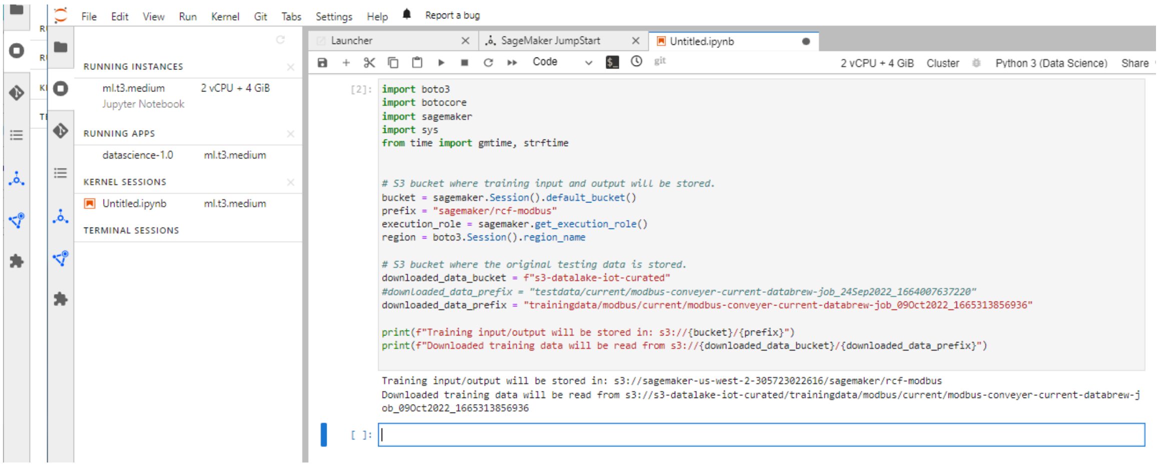
New project

Organize ML components and automate MLOps with built-in or custom project templates. View projects

New Autopilot experiment

Create prediction models from your data and start making predictions in a few clicks. View current experiments

