I2PANEMA

State of the Art

Deliverable No.	D.1.5	**Due Date**	01/05/2020
Type	Report	**Dissemination Level**	Public
Version	1.0	**Status**	
Description	State of the Art		
Work Package	WP1		

Authors

Name	Partner	e-mail
Christophe Joubert	PRO	cjoubert@prodevelop.es
José A. Clemente	PRO	jclemente@prodevelop.es
Metin Tekkalmaz	ERSTE	metin@ersteyazilim.com
Yildiray Kabak	SRDC	yildiray@srdc.com.tr
Michael Nast	URO	michael.nast@uni-rostock.de
Benjamin Rother	URO	benjamin.rother@uni-rostock.de
Kiran Shekhar	NXPGE	Kiran.shekhar@nxp.com
Ksenia Winkler	CATKIN	Ksenia.Winkler@catkin.eu
Jens Leveling	IML	jens.leveling@iml.fraunhofer.de
Christian Olms	IML	Christian.olms@iml.fraunhofer.de

History

Date	Version	Change
31-March-2019	0.1	ToC
18-Nov-2019	0.2	First draft
3-March-2020	0.3	Second draft
25-March-2020	0.4	Third / final draft for Internal Review
29-April-2020	1.0	Ready for submit

Key Data

Keywords	I2PANEMA, State of the Art, Reference Architecture, IoT
Lead Editor	José A. Clemente, PRO
Internal Reviewer(s)	Benjamin Rother (URO), Valentin Mees (LBF)

Abstract

The present document is a State of the Art of the different layers of which the I2PANEMA platform is composed.

The document firstly describes the different reference architectures for the Internet of Things (IoT) available, describing them briefly and selecting those more relevant given the scope of the project. These three are RAMI 4.0 and IIRA for their European and global relevance and IoT-A for its orientation to interoperability, a main requirement of I2PANEMA. Another important topic for I2PANEMA is the security. This is the reason to speak about International Data Spaces (IDS), an international initiative to exchange information in business environments.

Due to the growth of IoT, it is impossible to not speak about IoT Cloud platforms.

The next sections describe the State of the Art for each functional block:

- IoT Communication
- IoT Interoperability
- IoT Data Management
- IoT Operational Workflow
- IoT Visualization
- IoT Security & Privacy

The final chapter contains the conclusions of the document.

Statement of originality

This document contains material, which is the copyright of certain I2PANEMA consortium parties, and may not be reproduced or copied without permission. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

The information contained in this document is the proprietary confidential information of the I2PANEMA consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the project consortium as a whole nor a certain party of the consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

The information in this document is subject to change without notice.

Table of Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Explanation
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
AWS	Amazon Web Services
BPMN	Business Process Model and Notation
CCSA	China Communications Standards Association
CoAP	Constrained Application Protocol
CREATE-IoT	Cross fertilization through Alignment, Synchronization and Exchanges for IoT
EPI	European Platforms Initiative
HTTP	Hypertext Transfer Protocol
IIC	Industrial Internet Consortium
IIoT	Industrial IoT
IoT-A	Internet of Things Architecture
IoT	Internet of Things
IIRA	Industrial Internet Reference Architecture
ITU	International Telecommunication Union
LoRa	Long Range
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NIF	National ICT Interoperability Framework
OGC	Open Geospatial Consortium
PdM	Predictive Maintenance
QoS	Quality of Service
RA	Reference Architecture
RAMI 4.0	Reference Architectural Model Industry 4.0
RIS	River Information Services
SDW	Spatial Data on the Web
SNS	Simple Notification Service
SOA	Service-oriented Architecture
SoS	System-of-Systems
SSN	Semantic Sensor Network
WISE IoT	Worldwide Interoperability for Semantics IoT
WP	Work Package
W3C	World Wide Web Consortium
XMPP	Extensible Messaging and Presence Protocol

1. Introduction

The scope of the deliverable is to present the different types of IoT architectures that exist at the moment in order to present the different trends **Generic** and **Industrial** IoT architectures.

With the rise of **cloud platforms**, there are a large number of companies offering their IoT platform in the cloud. This is why this option must be taken into account and a point has been included where the most important ones will be mentioned.

A section has also been included about International Data Spaces (IDS). The relevance of this Reference Architecture (RA) is that it provides a secure data exchange in business ecosystems.

Once the different approaches have been presented, the document also includes one chapter for each of the different modules that constitute our project. The idea is to present a **State of the Art** of the different layers.

The document is intended to be used as a reference for software developers to have a general vision of the state of the art of the different solutions per module. This document will be used as a basis for the development of the upcoming deliverables, as for example: **D1.3 I2PANEMA Reference Architecture**.

2. IoT Architectures

The rise of the IoT concept (Kevin Ashton in 2009) has brought an increase in the use of information and communication technologies. This involves the need to search for a valid reference IoT architecture that supports the different environments and contexts of the world of things (person-to-machine or M2M, machine-to-machine). There have been several proposed architectures, many of them defined in specific contexts and providing solutions to a part of the "world of things".

The concept of Reference Architecture (RA) refers to the component and services layout and best practices of an IT system that is likely to be implemented recurrently with similar objectives but different contexts, constraints or business variations. An example could be the framework developed as a pattern with the intention of creating applications that would share the same pattern, characteristics, etc. In this case, instead of focusing on application (web, desktop, mobile), development would be focused on IoT scenarios (Perry, 2018; Fremantle, 2015).

Thus, establishing a RA and architectural patterns is a good practice in the scenarios faced in I2PANEMA. A RA for IoT in ports will lay the foundation for a common framework for the development of future systems and contributing in the medium term to reducing costs for ports compared to each of the individual solutions currently available. Other reasons for defining a RA are the following[1]:

- IoT devices are inherently connected. We need a way to uniformly interact with them (**I2PANEMA Interoperability**).
- There are billions of devices in the market and the number is growing quickly. So, it became necessary to have an architecture for scalability. Requirements vary between deployment events of the same technologies and they also change during the time.
- Management of devices (automatic updates, remote management) is needed, and these devices can change, evolve, be deprecated, substituted, etc.
- These devices collect sensitive data, therefore it is necessary to establish a security layer (**I2PANEMA Security**) that controls the communication between devices or with the platform that receives the data. Security protocols, patterns and technologies change across devices and time.
- A RA provides a starting point for architects looking to create IoT solutions as well as a strong basis for further development (**I2PANEMA**)

The use of a RA provides stability and reliability of the designed solution across multiple scenarios (as in I2PANEMA) and over time.

[1] Paul Fremtale. A Reference Architecture for the Internet of Things https://www.researchgate.net/publication/308647314_A_Reference_Architecture_for_the_Internet_of_Things

It is important to distinguish that a RA is more abstract than a system architecture that has been developed for a specific set of applications (I2PANEMA) with particular constraints and scenarios[2].

A RA for IoT is more complex than a traditional architecture due to the heterogeneity of concepts and technologies and due to relationships between the different technologies used. It is important to understand the impact on scalability and other parts of the system when choosing a certain design aspect (this is one of the factors to take into account in D1.4, System Design Document).

According to the book: "Internet of Things for Architects" (Perry, 2018), currently there are over 1.5 million different combinations of architectures to choose from.

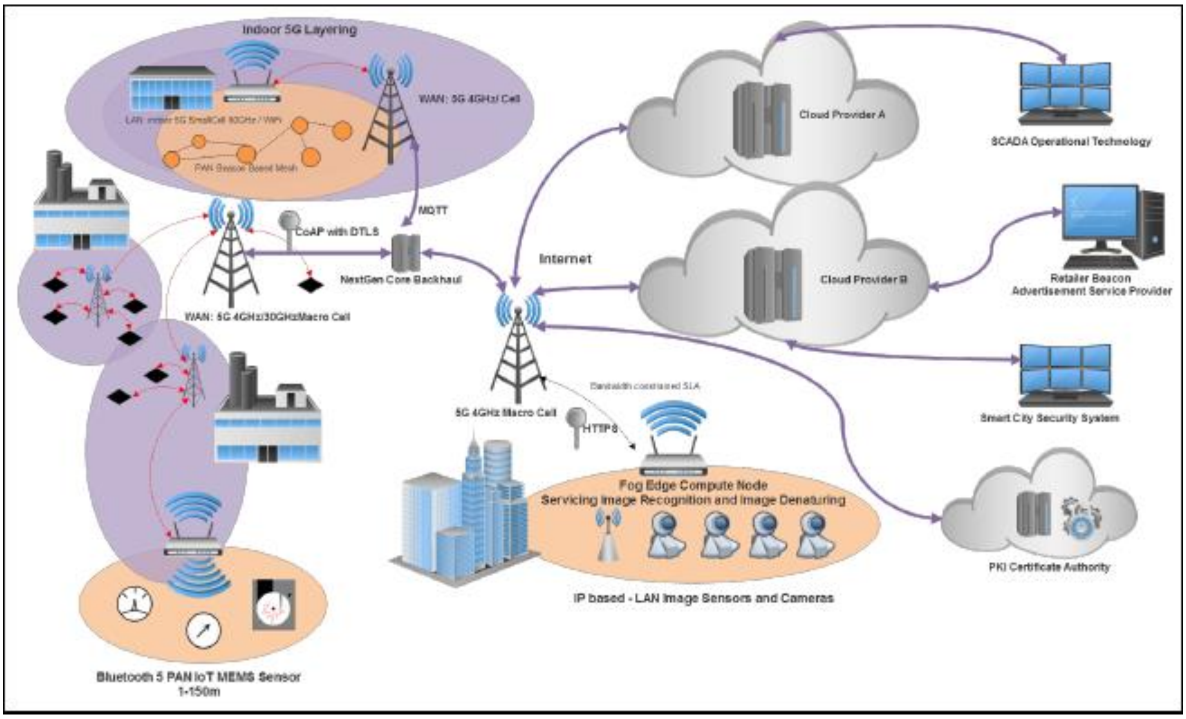

Indoor 5G Layering
LAN: Indoor 5G SmallCell 30GHz / WiFi
WAN: 5G 4GHz/ Cell
PAN Sensor Based Mesh
MQTT
CoAP with DTLS
NextGen Core Backhaul
Internet
WAN: 5G 4GHz/30GHzMacro Cell
Cloud Provider A
SCADA Operational Technology
Cloud Provider B
Retailer Beacon
Advertisement Service Provider
Smart City Security System
Bandwidth constrained SLA
5G 4GHz Macro Cell
HTTPS
Fog Edge Compute Node
Servicing Image Recognition and Image Denaturing
IP based - LAN Image Sensors and Cameras
PKI Certificate Authority
Bluetooth 5 PAN IoT MEMS Sensor
1-150m

Figure 1: IoT Design Choices

The full spectrum of various levels of IoT architecture from sensor to cloud and back

Currently, there are two main types of IoT architectures:

1. Generic ones, like IoT-A and WSO2, for example.
2. Industrial ones (**IIOT**), like RAMI 4.0 and IIRA.

These architectures will be detailed in the following subsections for both types (Atefeh Torkaman, 2016).

However, in recent years many service providers (Amazon, Google, Oracle, etc.) have begun to offer the services of their IoT platforms in the cloud. This trend can solve problems, as for example the size of computing resources and bring new issues, e.g. regarding security. It will be explained in more detail in section IoT Cloud Platforms.

2.1. Generic architectures

2.1.1. IoT-A

IoT-A[3] was a lighthouse EU-funded project (2010 - 2013) that established an Architectural Reference Model for the Internet of Things domain. The project can be considered the foundation for all the EU efforts done in this area since then.

[2] https://iotforum.org/wp-content/uploads/2014/09/120613-IoT-A-ARM-Book-Introduction-v7.pdf

[3] https://cordis.europa.eu/project/rcn/95713/factsheet/en

The IoT-A project developed common tools and methodologies to achieve a complete reference architecture for the existing and forthcoming IoT scenarios. It presents a series of characteristics that must be taken into consideration for the analysis of IoT architectures:

- **Architecture description**. Addresses the vision of architecture in relation to the project / product.
- **Model and distribution of information**. Addresses the problem of how information is treated and how it is distributed in the system.
- **Horizontality**. Ability to reuse the same blocks to provide different functionalities of the top layer.
- **Context knowledge and semantic capabilities**. Refers to the possibility of improving the information exchanged through semantic translators.
- **Technology specification and interoperability**. How much of the project / product depends on a particular technology and how to focus on interoperability.
- **Adaptation**. Capacity offered by the project / product regarding reactivity to environmental changes.
- **Programmability**. Defines APIs and specific standards for the development of an application.
- **Interface with the outside world**. Interfaces with which the end user interacts.
- **Work Plan**. Consideration of whether a product includes in its planning measures aimed at improving compatibility and / or development from the perspective of IoT.
- **Can the product interact with other articles?** Capacity or flexibility that a product presents when interacting with elements of the environment.
- **Obvious aspects of integration**. Evaluates whether a given product has taken into consideration basic aspects for integration into IoT architecture.

The IoT-A project has generated a series of results among which are the specifications of a reference architecture for the IoT. This RA has provided the understanding of an open architecture for the IoT and fully covers security and privacy issues as well as scalability and interoperability among other aspects.

Apart from research and private entities, as it will be shown below, standardization organisms have done work in this domain. ITU[4] has proposed a comprehensive reference model in IoT environment. Their contribution has been the recommendation ITU-T Y.2060[5] that clarifies the concept and scope of the IoT, identifying the fundamental characteristics and high-level requirements and describing the IoT reference model[6].

Figure 2 depicts the functional model of IoT-A.

[4] https://www.itu.int
[5] https://www.itu.int/rec/T-REC-Y.2060/en
[6] http://ijcsse.org/published/volume5/issue8/p1-V518.pdf

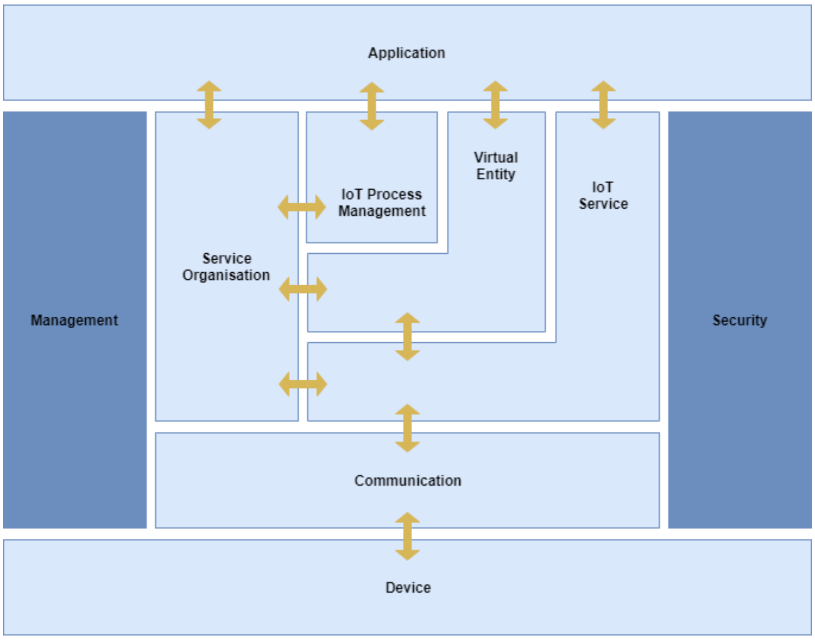


Figure 2: IoT-A's functional model

These are the functions of the different modules that compose the IoT's functional model:

- **Security**. Among its responsibilities are: Authorization, authentication, identity management.
- **Management**. It is responsible of: Configuration, State and Reporting.
- **IoT Process Management**. Responsible for: Process Modelling and Process Execution.
- **Service Organization**. Responsible for: Service Orchestration.
- **IoT Service**. Module responsible for: Registration of devices, historical data access, etc.
- **Virtual Entity**. It is responsible to handle the relations between the virtual entities (devices, sensors, etc.) and domain entities.
- **Communication**.