Notebooks and compute resources



RUNNING INSTANCES

ml.t3.medium 2 vCPU + 4 GiB

Jupyter Notebook

RUNNING APPS

datascience-1.0 ml.t3.medium

KERNEL SESSIONS

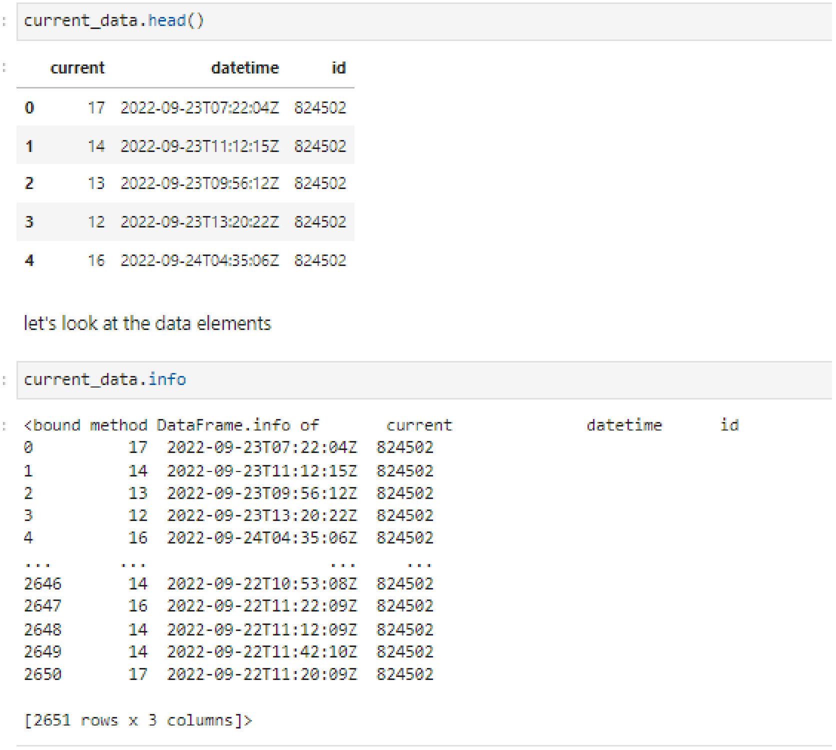
Untitled.ipynb ml.t3.medium

TERMINAL SESSIONS

```
[2]: import boto3
     import botocore
     import sagemaker
     import sys
     from time import gmtime, strftime


     # S3 bucket where training input and output will be stored.
     bucket = sagemaker.Session().default_bucket()
     prefix = "sagemaker/rcf-modbus"
     execution_role = sagemaker.get_execution_role()
     region = boto3.Session().region_name

     # S3 bucket where the original testing data is stored.
     downloaded_data_bucket = f"s3-datalake-iot-curated"
     #downloaded_data_prefix = "testdata/current/modbus-conveyer-current-databrew-job_24Sep2022_1664007637220"
     downloaded_data_prefix = "trainingdata/modbus/current/modbus-conveyer-current-databrew-job_09Oct2022_1665313856936"

     print(f"Training input/output will be stored in: s3://{bucket}/{prefix}")
     print(f"Downloaded training data will be read from s3://{downloaded_data_bucket}/{downloaded_data_prefix}")
```

```
Training input/output will be stored in: s3://sagemaker-us-west-2-305723022616/sagemaker/rcf-modbus
Downloaded training data will be read from s3://s3-datalake-iot-curated/trainingdata/modbus/current/modbus-conveyer-current-databrew-j
ob_09Oct2022_1665313856936
```

```
[ ]:
```



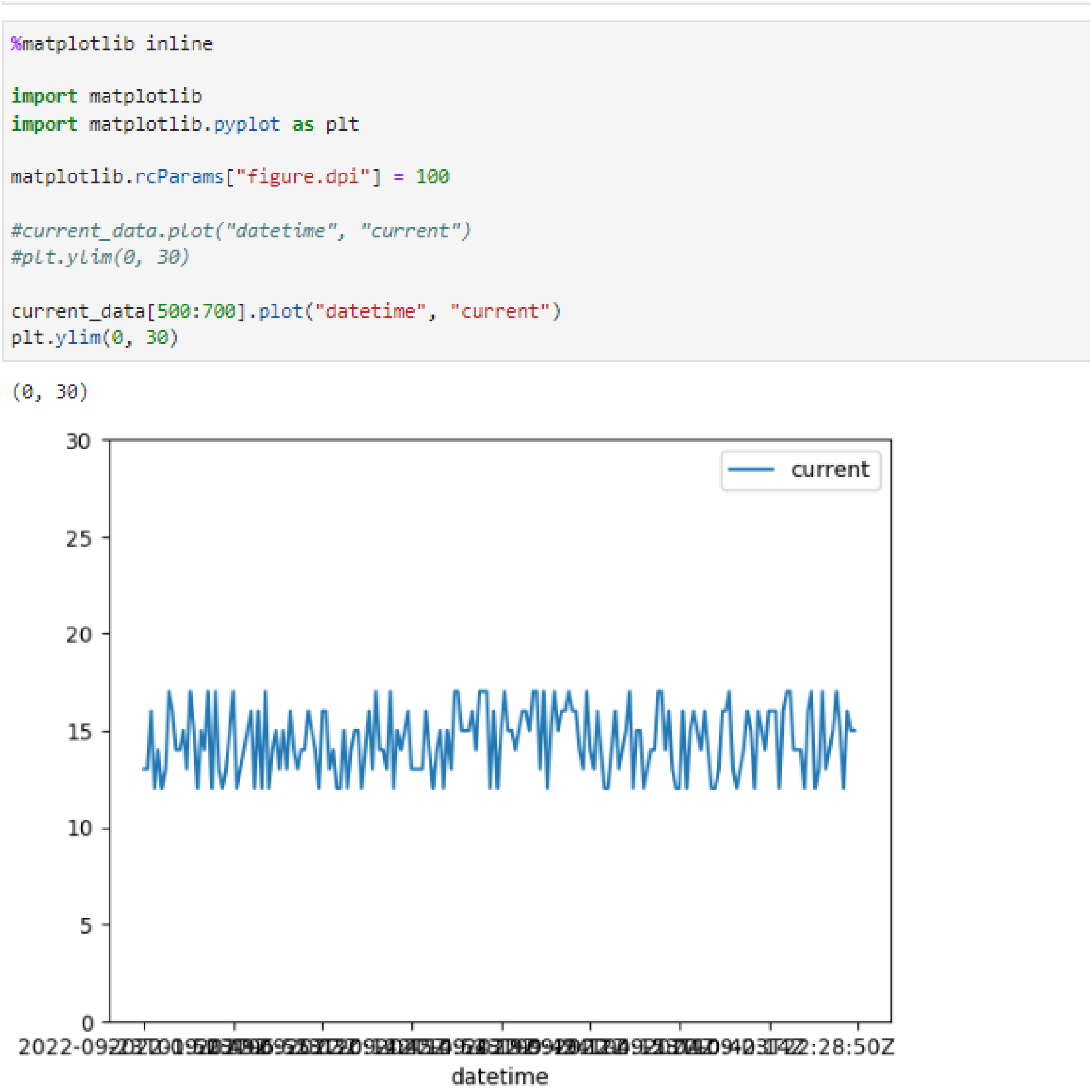
```
current_data.head()
```

	current	datetime	id
0	17	2022-09-23T07:22:04Z	824502
1	14	2022-09-23T11:12:15Z	824502
2	13	2022-09-23T09:56:12Z	824502
3	12	2022-09-23T13:20:22Z	824502
4	16	2022-09-24T04:35:06Z	824502

let's look at the data elements

```
current_data.info
```

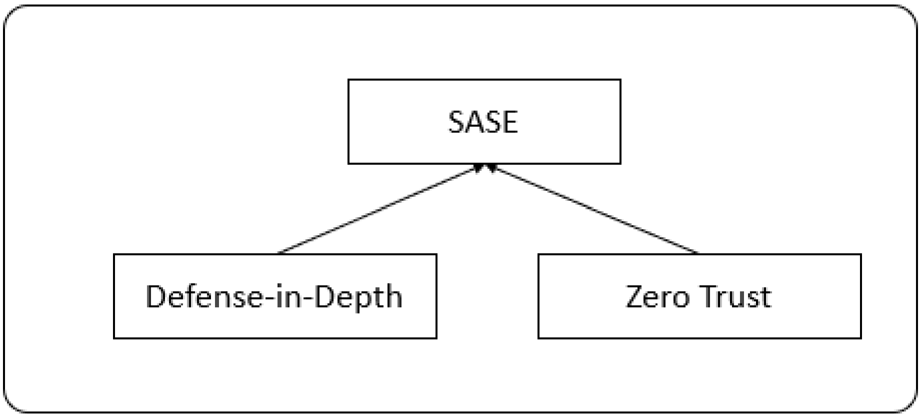
```
<bound method DataFrame.info of        current              datetime      id
0           17  2022-09-23T07:22:04Z  824502
1           14  2022-09-23T11:12:15Z  824502
2           13  2022-09-23T09:56:12Z  824502
3           12  2022-09-23T13:20:22Z  824502
4           16  2022-09-24T04:35:06Z  824502
...        ...                   ...     ...
2646        14  2022-09-22T10:53:08Z  824502
2647        16  2022-09-22T11:22:09Z  824502
2648        14  2022-09-22T11:12:09Z  824502
2649        14  2022-09-22T11:42:10Z  824502
2650        17  2022-09-22T11:20:09Z  824502

[2651 rows x 3 columns]>
```