IoT-A introduces the concepts of views and perspectives. Its design is as follows and shown in Figure 3:

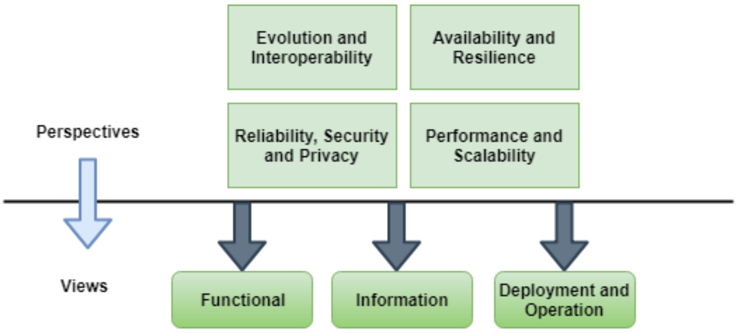

Evolution and Interoperability
Availability and Resilience
Perspectives
Reliability, Security and Privacy
Performance and Scalability
Views
Functional
Information
Deployment and Operation

Figure 3: IoT-A's views and perspectives

- **Views**. "Different angles for viewing an architecture that can be used when designing and implementing it"[7]. The views include: Functional view, Information view, Deployment and operation view, as seen in the image above.
- **Perspectives**. "Set of tasks, tactics, directives, and architectural decisions for ensuring that a given concrete system accomplishes one or more quality attributes"[8].

[7] http://www.iot-a.eu/

[8] http://www.iot-a.eu/

2.1.2. IoT Reference Architecture (WSO2)

This RA as shown in Figure 4 has been developed by the company **WSO2**[9]. Its proposal is based in their experience in the development of IoT solutions.

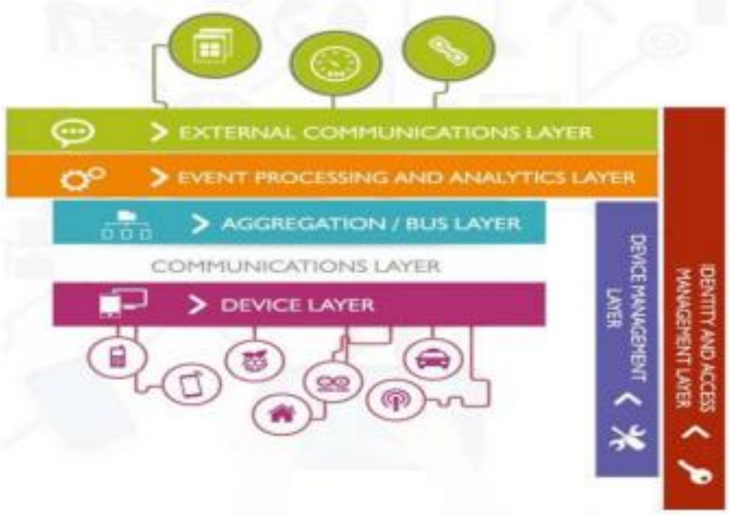

EXTERNAL COMMUNICATIONS LAYER
EVENT PROCESSING AND ANALYTICS LAYER
AGGREGATION / BUS LAYER
COMMUNICATIONS LAYER
DEVICE LAYER
DEVICE MANAGEMENT LAYER
IDENTITY AND ACCESS MANAGEMENT LAYER

Figure 4: WSO2's reference architecture

The layers are:

- External Communications Layer (Web/Portal, Dashboard, APIs)
- Event Processing and Analytics Layer
- Aggregation / Bus Layer (ESB and message broker)
- Communications Layer (MQTT, HTTP, XMPP, CoAP, AMQP, etc.)
- Devices

It also has cross-cutting layers:

- Device Manager
- Identity and Access Management

2.1.3. Korean IoT Reference Model

The **Korean Study Group** specification establishes a RA from communication and functional viewpoint as depicted in Figure 5.

[9] https://wso2.com

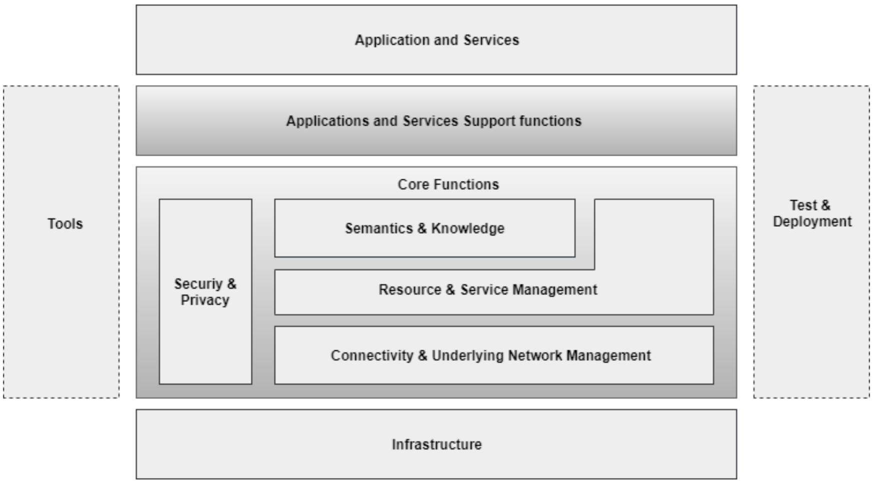

Application and Services
Applications and Services Support functions
Tools
Core Functions
Securiy & Privacy
Semantics & Knowledge
Resource & Service Management
Connectivity & Underlying Network Management
Test & Deployment
Infrastructure

Figure 5: Functional view of IoT RA (Korean IoT Reference Model)

Basically this architecture consists of six blocks (thinking in high level functional blocks):

- Application and Services.
- Application and Services Support functions.
- Tools.
- Test & Deployment.
- Core functions.
- Infrastructure.

2.1.4. Chinese IoT Reference Model

CCSA[10] (Shanzhi Chen, 2014) has proposed a RA which consists of several layers as shown in Figure 6:

- Sensing layer. Connects sensors, controllers, RFID readers, and location sensing devices to IoT network layer.
- Network and service layer. Includes backbone networks and resource administration platforms.
- Application layer. Includes various applications in IoT system.

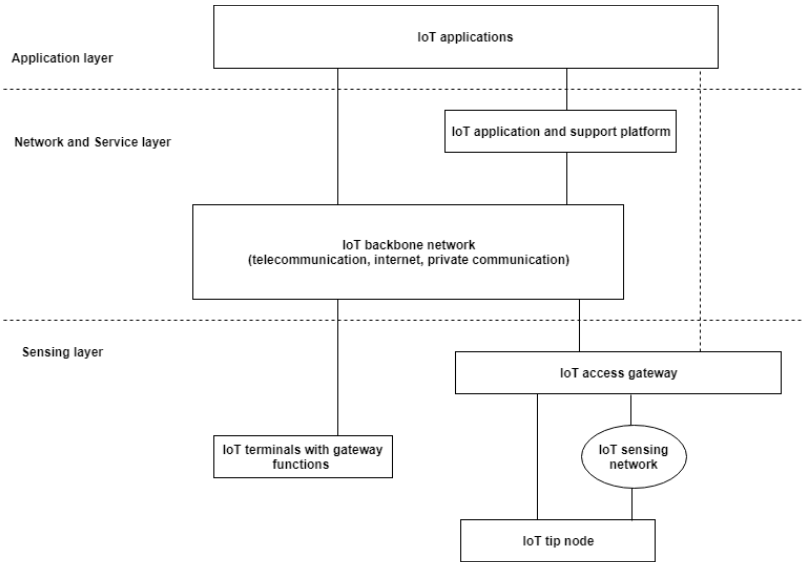

Application layer
IoT applications
Network and Service layer
IoT application and support platform
IoT backbone network
(telecommunication, internet, private communication)
Sensing layer
IoT access gateway
IoT terminals with gateway functions
IoT sensing network
IoT tip node

Figure 6: IoT reference architecture proposed by CCSA

[10] http://www.ccsa.org.cn/english/

2.1.5. ETSI IoT Reference Model

ETSI[11] high level reference architecture as shown in Figure 7 is built on two domains:

1. **The Device and Gateway Domain:** M2M area network is the connection between M2M Devices and M2M Gateways and it includes Personal Area Network technologies (IEEE 802.15.1, ZigBee, Bluetooth) or local networks (PLC, M-BUS, Wireless M-BUS, KNX). (Etsi.org, 2019).

2. **The Network Domain:** Access network allows the M2M Device and Gateway Domain to communicate with the core network which provides IP connectivity, service and network control functions, interconnection with other networks, and roaming. Different Core Networks offer different features sets and Core Networks include 3GPP CNs, ETSI TISPAN CN and 3GPP2 CN. (Etsi.org, 2019).

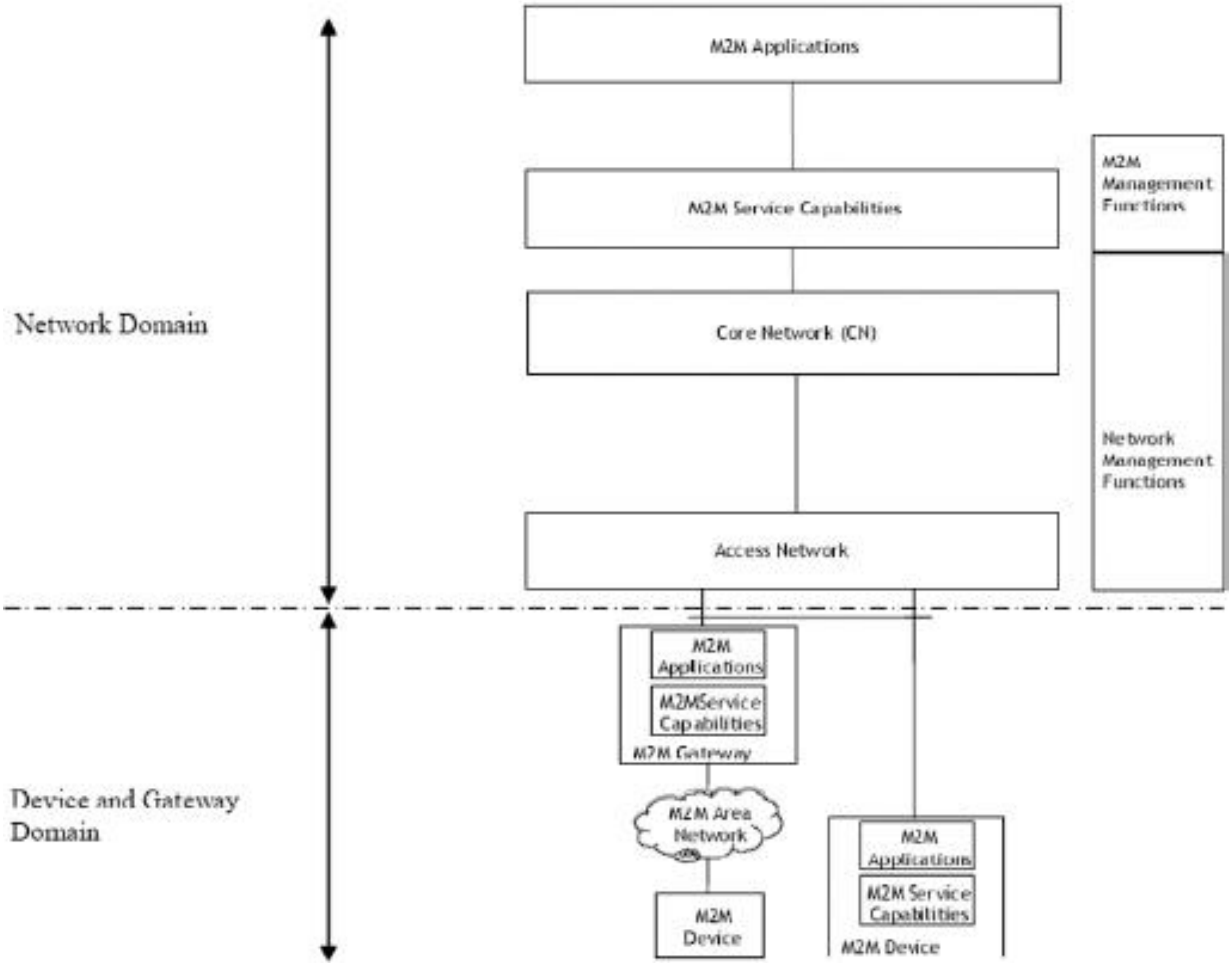


Figure 7: ETSI IoT Reference Architecture. (Source: Etsi.org, 2019)

2.1.6. GISFI IoT Reference Model

GISFI is the acronym for The **Global ICT Standardization Forum for India**. The GISFI[12] IoT reference architecture is composed of several layers (devices, gateway and service platform) as shown in Figure 8:

[11] https://www.etsi.org/
[12] https://www.gisfi.org

- **IoT Device Layer.** IoT devices like sensors, actuators (including heterogeneous devices), network-enabled objects, and small data source networks. Different communication standards like ZigBee, ZWave, ANTS, Wi-Fi, etc. are supported. (Gisfi.org, 2019).
- **IoT Gateway Layer:** As the name suggests, the layer consists of IoT gateways. This layer helps abstraction of Device Layer, which hosts heterogeneous devices, and provides a more uniform interface to the IoT Service Platform Layer. (Gisfi.org, 2019).
- **IoT Service Platform Layer:** By defining different IoT service abstractions to be used by multiple applications this layer can be extended to application services (Gisfi.org, 2019).