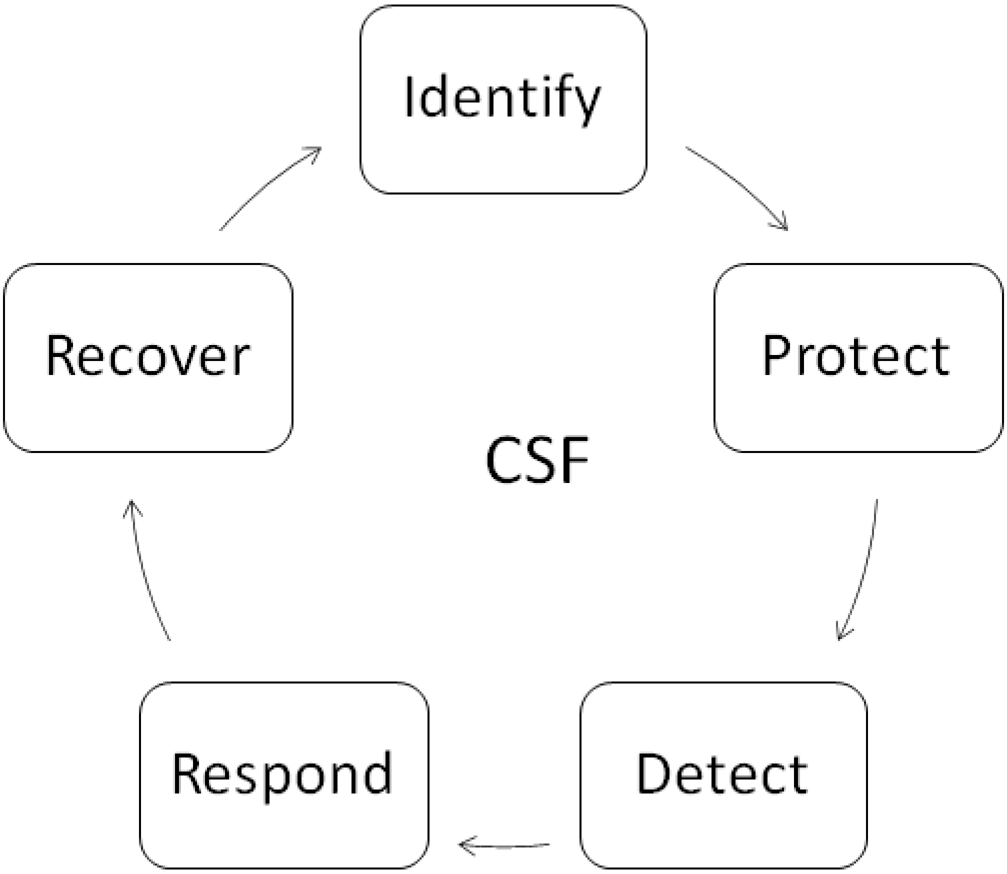


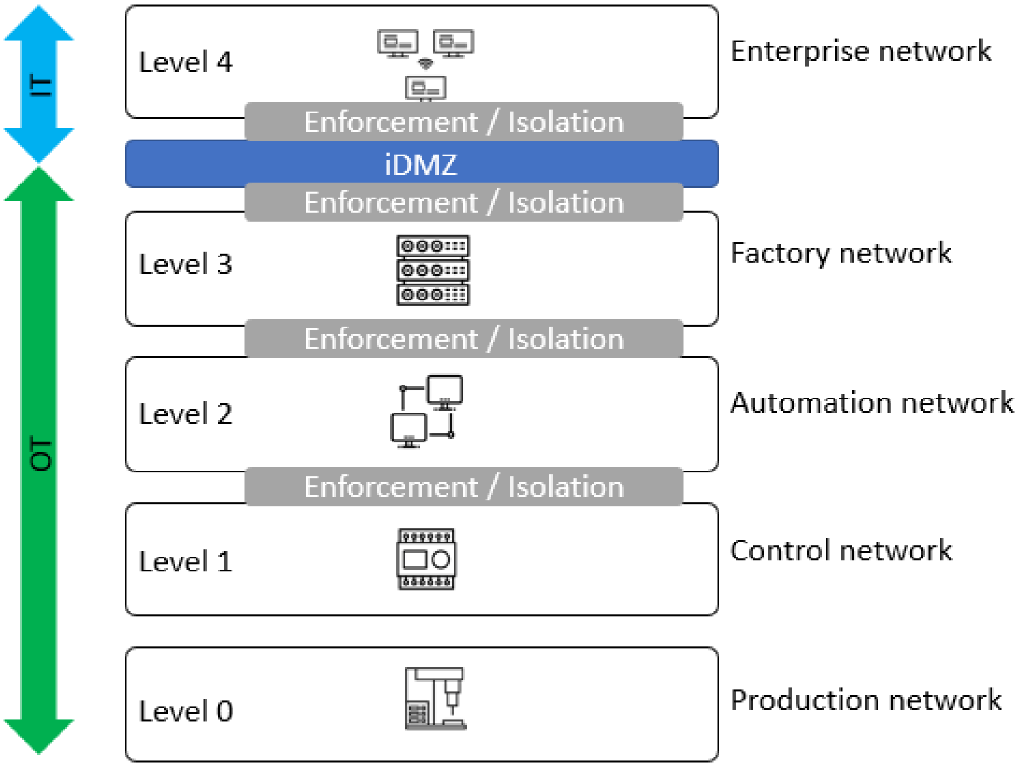
```
%matplotlib inline

import matplotlib
import matplotlib.pyplot as plt

matplotlib.rcParams["figure.dpi"] = 100

#current_data.plot("datetime", "current")
#plt.ylim(0, 30)

current_data[500:700].plot("datetime", "current")
plt.ylim(0, 30)
```