The Device Layer can communicate with the devices for information exchange using various legacy technologies. However, the Gateway and Service Platform Layers are expected to be connected over an IoT Core/Backbone network which is thought to be an IP based network having IoT capabilities. (Gisfi.org, 2019)

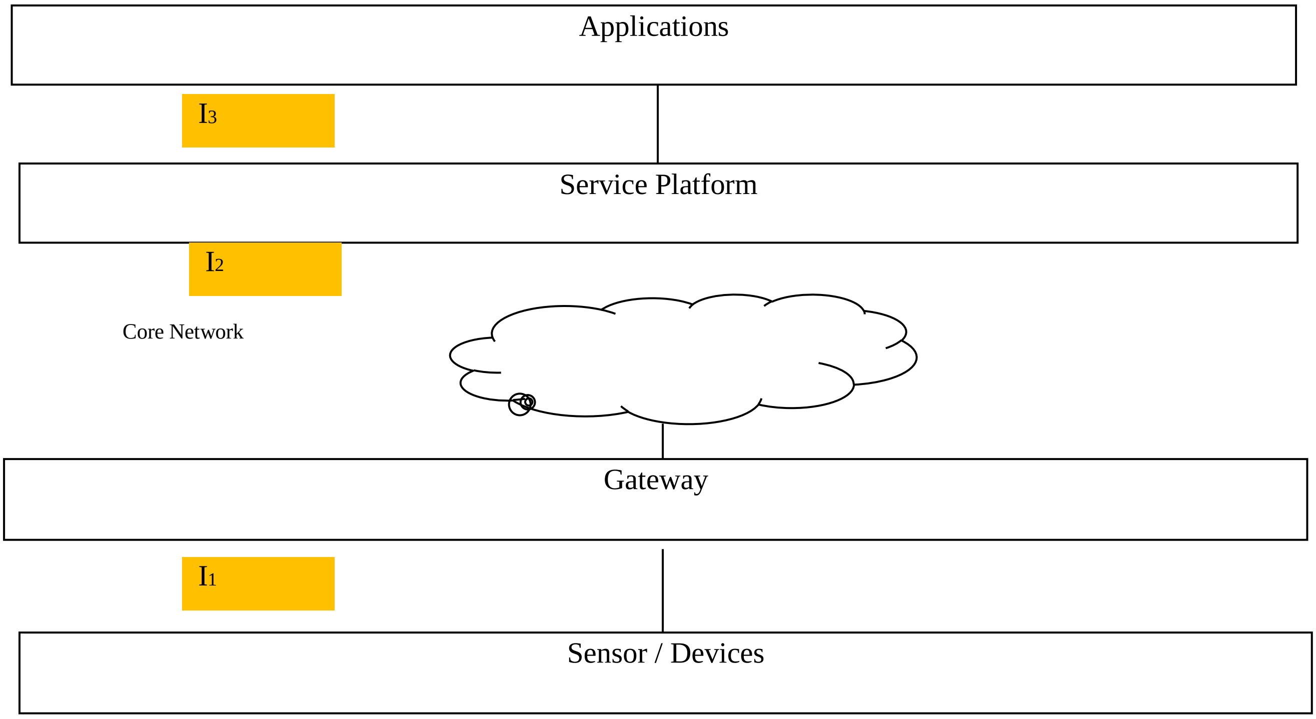


Figure 8: GISFI IoT Reference Architecture. (Source: Gisfi.org, 2019)

2.2. Industrial IoT architecture (IIoT)

In addition to these options there are other proposals that have emerged with the intention of establishing a RA for the Internet of Things, but from an industrial point of view (IIoT).

However, none of them reached a relevant acceptance. Among others, the following architectures for the IIoT have been considered.

2.2.1. Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0[13])

RAMI 4.0 defines a service-oriented architecture (SOA), in which each of the components / modules provides services to the other components via a communication protocol across a network.

The principles of SOA are independent of suppliers, products and technologies. The goal is to divide complex processes into packages that are easy to understand. This also includes data privacy and security (I2PANEMA Security and Privacy).

RAMI 4.0 promotes the principal aspects of the Industry 4.0:

[13] https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf

- **Interoperability**. Devices, machines and people need to communicate among each other.
- **Real-time data**. A smart factory must be able to store and process data in real-time.
- **Service oriented**. Production is oriented to the client. The products are created following the specification of the clients.
- **Modularity**. The factory acts as a module, adapting itself to the market trends, stationarity.

A major goal of RAMI 4.0 is to make sure that all participants involved in Industry 4.0 discussions and activities have a common framework to understand each other. Figure 9 depicts the RAMI 4.0 architecture:

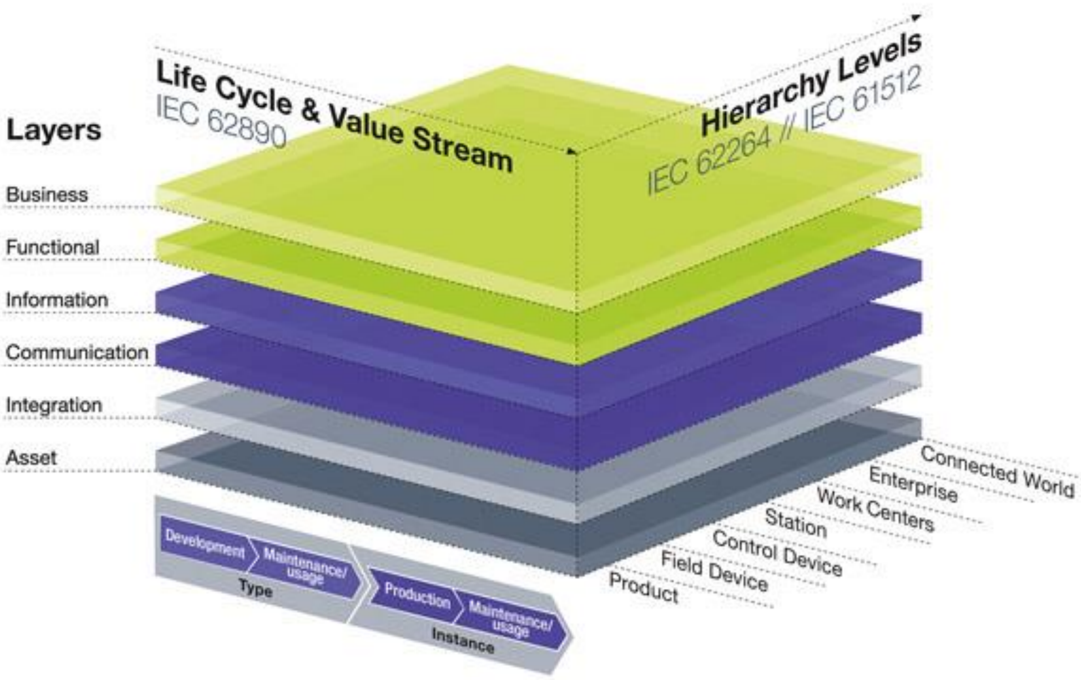


Figure 9: Reference Architectural Model Industry 4.0 (RAMI 4.0)

2.2.2. Industrial Internet Reference Architecture (IIRA[14])

IIRA is an architecture based on standards designed for IIoT systems. The value of this architecture is its fast applicability (the life cycle of the product is taken into account).

The IIC[15] Architecture Task Group under the Technology Working Group is responsible for the IIRA.

Figure 10: Industrial Internet Consortium (responsible for IIRA)

[14] https://www.iiconsortium.org/IIRA-1.7.htm

[15] www.iiconsortium.org

It has been built and designed with a high level of abstraction with the idea to support the requisite broad industry applicability. This architecture can be divided in four viewpoints as depicted in Figure 11:

- **Business**. Concerns to the identification of stakeholders and their business vision, values and objectives.
- **Usage**. Addresses the concerns of expected system usage.
- **Functional**. Focuses on the functional components, their interrelation and structure, the interfaces and interactions between them.
- **Implementation**. Deals with the technologies required to implement functional components, their communication schemes and their lifecycle procedures. These components are coordinated by activities (Usage viewpoint) and supportive of the system capabilities (Business viewpoint).

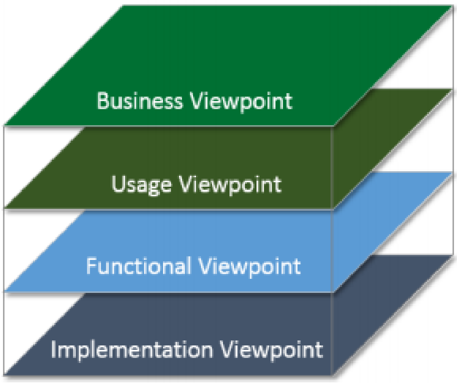


Figure 11: IIRA Architecture Viewpoints

2.2.3. RAMI 4.0 vs IIRA

Both architectures are complementary. IIRA analyses IIoT in all industries, with an emphasis on homogeneity and interoperability between industries, while RAMI 4.0 focuses on manufacturing and related value chain life cycles.

The great added value of both architectures consists in allowing interoperability between those IIoT systems that are built on the basis of these reference architectures. Figure 12 shows the mapping process between these two RA.

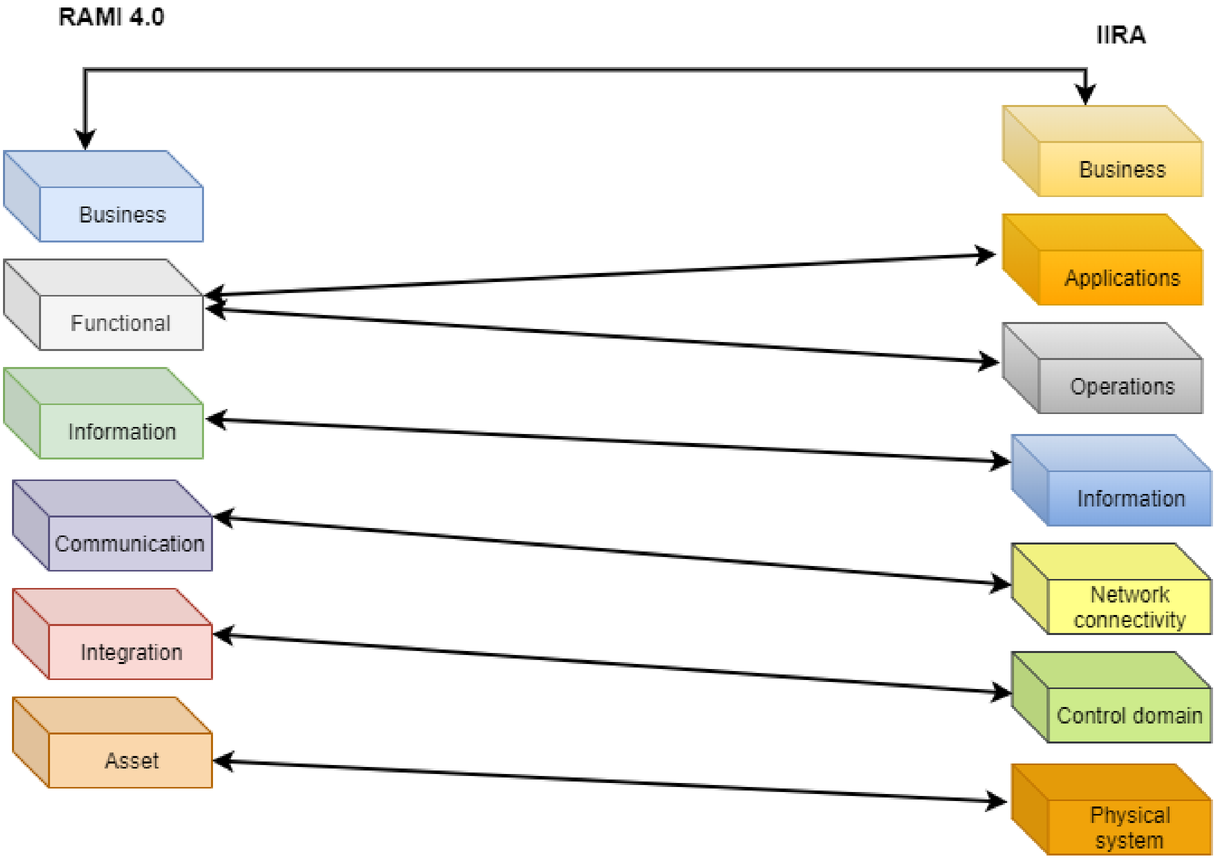


Figure 12: Equivalence from RAMI to IIRA architecture

2.3. International Data Spaces (IDS)

This section provides two views on the International Data Spaces (**IDS**): the general structure of the Reference Architecture Model of IDS and a brief view of the core concepts of the architecture of IDS connectors.

2.3.1. General structure of the Reference Architecture Model of IDS

This RA proposes a model that includes requirements for secure data exchange in business ecosystems. The general structure of the Reference Architecture Model is illustrated in Figure 13. The model is made up of five layers: business, functional, process, information, and system layer.

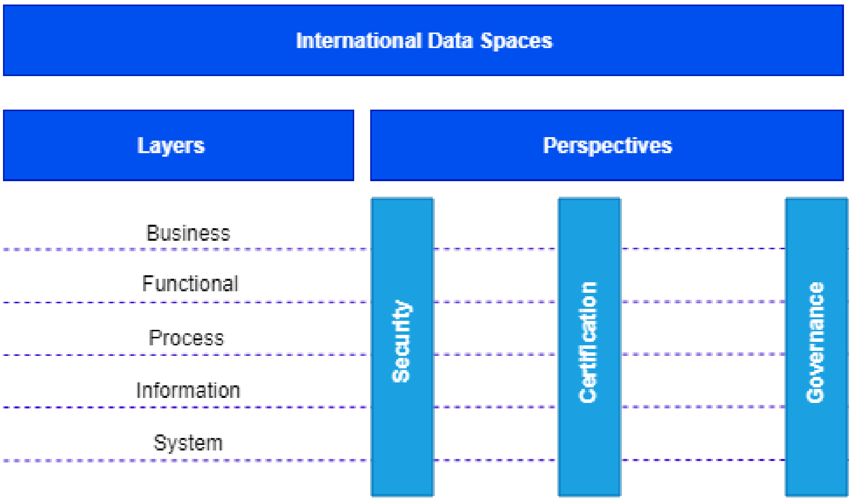


Figure 13: General structure of IDS Reference Architecture Model

1. **Business layer**. Specifies and categorizes the different roles which the participants of IDS can assume and specifies the main activities and interactions connected with each of these roles.
2. **Functional layer**. Defines the functional requirements of the IDS.
3. **Process layer**. Specifies the interaction among the different components of the IDS (using Business Process Model and Notation (BPMN)).
4. **Information layer**. Defines a conceptual model which makes use of linked-data principles for describing both the static and the dynamic aspects of the IDS's constituents.
5. **System layer**. Concerns the decomposition of the logical software components, considering aspects such as integration, configuration, deployment and extensibility of these components.

The information up to this point has been extracted from the Reference Architecture Model (International Data Spaces Association, v3.0, April 2019).

2.3.2. Core concept and notion of IDS connectors

The core notion of the International Data Spaces is the definition of reference architecture for "link[ing] different cloud platforms through secure exchange and trusted sharing of data" [12, p.14]. The aim is to enable participants to exchange sensitive data while retaining control over how their data is used and to promote the innovative use of data for products, services, and business models. In summary, the IDS offers participants the opportunity to exchange data for greater benefit while maintaining their data sovereignty [12, p.14].

This is realized through the use of connectors as shown in Figure 14. A connector is a software component that links a cloud environment or a device to a data space. The architecture of the IDS consists of a distributed network of connectors, data brokers and clearing houses without centralized data storage. Each connector can define rules that the other connectors must fulfil to use its data [12, p.13].

The connectors are the gatekeepers between data and internal infrastructure and the data space. The broker only exchanges (machine readable) metadata of the data source and data usage policies. The actual data flows directly between connectors, not through a broker [12, p. 56]. Further, the connectors load and execute data apps, which encapsulate data processing or transformation services [12, p. 57].

Thus, a participant in a data space can technically control to what extent and with what accuracy a connector uses the participant's data.

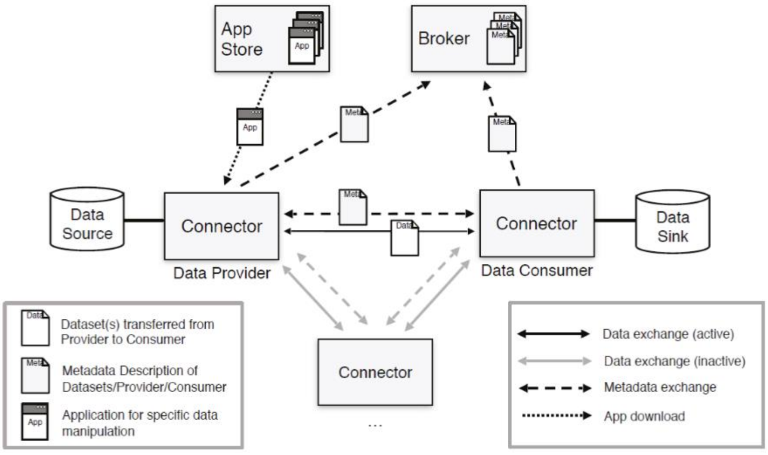


Figure 3.31: Interaction of technical components

Figure 14: Interaction of the technical components of IDS (Source: find figure in [12])

2.4. IoT Cloud Platforms

The growth of IoT causes an increase on the size of the computing resources and brings new issues to the systems like scalability, security etc. which can be solved technically with cloud computing capabilities. The clouds that exist today provide services for setting up servers, configuring networks, creating databases and for other infrastructure tasks to launch a basic IoT platform and they are being improved constantly as the technology develops.

The article "A survey of IoT cloud platforms"[16] compares 26 IoT cloud platforms regarding their appropriateness for the specific application domains but there exist more platforms[17] with different features. This section will focus on the big competitors in the market and try to give a general overview of the reference architecture of these cloud platforms.

2.4.1. AWS (Amazon) IoT

AWS is **Amazon's** cloud computing platform. AWS IoT[18] provides functionalities to build IoT solutions for use cases across a wide range of devices. It integrates with AI services, can scale as the device fleet grows and offers security features to respond immediately to potential security issues.

AWS presents different reference architecture diagrams for architectural guidance to build an application that uses AWS Cloud. Figure 15 depicts one of these RA diagrams, the "Machine Learning model with Modbus communication" reference architecture. It creates a Predictive Maintenance (PdM) Machine Learning (ML) model using AWS IoT SiteWise and AWS IoT Analytics with Amazon SNS anomaly detection notifications.

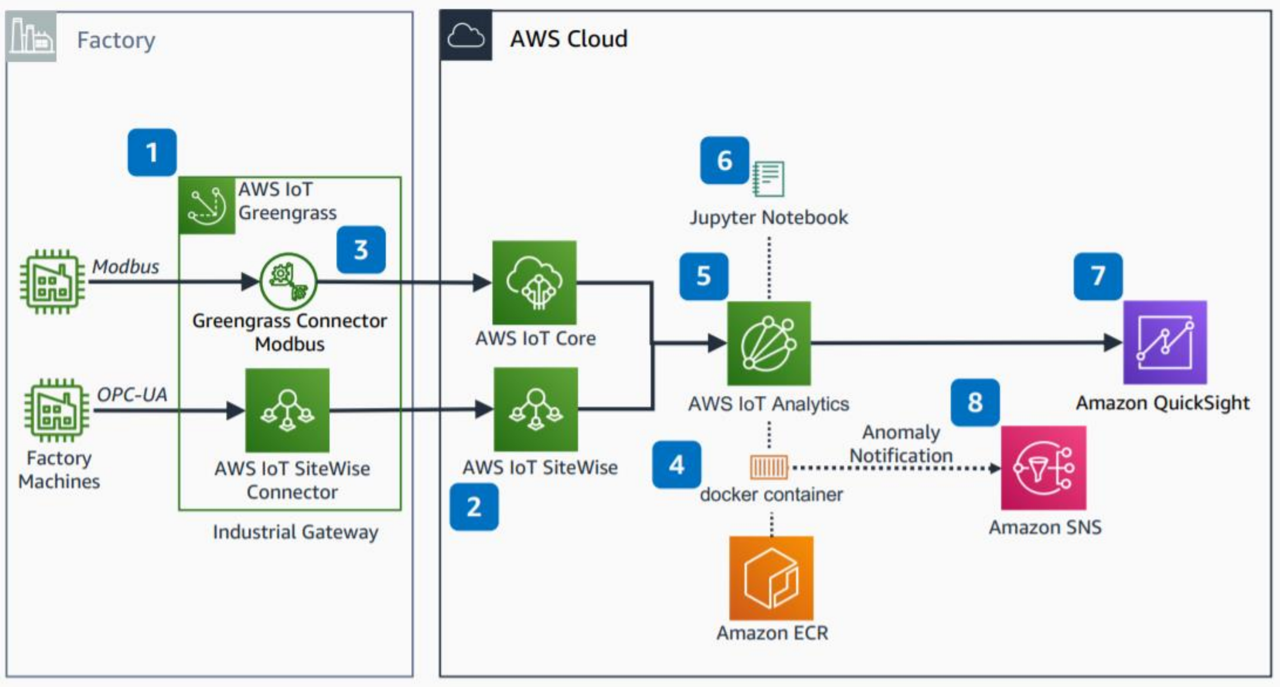


Figure 15: Machine Learning Model with Modbus Communication Reference Architecture. (Source AWS Industrial Predictive Maintenance)

[16] https://doi.org/10.1016/j.fcij.2017.02.001

[17] Some other IoT cloud platforms on the market:

- thethings.io (https://thethings.io)
- KAA (http://www.kaaproject.org/)
- Temboo (https://temboo.com)
- SensorCloud (http://www.sensorcloud.com)
- Ayla IoT Fabric (https://www.aylanetworks.com)
- Exosite (https://exosite.com)
- Arrayent (http://www.arrayent.com)
- OpenRemote (http://www.openremote.com)
- Arkessa (http://www.arkessa.com)
- Jasper Control Center (https://www.jasper.com)
- AerCloud (http://www.aeris.com)
- ThingSpeak (https://thingspeak.com)

[18] https://aws.amazon.com/iot/

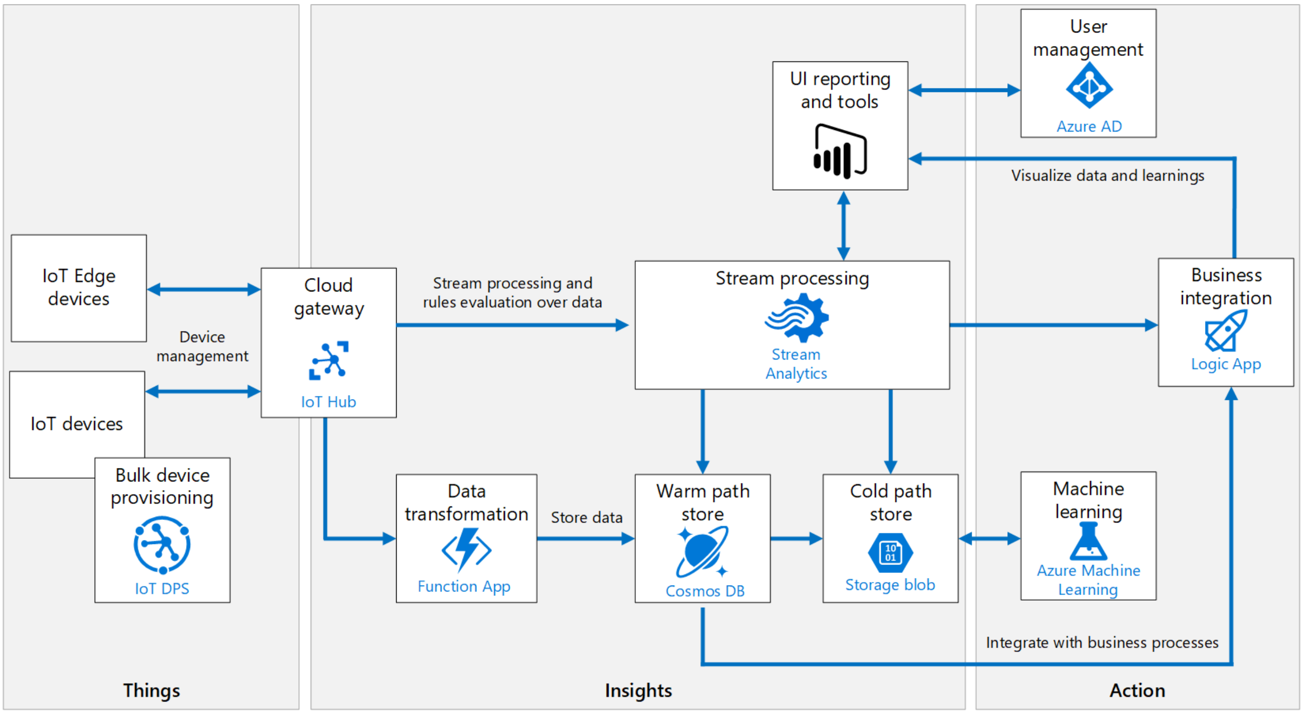
1. Deploy an AWS IoT SiteWise Gateway to connect to the factory machines OPC-UA Servers
2. Create a view in AWS IoT SiteWise and define the factory machines as assets. Define the Metrics to be monitored for the factory machines.
3. Configure a Modbus Greengrass Connector on AWS IoT Greengrass to send Modbus data to AWS IoT Analytics using a rule in AWS IoT Core
4. Build a Docker image and add it to Amazon Elastic Container Registry (Amazon ECR).
5. In AWS IoT Analytics, create a container data set from the AWS IoT SiteWise data store and link it to your Docker container
6. From AWS IoT Analytics, create a new Jupyter Notebook for the data set created from AWS IoT SiteWise to create a Predictive Maintenance (PdM) Machine Learning (ML) model.
7. Visualize your analysis using Amazon QuickSight on the AWS IoT Analytics data source.
8. Create a topic for anomaly detection notifications in Amazon Simple Notification Service (Amazon SNS) and configure the trigger in your model.

2.4.2. Microsoft Azure IOT

Azure IoT is **Microsoft's** IoT cloud platform. Azure IoT[19] is designed for different industry needs so it can be used from manufacturing to transportation or energy industries. It recommends a cloud native, micro service based, and server-less architecture for IoT solutions. It suggests building subsystems as discrete services that are independently deployable, and able to scale independently. Other suggestions regarding the architecture:

- Monitoring individual subsystems as well as the IoT application as a whole,
- Communicating over REST/HTTPS using JSON,
- Hybrid cloud and edge compute strategy, using an orchestrator like Azure Kubernetes Services to scale individual subsystems horizontally,
- PaaS services like Azure App Services that offer built-in horizontal scale capabilities. (Microsoft Azure IoT Reference Architecture, 2018)

Figure 16 shows the RA based on these suggestions by Azure IoT.



[19] https://azure.microsoft.com/en-in/overview/iot/

Figure 16: Recommended architecture for IOT applications on Azure using PaaS components (Source, https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot/)

2.4.3. IBM Watson IoT Platform

IBM Watson IoT Platform[20] offers a set of preselected services to form a public SaaS solution on the **IBM** Cloud framework. It supports functions of connecting, storing, analysing, managing and monitoring support through the Platform dashboard. The cloud framework handles back end operations like security, resource allocation, backup and high availability. (IBM Knowledge Center, 2019)

As depicted on Figure 17, IBM Watson IoT platform uses the following data flow:

1. IoT devices send data to the Platform Service, which acts as a message broker and real-time handler of IoT data.
2. Device data is written to Cloudant® NoSQL DB for short term historian access and to Db2 Warehouse on Cloud for intermediate length storage and analytics. IBM Event Streams are for real-time retrieval of current data.
3. IBM Cloud Object Storage is used for long term storage.
4. The Watson IoT Platform Analytics add-on uses the Db2 Warehouse as the data lake, and it imports and ingests data from external sources. (IBM Knowledge Center, 2019)

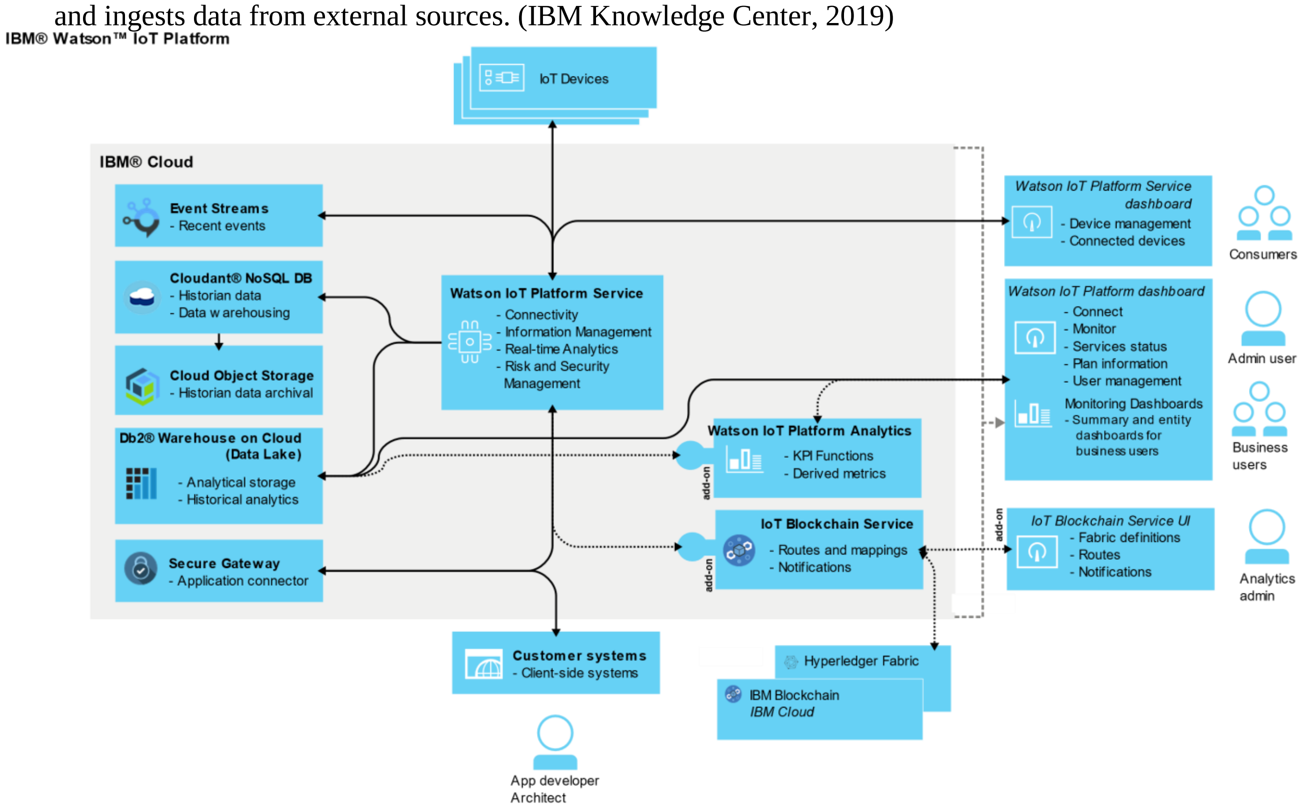


Figure 17: Product Architecture for IBM Watson IOT Platform

(Source: IBM Knowledge Center, 2019)

2.4.4. Google Cloud IoT

Google Cloud IoT is **Google's** IoT cloud platform. Google Cloud IoT[21] provides solutions both at the edge and in the cloud for IoT devices. Its tools to connect, process, store, and analyse data create a scalable, fully managed cloud service.