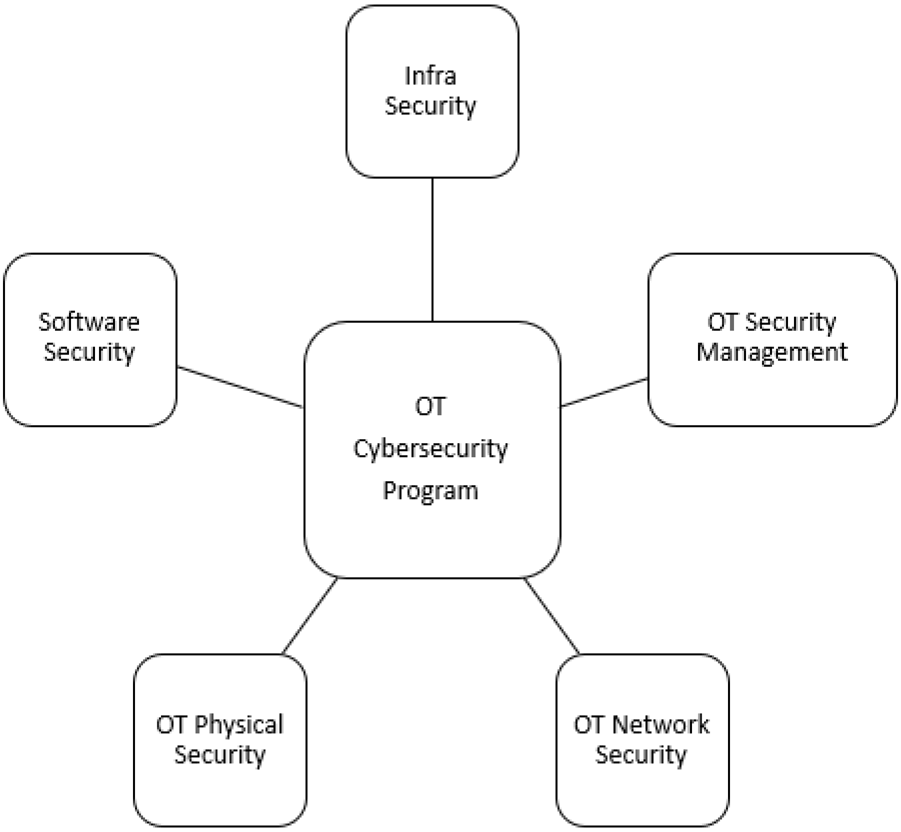
```
(0, 30)
```

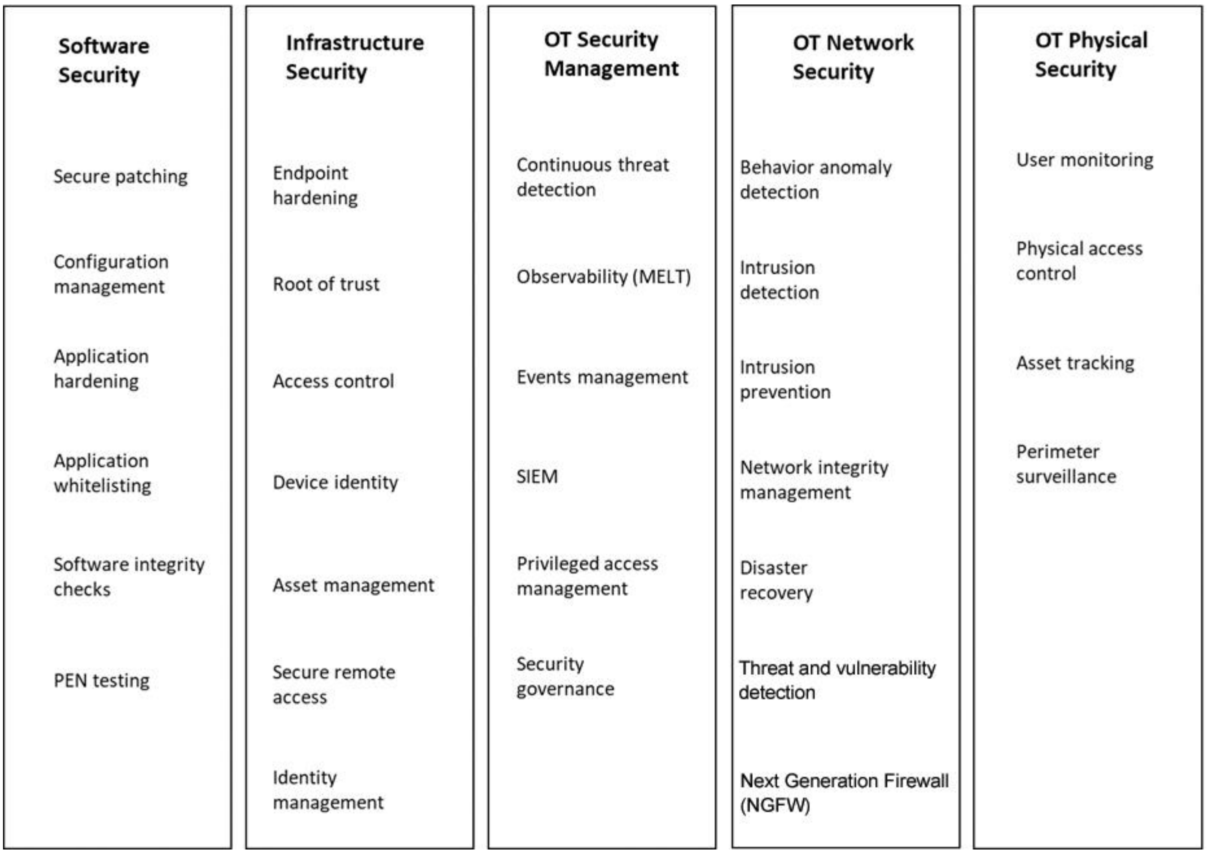
Appendix 1 - General Cybersecurity Topics