As it is depicted in Figure 18Figure 18, the reference architecture for Google Cloud IOT Platform consists of:

[20] https://www.ibm.com/internet-of-things/solutions/iot-platform/watson-iot-platform

[21] https://cloud.google.com/solutions/iot/

1. Cloud Dataflow for streaming and batch analytics.
2. Cloud Pub/Sub module work for ingesting connection and management.
3. Cloud IOT Core responsible for device connection and management.
4. BigQuery module designed for data warehouse and fast querying
5. Cloud ML Engine for training, deploying and running ML models

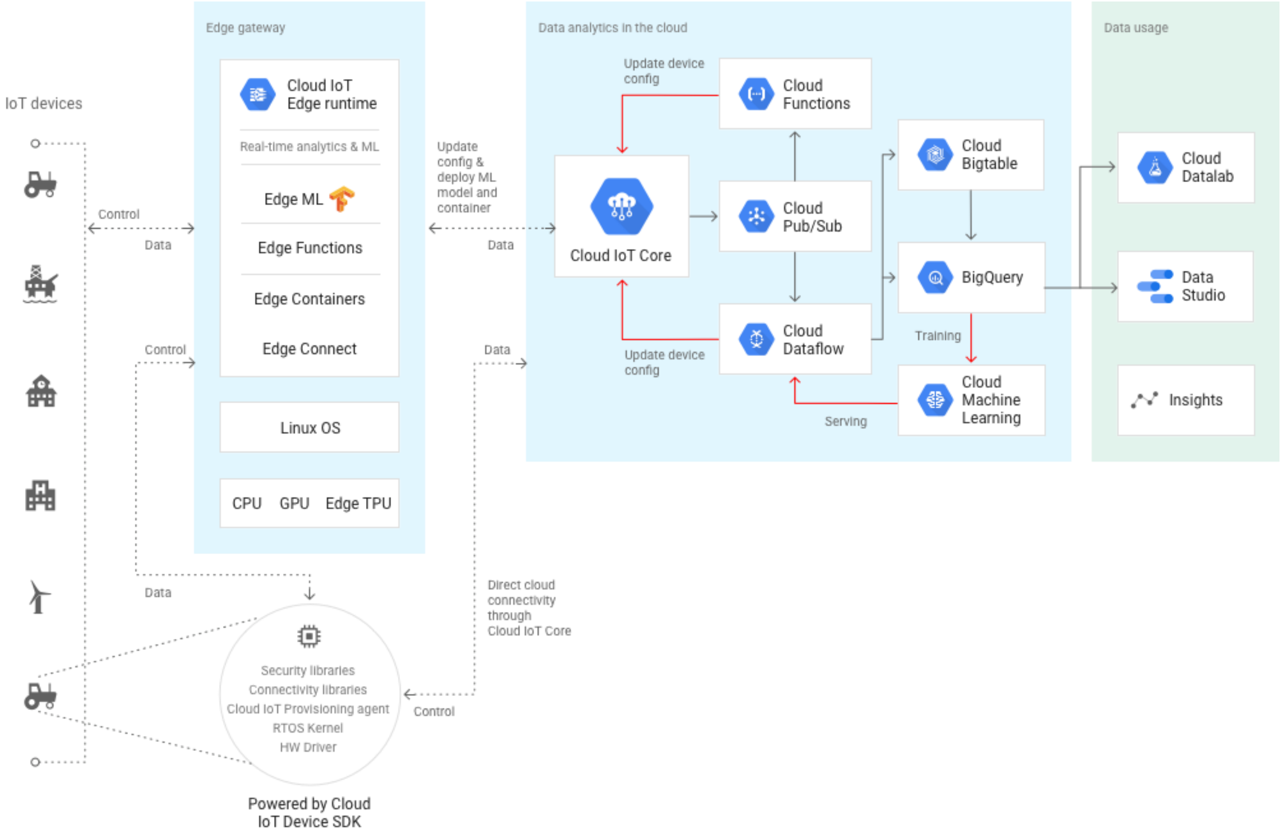


Figure 18: Reference Architecture for Google Cloud IOT Platform
(Source, https://cloud.google.com/iot-core/)

2.4.5. Cisco IoT Cloud Connect

Cisco IoT[22] provides the Cisco Kinetic IoT platform to extract, compute and move data. It provides a secure IoT solution architecture enhanced with IoT security services and tools like Cisco DNA Centre and Field Network Director to integrate IT and OT infrastructure. Control IoT edge applications with IOx. Cisco has solutions related to IoT in Energy, Manufacturing, Transportation, Cities and communities, Retail, Education etc.

The proposed IoT reference model which can be seen in Figure 19 is comprised of the following seven levels:

1. **Physical Devices and Controllers**. A wide range of heterogeneous devices that send and receive information.
2. **Connectivity**. Reliable, timely information transmission between devices (Level 1) and the networks
3. **Edge (Fog) Computing**. High-volume data analysis and transformation.
4. **Data Accumulation**. Data in motion is converted to data at rest.
5. **Data Abstraction**. Rendering data and its storage in ways that enable developing simpler, performance-enhanced applications.

[22] https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

6. **Application**. The layer where information is interpreted. Software at this level interacts with Level 5 and data at rest.
7. **Collaboration and Processes**. This layer focuses on people and business processes. (CISCO, The Internet of Things Reference Model, 2019)

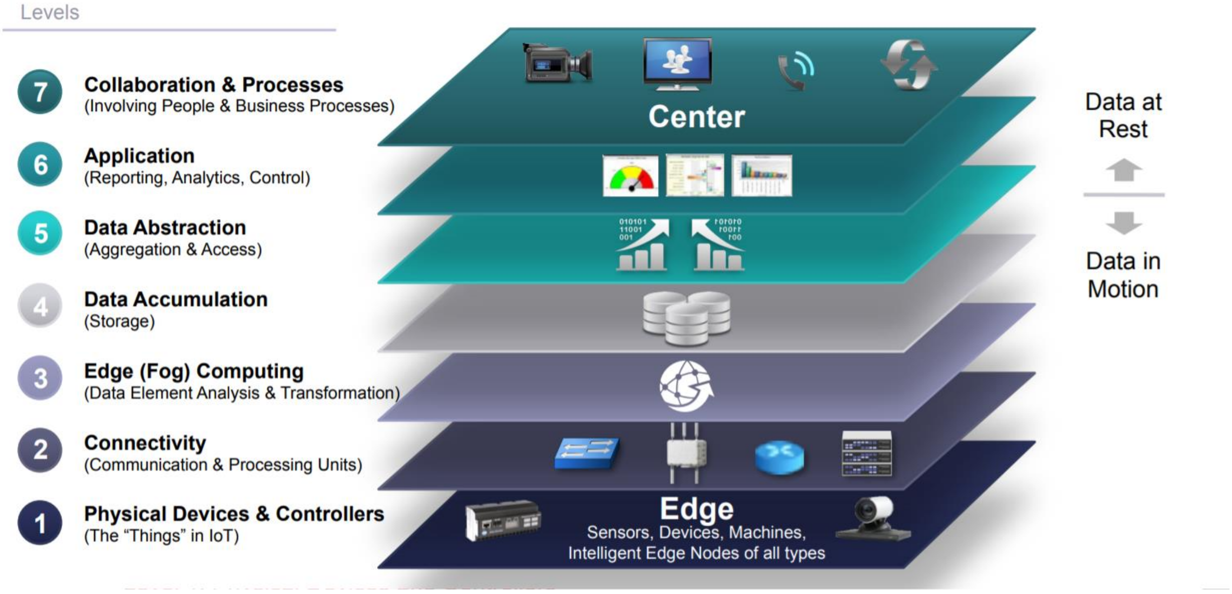

Levels
7 Collaboration & Processes
(Involving People & Business Processes)
6 Application
(Reporting, Analytics, Control)
5 Data Abstraction
(Aggregation & Access)
4 Data Accumulation
(Storage)
3 Edge (Fog) Computing
(Data Element Analysis & Transformation)
2 Connectivity
(Communication & Processing Units)
1 Physical Devices & Controllers
(The "Things" in IoT)
Center
Edge
Sensors, Devices, Machines,
Intelligent Edge Nodes of all types
Data at Rest
Data in Motion

Figure 19: Reference Architecture for Cisco Cloud IOT Platform (Source: CISCO, The Internet of Things Reference Model, 2019)

2.4.6. Oracle IoT Platform

Oracle's IoT platform[23] gives enterprises the tools and technologies to integrate, analyse, build, and deploy IoT solutions that deliver analytical insights from real-time IoT data into existing applications as shown in Figure 20:

- Devices are connected through the Device Layer to the system.
- The data obtained from devices is carried by a Communication Service Provider Network.
- The sensors/gateways and virtual enterprise infrastructure are connected to each other via the IoT Infrastructure Layer.
- The IoT Services Layer, which includes a set of services to build IoT applications, is completely independent from devices, protocols and semantics.
- Developers can build applications via the IoT Developer Services Layer. (Internet of Things: Role of Oracle Fusion Middleware, 2019)

[23] https://www.oracle.com/tr/internet-of-things/

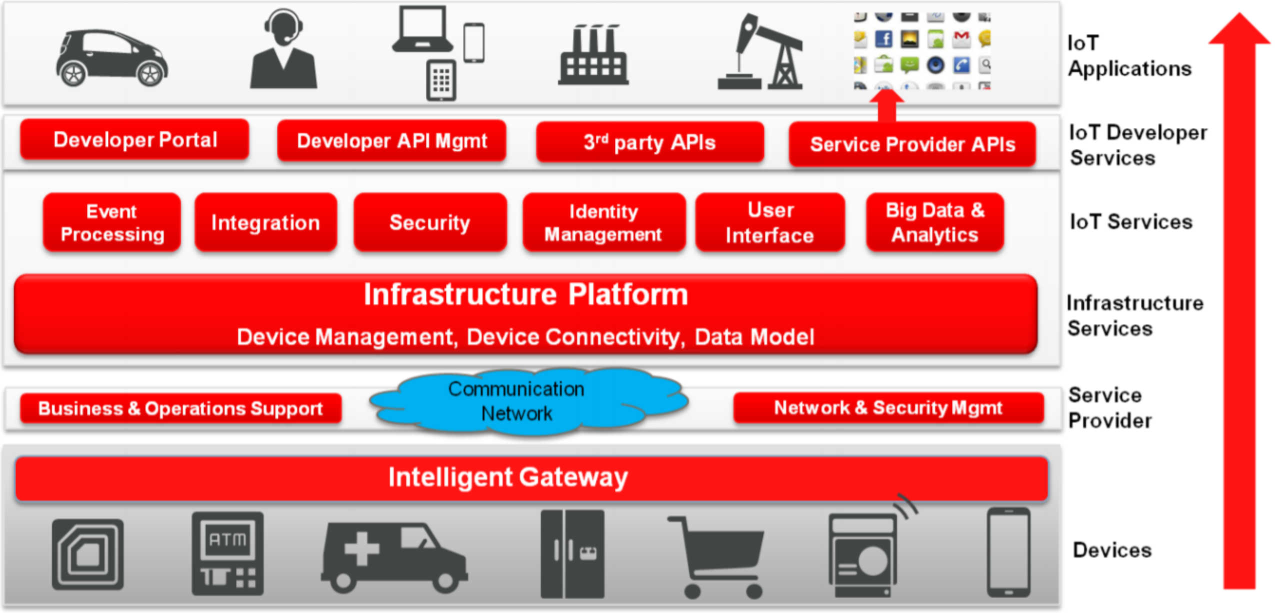


Figure 20: Reference Architecture for Oracle Cloud IoT Platform
(Source: Internet of Things: Role of Oracle Fusion Middleware, 2019)

2.5. Conclusion

As we have seen, there is no common RA. There are several attempts to create a standard and major differences among them exist.

However, among the features of the future I2PANEMA's RA will be some of the features seen so far:

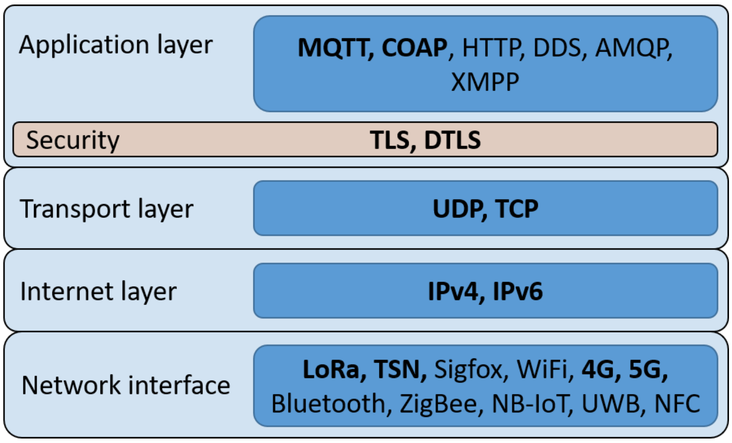
- **Industrial focus**. The applications based on I2PANEMA will be developed in industrial environments, such as ports, and thus the specific requirements for industry will match better than generic architectures.
- Focus on **interoperability**.
- **Secure data exchange** in our business ecosystem (ports).

3. IoT Communication

This layer will be responsible for data acquisition. Its purpose is to provide a standard way to import data from the existing data sources. The solution provides a standard way to acquire data from different data sources that implement various protocols and data types. The idea is to provide a standard way to import data into I2PANEMA Data Management in order to allow an easy integration of any kind of data sources available on each port.

3.1. IoT Protocol Stack

The world of IoT communication offers a wide range of protocols that have established themselves in the past. The IoT stack, depicted in Figure 21, gives a brief overview of the technologies which are relevant for I2PANEMA port communication. On the application layer two of the most widely used protocols are the Message Queuing Telemetry Transport (MQTT) and the Constrained Application Protocol (CoAP). For more resource constrained devices there is an extension called MQTT for Sensor Networks (MQTT-SN). Further application layer protocols are HTTP, DDS, AMQP and XMPP. All these protocols rely on either TCP or UDP using IPv4 and IPv6. Usually security is built on top of the transport layer. Depending on the usage of TCP and UDP, Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS) is applied. In general, IoT communication is based on wireless networking technologies, such as Wi-Fi, 4G/5G, LPWANs like LoRaWAN and NB-IoT or Bluetooth. In some cases, also wired technologies like Ethernet are used. With respect to real time requirements Time Sensitive Networking (TSN) can be an option.



Application layer	**MQTT, COAP**, HTTP, DDS, AMQP, XMPP
Security	**TLS, DTLS**
Transport layer	**UDP, TCP**
Internet layer	**IPv4, IPv6**
Network interface	**LoRa, TSN,** Sigfox, WiFi, **4G, 5G,** Bluetooth, ZigBee, NB-IoT, UWB, NFC

Figure 21: IoT Protocol Stack

3.2. Application Layer Protocols

Application layer protocols define the way data is exchanged on top of TCP or UDP transport. This includes, e.g., data format, message types and reliability. In IoT, application layer protocols need to be designed to meet certain constraints, for example with respect to bandwidth, energy consumption and computing power. Another decisive factor is the communication pattern. The classic World Wide Web communication is based on pure client/server. Here, communication takes place in a request/response manner with only one-to-one connectivity. This approach, although applicable to many IoT use cases, has a few disadvantages. Firstly, clients have to query data periodically (polling) which increases the resource consumption (energy, computing power and bandwidth). Secondly, it prohibits one-to-many or many-to-many communication, which is substantial for IoT communication, where thousands of sensors send data to multiple servers at the same time. Additionally, for this approach communication is inherently synchronous and doesn't allow decoupling of devices. One approach which has prevailed alongside client/server is publish/subscribe which addresses these problems. Here, data is published to a single device or a group of devices in an event-based manner. Regardless of how publish/subscribe is implemented, e.g. with or without broker, this eliminates the problems of polling and limitation to one-to-one communication. Besides, in most of the implementations communication is managed by a central instance, which decouples the devices. However, it also leads to this instance being a single point of failure which is one major disadvantage.

3.2.1. Message Queuing Telemetry Transport (MQTT)

MQTT is one of the most widely used application layer protocols in IoT. It is a lightweight M2M protocol, which is based on a broker-based publish/subscribe approach (see Figure 22). A publisher can publish data on a topic or a group of topics. For a subscriber, who can subscribe to topics, these serve as a kind of content filter. Topics are strings consisting of one or multiple topic levels separated by slashes. MQTT supports single-level (+) and multi-level (#) wildcards to publish on multiple topics at the same time by replacing either one, or multiple topic levels. A subscriber gets notified each time data is published on the topics he subscribed to. In this approach, the broker is responsible for distribution of the messages, subscription management and storing the messages. Since there is no direct connection between publisher and subscriber, communication is decoupled in space, time and synchronization. That means, both do not need to know each other (address, port, etc.), they do not have to run at the same time and they can communicate in an asynchronous manner.

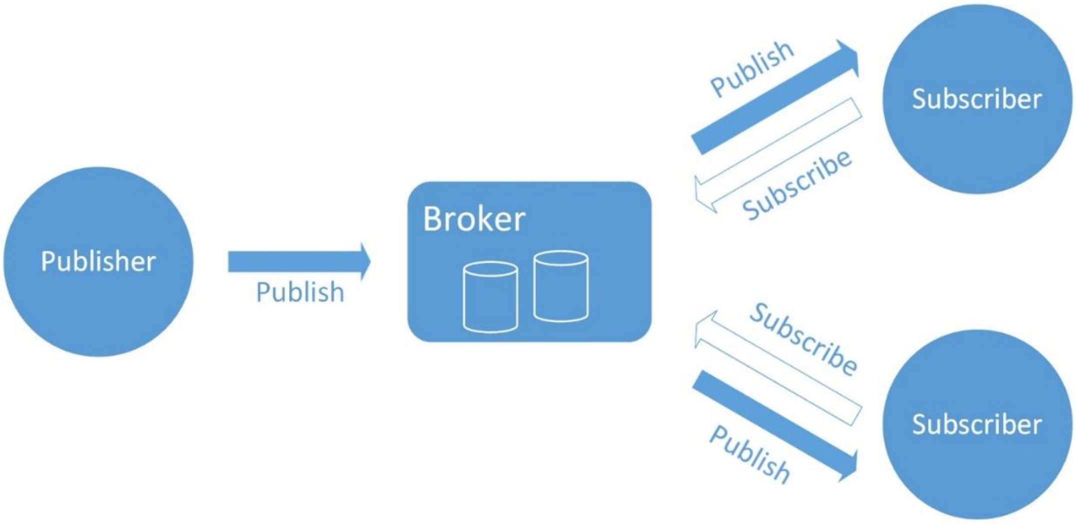


Figure 22: MQTT – Broker-based Publish/Subscribe

MQTT is based on TCP and it supports TLS for security. This includes authentication and encryption mechanisms. MQTT provides three qualities of service (QoS) levels. The first level (0) doesn't give any guarantees, that means the message is delivered at most once (fire and forget). The second level (1) ensures that the message is delivered at least once. That means the sender waits for message delivery to be acknowledged. Here, the message can potentially be transmitted more than once. For the highest QoS level (2), the message is delivered exactly once by using a four-step handshake. The higher the QoS level, the higher the reliability will be. However, the communication overhead increases as well. MQTT also supports persistent sessions, message queuing and retained messages[24]. Last will and Testament (LWT) messages can be used by clients to leave a message at the broker to still respond after disconnect. LWT messages are sent during the connection process with the broker. MQTT supports WebSocket communication, which enables the use of MQTT in a web browser.

MQTT for Sensor Networks (MQTT-SN) is an extension of MQTT developed for the use in resource constrained environments. IoT devices are usually limited in bandwidth, energy consumption, CPU and memory. Thus, MQTT-SN is optimized to meet these constraints. The main differences, compared to MQTT, are[25]:

- Usage of UDP instead of TCP
- Smaller message sizes
- Smaller topic id, pre-defined topics and shorter topic names
- Gateway discovery
- Client sleep mode
- QoS level -1

As depicted in Figure 23, there are two different ways to integrate MQTT-SN into MQTT. Both ways require an MQTT-SN to MQTT gateway. The first way uses an external gateway to translate MQTT-SN messages into MQTT compliant messages. The other way uses a gateway which is directly integrated into the broker. Further, two types of gateways are distinguished. A transparent gateway uses one MQTT connection for each MQTT-SN client. An aggregating gateway instead uses one single connection between gateway and the broker. If a client is not located inside the gateways network, an MQTT-SN forwarder can be used which just forwards the MQTT-SN packets without changing them.

[24] https://www.hivemq.com/mqtt-essentials/
[25] www.mqtt.org/

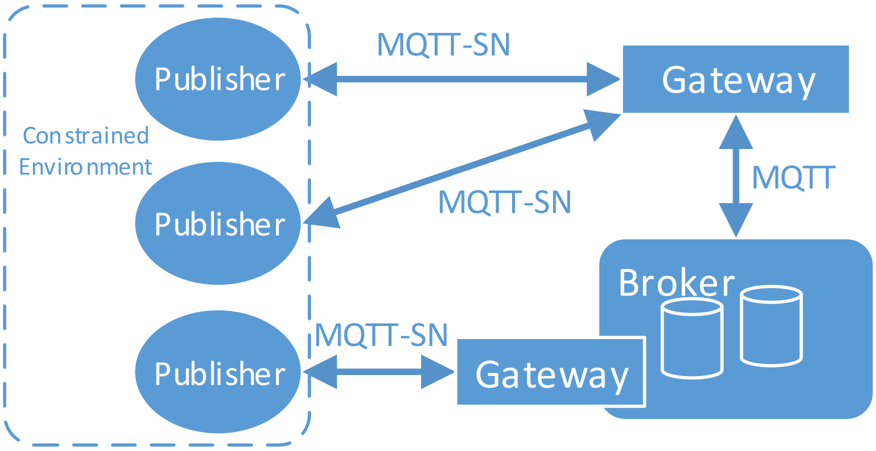


Figure 23: Integration of MQTT-SN using Gateways

3.2.2. Constrained Application Protocol (CoAP)

CoAP is a lightweight RESTful M2M protocol developed specifically for the use in resource constrained environments and specified by the Internet Engineering Task Force (IETF) in RFC 7252. As a RESTful protocol, it is characterized by client/server communication, statelessness and the CRUD (Create, Read, Update and Delete) operations GET, PUT, POST, DELETE to access resources by URL/URI. Thus, it is similar to HTTP and also supports HTTP access via proxies as shown in Figure 24. However, it is neither a replacement, nor lightweight version of it.

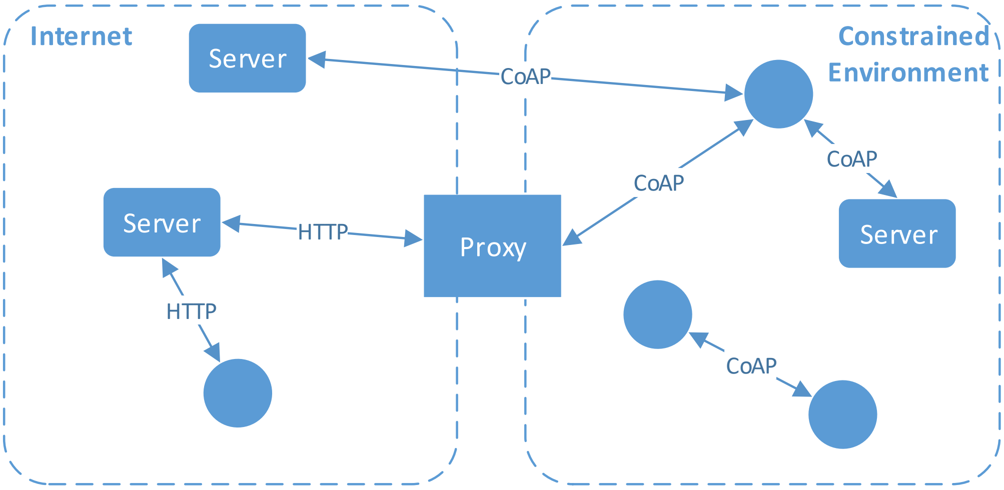


Figure 24: The Constrained Application Protocol (CoAP) Architecture

The CoAP core standard (RFC 7252[26]) is based on UDP. Security can be applied using DTLS or Object Security for Constrained RESTful Environments (OSCORE) which is a substandard (RFC 8613[27]) of CoAP. Additionally, there is an extension for CoAP over TCP, TLS and WebSockets (RFC 8323[28]). The communication is based on client/server. Servers provide resources via URI which can be accessed by clients using the CRUD operations in request/response manner. However, there also exists a draft for broker-based publish/subscribe. For reliability, CoAP defines confirmable and non-confirmable messages, as well as retransmission, deduplication and message order. Table 1 shows an overview of the RFC documents for CoAP which describe additional features.

[26] https://tools.ietf.org/html/rfc7252
[27] https://tools.ietf.org/html/rfc8613
[28] https://tools.ietf.org/html/rfc8323

Table 1: Overview of Basic CoAP RFC Documents

RFC No.	Title
RFC 7252	The constrained Application Protocol (CoAP)
RFC 6690	Constrained RESTful Environments (CoRE) Link Format
RFC 7641	Observing Resources in the Constrained Application Protocol (CoAP)
RFC 7390	Group Communication for the Constrained Application Protocol (CoAP)
RFC 7959	CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets
RFC 8613	Object Security for Constrained RESTful Environments (OSCORE)
Draft	Sensor Measurement Lists (SenML)
Draft	CoRE Resource Directory
Draft	Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)

3.2.3. Data Distribution Service (DDS)

DDS is a standard for M2M communication, being standardized by the Object Management Group (OMG). In contrast to MQTT, DDS implements a broker-less publish/subscribe using UDP multicast messages for data distribution instead of a centralized broker. As depicted in Figure 25 data writers can publish data on specific topics which can be received by one or multiple data readers (subscribers). For message delivery, a set of Quality of Service (QoS) policies is specified. These include reliability, liveliness and security features. For security there exists a standard extension called DDS security. Security can also be applied through DTLS on top of UDP. Content filters can be used to filter data based on topics using filter expressions, e.g., "is equal to", "is less than" or "is greater than". Filter expressions can also be linked with "AND" or "OR". In contrast to MQTT, for example, DDS also specifies its own type system.

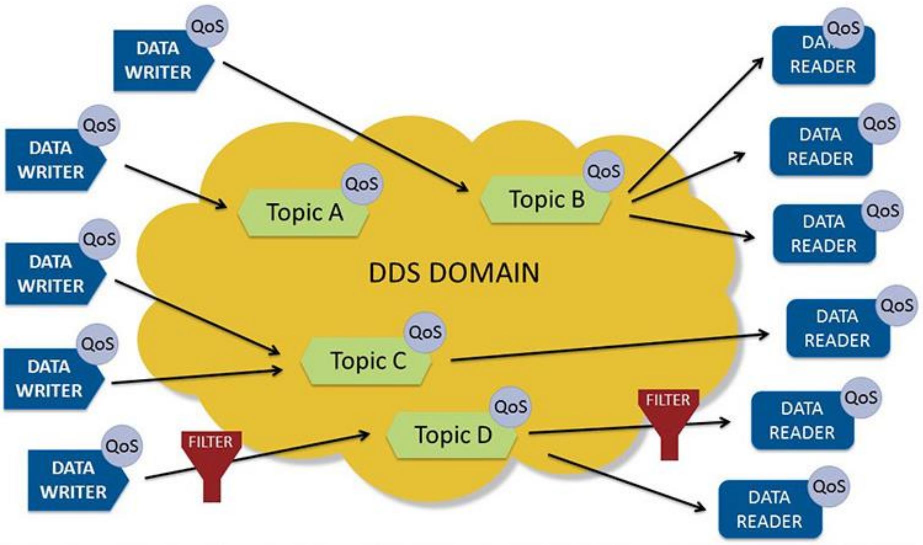


Figure 25: The Data Distribution Service (DDS) Architecture[29]

3.2.4. Advanced Message Queuing Protocol (AMQP)

The Advanced Message Queuing Protocol (AMQP) is a binary protocol for message-oriented communication by means of broker-based publish/subscribe.

[29] https://www.dds-foundation.org/

As shown in Figure 26, the AMQP publish/subscribe model is very similar to MQTT. The main difference is the internal structure of the server (broker). An AMQP server consists of an exchange component and a message queue. The exchange component is responsible for forwarding the messages published by the clients based on different exchange types. The message queue stores these messages and distributes them to the subscribers. The binding specifies how the messages are matched to the corresponding message queue.

There are four different exchange types: direct, fanout, topic and headers exchange.

Direct exchange allows messages to be sent to exactly one subscriber. This is achieved by matching a routing key defined in the message and a unique binding key of a message queue.

Fanout exchange distributes the messages to all available message queues without checking the routing keys.

Topic exchange extends the concept of direct exchange by wild cards. Thus, the routing key does not need to match the binding completely, but only partially. This allows messages to be distributed to multiple message queues.

Headers exchange realizes routing from the exchange component to the message queues by evaluating the header of a message instead of a routing key.