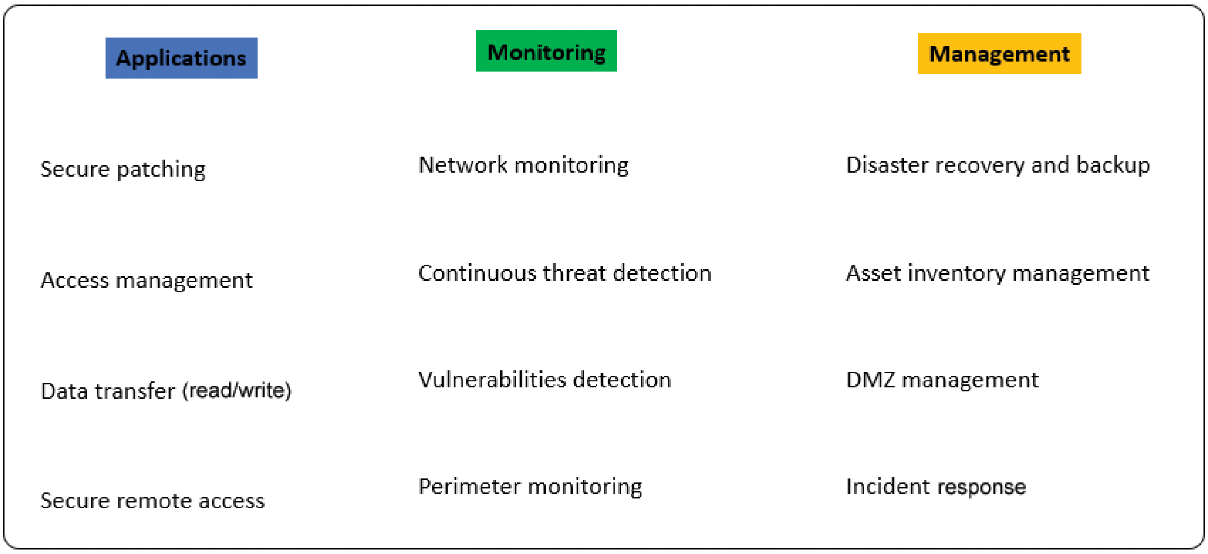



IT
OT
Level 4
Enterprise network
Enforcement / Isolation
iDMZ
Enforcement / Isolation
Level 3
Factory network
Enforcement / Isolation
Level 2
Automation network
Enforcement / Isolation
Level 1
Control network
Level 0
Production network


Infra Security
Software Security
OT Security Management
OT Cybersecurity Program
OT Physical Security
OT Network Security



Software Security	Infrastructure Security	OT Security Management	OT Network Security	OT Physical Security
Secure patching	Endpoint hardening	Continuous threat detection	Behavior anomaly detection	User monitoring
Configuration management	Root of trust	Observability (MELT)	Intrusion detection	Physical access control
Application hardening	Access control	Events management	Intrusion prevention	Asset tracking
Application whitelisting	Device identity	SIEM	Network integrity management	Perimeter surveillance
Software integrity checks	Asset management	Privileged access management	Disaster recovery	
PEN testing	Secure remote access	Security governance	Threat and vulnerability detection	
	Identity management		Next Generation Firewall (NGFW)	



Applications	Monitoring	Management
Secure patching	Network monitoring	Disaster recovery and backup
Access management	Continuous threat detection	Asset inventory management
Data transfer (read/write)	Vulnerabilities detection	DMZ management
Secure remote access	Perimeter monitoring	Incident response