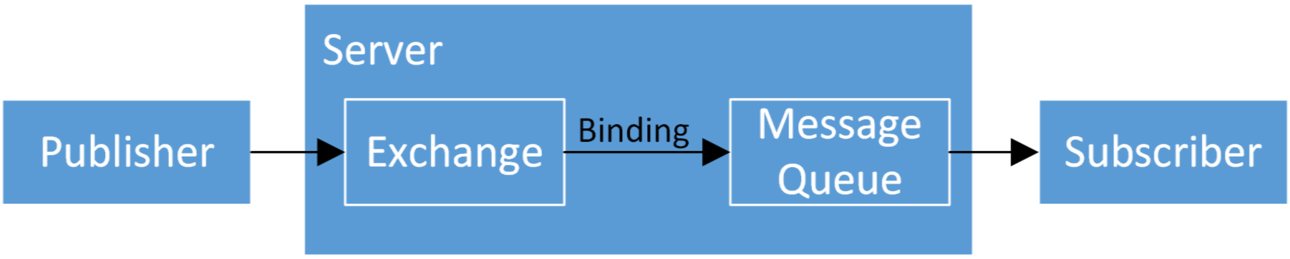


Figure 26: The AMQP Publish/Subscribe Model[30]

3.2.5. Extensible Messaging and Presence Protocol (XMPP)

The Extensible Messaging and Presence Protocol (XMPP) is an application layer protocol which is specified as an open RFC standard by the IETF. It enables the exchange of structured data based on XML and was originally designed for instant messaging (IM). The XMPP Core standard (RFC 6120) specifies the following functionalities[31]:

- Setup and teardown of XML streams
- Channel encryption
- Authentication
- Error handling
- Communication primitives for network availability and request-response interactions

XMPP uses TCP for transport and supports TLS for security. The communication is based on the client/server principle. As depicted in Figure 27 two clients on different servers can communicate with each other using their related server as intermediate. A client can establish a connection to its server which can either forward messages to other clients being connected to the same server or route messages to a remote server. Thus, end-to-end communication between clients is logically peer-to-peer-based. Additionally, there is also an extension for XMPP using the publish/subscribe pattern which is described in the XMPP specification XEP-0060[32].

[30] http://www.amqp.org/
[31] https://tools.ietf.org/html/rfc6120
[32] https://xmpp.org/extensions/xep-0060.html

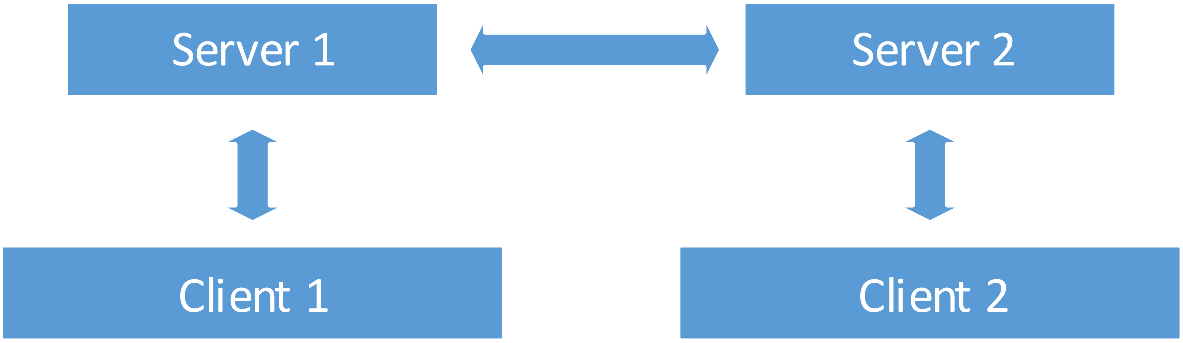


Figure 27: XMPP Client/Server Architecture

3.2.6. WebSocket

The WebSocket Protocol, being standardized by the IETF (RFC 6455[33]), was originally developed for bidirectional communication between web applications (client) and web servers (server), to avoid periodically polling and multiple connections on the client-side. Instead, a client opens a single and permanent connection to the server using TCP. Communication takes place in two steps. In the first step, the client establishes the connection to the server using a handshake. In the second step, if the handshake process was successful, both can communicate in a bidirectional manner. As communication is based on TCP, security can be applied by means of TLS.

3.2.7. Comparison of IoT Application Layer Protocols

Table 2 gives a brief summary and comparison of the aforementioned application layer protocols.

Table 2: Comparison of Application Layer Protocols[34]

Protocol	Standard	RESTful support	Transport	Security	QoS	Communication Pattern
CoAP	IETF	Yes	UDP (TCP)	DTLS (TLS)	Yes	Client/Server Publish/Subscribe
MQTT	OASIS	No	TCP	TLS	Yes	Publish/Subscribe
MQTT-SN	IBM	No	TCP	TLS	Yes	Publish/Subscribe
XMPP	IETF	No	TCP	TLS	No	Client/Server Publish/Subscribe
AMQP	ISO/IEC	No	TCP	TLS	Yes	Publish/Subscribe
DDS	OMG	No	UDP	DTLS	Yes	Publish/Subscribe
HTTP	IETF/W3C	Yes	TCP	TLS	No	Client/Server
WebSocket	IETF	Yes	TCP	TLS	No	Client/Server

3.3. Network Interfaces

In general, the choice of suitable networking technologies depends on the use-case, and thus on different parameters, e.g. bandwidth, latency, data-rate, robustness, scalability or real-time capability. As already mentioned at the beginning, in IoT the communication often relies on wireless technologies. Here, also the

[33] https://tools.ietf.org/html/rfc6455

[34] A. Čolaković, M. Hadzialic, "Internet of Things (IoT): A Review of Enabling Technologies, Challenges, and Open Research Issues", Computer Networks, 144, 2018

range is an important factor. As illustrated in Figure 28, wireless communication technologies can be classified into the following groups (with rising range): Proximity, Wireless Personal Area Network (WPAN), Wireless Local Area Network (WLAN), Wireless Metropolitan Area Network (WMAN) and Wireless Wide Area Network (WWAN). In the beginning of IoT, short-range (Proximity and WPAN) communication technologies, like NFC, Bluetooth, and ZigBee, were of particular importance. However, a shift towards long-range communication technologies, especially Low Power Wide Area Network (LPWAN), has been observed. A detailed description and characterization of the relevant technologies for I2PANEMA IoT communication w.r.t. the aforementioned parameters are provided in the following sections.

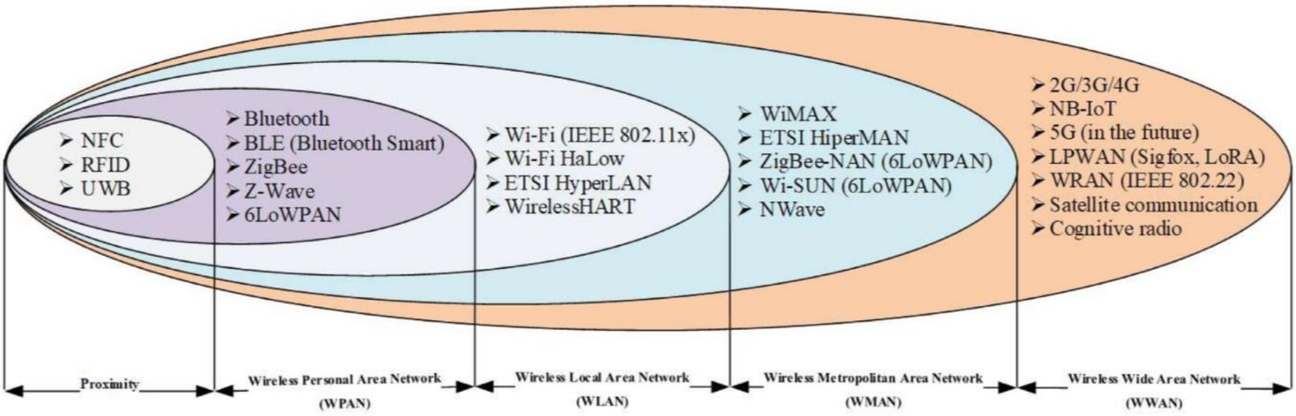


Figure 28: Wireless Communication Technologies [34]

3.3.1. LoRaWAN

LoRa (Long Range) is a proprietary physical layer protocol developed by the US semiconductor manufacturer Semtech[35]. It uses a Chirp Spread Spectrum (CSS) modulation for long range (up to more than 10 km) and low power (up to 10 years of battery life) radio communication. In contrast to some similar technologies, like NB-IoT, it uses a license-free frequency band (868 MHz in Europe).

LoRaWAN is an LPWAN protocol, which defines a MAC layer on top of the LoRa physical layer protocol (LoRa PHY) as depicted in Figure 29. LoRaWAN devices can be classified by three classes which differ in the way downlink traffic is scheduled. Class A devices can communicate bi-directionally using two short downlink windows after each uplink transmission. This class is mandatory, and thus, each device has to support Class A communication. Class B devices can additionally open a receive slot at a certain time. Class B devices use beacon messages to synchronize with the network server which leads to a higher energy consumption compared to Class A devices. Class C devices can listen for downlink traffic continuously. That means Class C devices have the highest energy consumption.

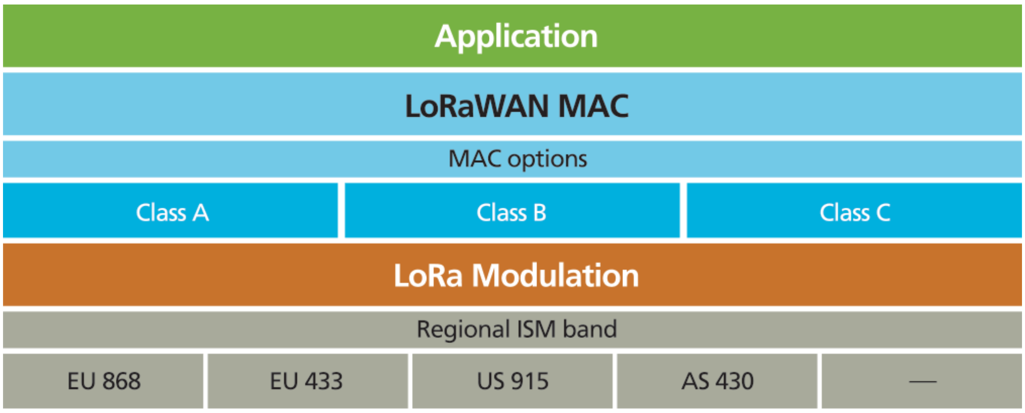


Figure 29: LoRaWAN Stack[36]

[35] https://www.semtech.com/lora
[36] https://www.semtech.com/uploads/images/what-LoRa-table-illustration-web.gif

LoRaWAN defines a star-of-stars topology where all end nodes are connected to one or multiple gateways via a single hop (see Figure 30). The gateway itself is connected to a network server via an IP-based network. Applications can be provided by one or multiple application servers. LoRaWAN supports end-to-end security between end nodes and application servers using device- and application-specific keys. An application server can be used to host its own applications or to integrate external services using MQTT or HTTP(S), for example. LoRaWAN has the following characteristics (see Table 3):

- Long range communication
- Low power consumption
- End-to-end security
- Low costs
- High scalability
- High interference immunity
- Low bandwidth
- Low data rates

One major drawback of LoRa/LoRaWAN is its limitation w.r.t. duty-cycle (0.1% / 1% of air-time) and bandwidth (250 kHz), which makes it not suitable for use-cases with high data traffic, whereas it is well suited for IoT applications where bandwidth, real-time and large payloads play a minor role. Examples are smart metering, street lightening or smart agriculture. As it is widely used in smart city scenarios, hence it can also be suitable for deployment in smart ports.

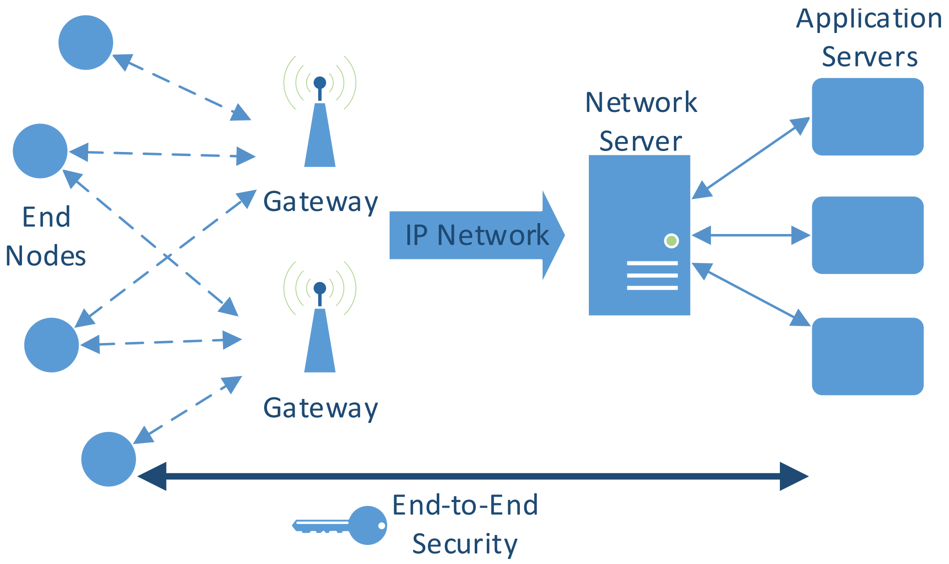


Figure 30: LoRaWAN Network Architecture[37]

3.3.2. Other LPWANs – NB-IoT and Sigfox

Besides LoRaWAN, there are some other LPWAN technologies, all having some advantages and disadvantages in comparison to LoRaWAN (for a detailed overview and comparison see Table 3). The best known are NB-IoT and Sigfox.

NB-IoT is an LPWAN which uses a licensed frequency band (700 – 900 MHz) and is developed and standardized by the 3GPP. One of the main advantages over LoRa is that it supports higher data rates. On the other hand, it is more expensive as it has higher costs due to license fees and hardware. Another drawback is the low immunity against interference.

[37] https://de.farnell.com/wcsstore/ExtendedSitesCatalogAssetStore/cms/asset/images/common/campaign/internet_of_things/architecture-lora.jpg

Sigfox is an LPWAN which was developed by the French telecommunications company Sigfox. It uses an unlicensed ISM frequency band. The technology is standardized by Sigfox in collaboration with European Telecommunication Standard Institute (ETSI). An advantage compared to LoRa is the range of up to 40 km in rural areas. However, the bandwidth and data rate are very limited.

Table 3: Comparison of LPWAN Technologies[38,39]

	LoRa	Sigfox	NB-IoT
Data Rate	50 kbps	100 bps	200 kbps
Modulation	CSS	BPSK	QPSK
Interference Immunity	Very high	Very high	Low
Range	5 km (urban), 20 km (rural)	10 km (urban), 40 km (rural)	1 km (urban), 10 km (rural)
Spectrum [MHz]	868	868	700 - 900
Band	ISM, unlicensed	ISM, unlicensed	Cellular, licensed
Bandwidth	250 kHz and 125 kHz	100 Hz	180 kHz
Costs	Low	Low	High
Security	Yes	No	Yes

3.3.3. 4G and 5G Mobile Communication Networks

In contrast to LPWANs, mobile communication networks offer high bandwidth and low latency for long range radio communication. The 4th generation of mobile communication standards (4G) is the current state-of-the-art for mobile communication. The 5th generation (5G), being the successor of 4G, will be the future technology in that area. Both standards are standardized by the 3rd Generation Partnership Project (3GPP).

In general, all mobile networks have a very similar network architecture as shown in Figure 31. From a very high-level view, a network consists of a core network and a radio access network (RAN). The core network manages the traffic between various subnets and the connection of the RAN to the internet. A RAN consists of macro cells (base stations) which implement the radio communication with the end nodes. End nodes can communicate with any macro cell in range. The macro cell itself is connected to the core network. Additionally, 5G networks will make use of small cells for high data rates at shorter ranges. This also includes Massive MIMO (Multiple Input/Multiple Output) for higher network capacity.

The overall characteristics of 4G/5G are:

- Long range communication
- High bandwidth
- High data rates
- Low latency
- High costs
- High energy consumption

The 5G standard, being currently under development, will support three service profiles, namely Enhanced Mobile Broadband (eMBB) for higher speed, Massive Machine Type Communication (mMTC) for narrowband communication and Ultra Reliable Low Latency Communication (URLLC) for better reliability

[38] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, “A comperative study of LPWAN technologies for large-scale IoT deployment”, ICT Express, Volume 5, Issue 1, 2019

[39] M. Lauridsen, H. Nguyen, B. Vejlgaard, I. Z. Kovacs, P. Mogensen and M. Sorensen, "Coverage Comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km² Area," *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, Sydney, NSW, 2017, pp. 1-5.

and latencies down to less than 1ms. URLLC, which possibly will also enable real-time communication, will be part of 3GPP Release 16 in 2020 and can be of certain interest for future developments.

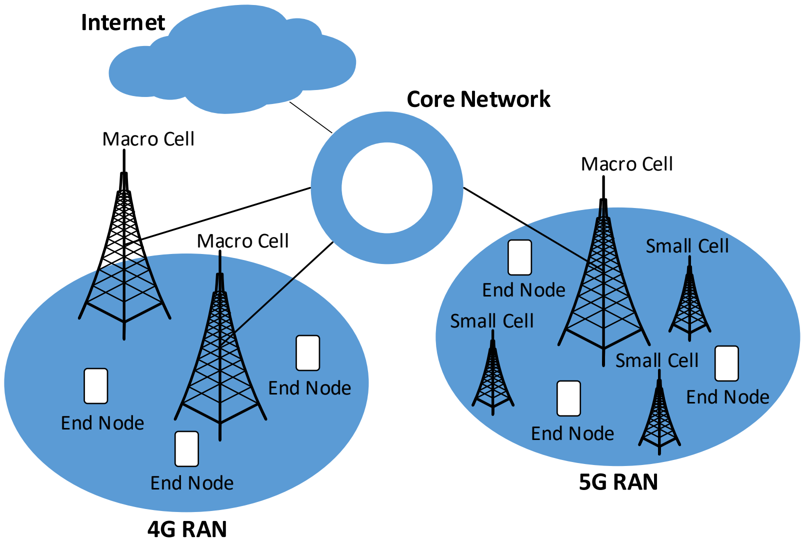


Figure 31: Simplified 4G and 5G Network Architecture

3.3.4. Ethernet and Time Sensitive Networking (TSN)

The IEEE 802.1 Time-sensitive Networking (TSN) is one of the future technologies in the industry and extends the standard IEEE 802.3 Ethernet with mechanisms for deterministic data transfer[40]. TSN is a family of standards which is still under development by the TSN task group. In particular, the following four components are described by these standards[41] as depicted in Figure 32:

- Synchronization
- Reliability
- Latency
- Resource management

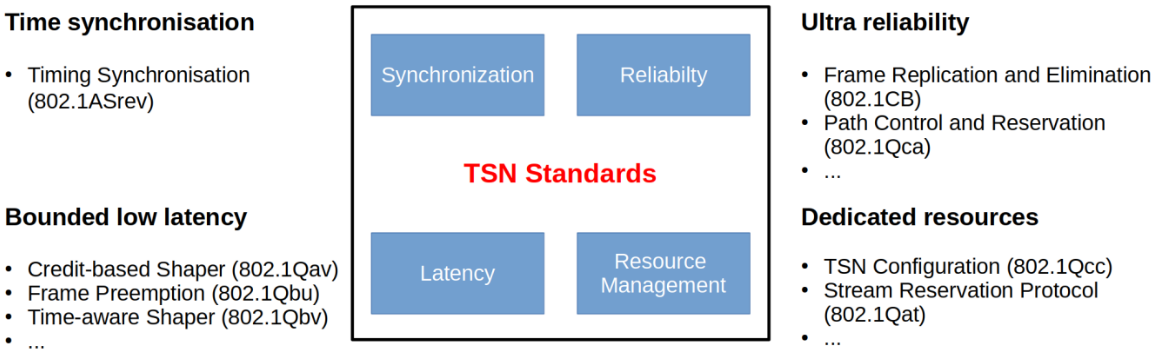

Time synchronisation
• Timing Synchronisation (802.1ASrev)
Bounded low latency
• Credit-based Shaper (802.1Qav)
• Frame Preemption (802.1Qbu)
• Time-aware Shaper (802.1Qbv)
• ...
Synchronization
Reliabilty
TSN Standards
Latency
Resource Management
Ultra reliability
• Frame Replication and Elimination (802.1CB)
• Path Control and Reservation (802.1Qca)
• ...
Dedicated resources
• TSN Configuration (802.1Qcc)
• Stream Reservation Protocol (802.1Qat)
• ...

Figure 32: TSN components[42]

Synchronization: In order to establish a common time base among the network participants and nodes, a generalized gPTP profile is provided in the standard IEEE 802.1ASrev for accurate time synchronization.

Latency: To meet real-time requirements and ensure low latency in data transmission, various scheduling algorithms have been developed and can be used at the same time. For instance, the time-triggered gating

[40] D. R. Hummen, S. Kehrer, and D. O. Kleineberg, "Tsn – time sensitive networking," Tech. Rep., 2017.
[41] A. Mildner, "Time sensitive networking for wireless networks – a state of the art analysis," 2019.
[42] D. Bruckner, M.-P. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, "An introduction to OPC UA TSN for industrial communication systems," Proceedings of the IEEE, vol. 107, no. 6, pp. 1121–1131, Jun. 2019.

mechanism “Time-aware Shaper” as specified in the IEEE 802.1Qbv standard is used for granting exclusive access to the transmission medium periodically for a limited time.

Reliability: Similar to the latency requirement, safety-critical real-time applications require high reliability to ensure correct functionality. For that purpose, the standard IEEE 802.1CB provides the replication of Ethernet frames, which are transmitted on disjoint paths. The elimination of the duplicated frames will be done on the last node in the network.

Resource management: According to the IEEE 802.1Qcc standard, which describes the configuration and the resource management of TSN systems, the following three architecture models exist:

- fully centralized
- fully distributed
- centralized network/distributed user model

The fully centralized model consists of a Centralized Network Configuration (CNC) and a Centralized User Configuration (CUC) system as depicted in Figure 33. The configuration of the network is done as follows: First, the talkers or listeners use an application-specific transfer protocol to send their network requirements directly to the CUC instance. The CUC sends the received requirements to the CNC to use them for resource allocation and configuring the underlying TSN bridges via NETCONF protocol[43].

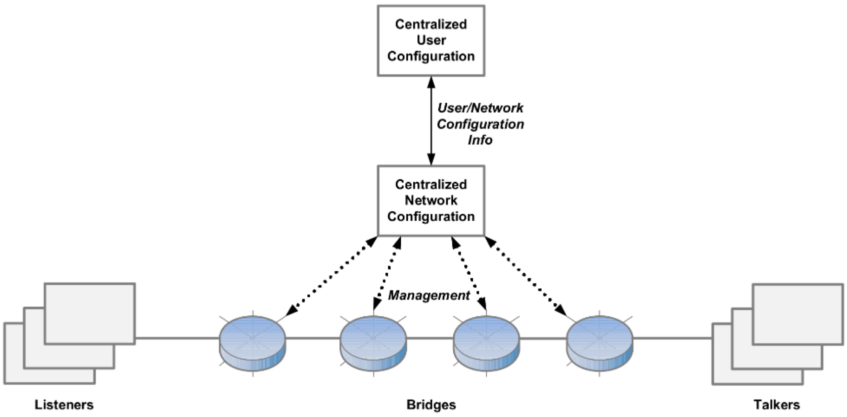

Centralized User Configuration
User/Network Configuration Info
Centralized Network Configuration
Management
Listeners
Bridges
Talkers

Figure 33: TSN architecture: Fully centralized model[44]

3.3.1. Short Range Communication – WLAN, Bluetooth (BLE), NFC, UWB and ZigBee

In addition to pure wireless LAN (WLAN), there are many wireless technologies such as Bluetooth, ZigBee, Near-field communication (NFC) and ultra-wideband (UWB) for short-range and proximity communication[45]. A comparison of the features of the mentioned wireless standards is given in Table 4.

- **WLAN:** Wireless LANs (WLAN) are wireless local area networks (LAN), based on the IEEE 802.11 family of standards, that transmit their data by radio. WLANs work with microwaves in the two frequency ranges of 2.4 GHz, in the ISM band, and 5 GHz. They use various modulation methods, coding techniques and radio transmission methods.
- **Bluetooth:** Bluetooth is a standard for short-range wireless communication. It was originally developed to use radio technology to eliminate the many cable connections between computers, peripherals, and consumer electronics devices. With the current version 5, devices can be connected with a distance up to 400m.

[43] A. Mildner, “Time sensitive networking for wireless networks – a state of the art analysis,” 2019.
[44] “IEEE standard for local and metropolitan area networks – bridges and bridged networks – amendment 31: Stream reservation protocol (srp) enhancements and performance improvements,” IEEE Std 802.1Qcc-2018 (Amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018), pp. 1–208, Oct 2018.
[45] https://www.itwissen.info/

- **ZigBee:** ZigBee is an IEEE 802.15.4-based specification for wireless networks with low data traffic such as home automation and sensor networks. It is a low data rate, low-power and close proximity wireless ad hoc network.
- **NFC:** Near Field Communication (NFC) is an international transmission standard based on RFID technology for the contactless exchange of data via electromagnetic induction using loosely coupled coils over short distances of a few centimetres. The NFC technology can be used for authorization or access control and for contactless payments with NFC credit cards.
- **UWB:** According to the definition of the Federal Communications Commission (FCC), ultra-wideband (UWB) is a broadband signal with a bandwidth of more than 500 MHz in the frequency range between 3.1 GHz and 10.6 GHz. This technology is a WLAN technology that offers higher transmission speeds than IEEE 802.11.

Table 4 Comparison of the features of key wireless standards[46]

	Bluetooth	ZigBee	WLAN	NFC	UWB
Frequency	2.4 GHz	sub-GHz / 2.4 GHz	2.4/5GHz	13.56MHz	3.1-10.6 GHz
Range (ideal condition)	100-400m	100m	50m	10cm	10-50m
Data Rate	<25Mbps	250kbps	600Mbps; ax: 10Gbps	424kbps	480-1320 Mbps
Power	Low	Low	High	Low	Low

4. IoT Interoperability

This layer is responsible for implementing the **interoperability** functions among the different modules of the I2PANEMA platform, as well as with other external systems.

The **IEEE**[47] defines interoperability as the ability of two or more systems or components to exchange information and use the information exchanged.

Figure 34: IEEE

Data source can be any kind of information useful for I2PANEMA:

- *Port ICT legacy* (PMS, PCS, TOS, etc., for more information check **D1.1. Stakeholders requirements and business scenarios**). Ports have their own information solutions that could provide a lot of usable data for I2PANEMA.
- *IoT platform*. To measure environmental information, ports could deploy some sensors managed by an IoT platform of their choice.
- *External services*. Services provide some information about, e.g., ship positions and environmental data like air quality.

As several kinds of sources exist that provide the same data with different formats, I2PANEMA Interoperability has to provide a standard view of each data type in order to import them in the same way into the I2PANEMA Data Management.

Before to speak about the state of the art of the interoperability it is important to know the benefits it can bring:

- Lower cost associated with interoperable systems as fewer resources and additional maintenance is required.

[46] https://www.electronicdesign.com/industrial-automation/examination-lpwan-technology-iot
[47] https://www.ieee.org/

- Access to information can be given to all appropriate stakeholders.
- Quality of data is improved as more sources can be brought together.
- Minimizes time needed to process data, thus increasing organizational efficiency.

It is possible to distinguish several initiatives related to the interoperability: International ones, like **WISE IoT** and **CREATE-IoT**, and research interoperability projects like **symbIoTe** and **bIoTope**.

In the following subsections some of the main interoperability initiatives from the literature for both types will be detailed.

4.1. International Initiatives

4.1.1. WISE IoT

WISE IoT[48] is a collaboration project between Europe and Korea.

Figure 35: WISE IoT logo

Objectives:

- It aims at deepening the interoperability and interworking of existing IoT systems.
- WISE IoT aims at building up federated and interoperable platforms ensuring end-to-end security and trust for reliable business environments with a multiplicity of IoT applications.

4.1.2. CREATE IoT

CREATE-IoT[49]'s goal is to stimulate collaboration between IoT initiatives, foster the take up of IoT in Europe and support the development and growth of IoT ecosystems based on open technologies and platforms.

Figure 36: CREATE-IoT logo

Actions to be included:

- CREATE-IoT is aligned with the Alliance for Internet of Things Innovation (AIOTI), coordinating and supporting the maintenance of the ecosystems developed through mapping the pilot architecture approaches, address interoperability and standards approaches, and semantic levels for object connectivity, protocols, data formats, privacy, security, trusted IoT, open APIs, and share the road-mapping with international initiatives.

4.1.3. NIF

NIF[50] relates to other initiatives developed by the Malta Information Technology Agency (MITA). It deals with the development and support of public services (Information Systems Framework, Information Technology Strategic Plan, the ICT Governance Framework and the Compliance Management Framework).

Objectives:

- NIF will provide a series of tools and guidelines to support public sector organizations in undertaking interoperability initiatives. It introduces concepts of semantic interoperability and identifies steps towards the standardization of government data assets in this context.

[48] http://www.eglobalmark.com/wise-iot/
[49] https://european-iot-pilots.eu/create-iot/
[50] https://mita.gov.mt/en/Technology/Initiatives/Interoperability/Document/NIF_framework.pdf

4.1.4. Semantic Sensor Network Ontology

To describe sensors, actuators, samplers, their observations, actuation, and sampling activities, World Wide Web Consortium (W3C[51]) and Open Geospatial Consortium (OGC[52]) Spatial Data on the Web (SDW) working group developed a set of ontologies which includes a lightweight core module called SOSA[53] (Sensor, Observation, Sampler, and Actuator) and an extension module called SSN[54] (Semantic Sensor Network). (Haller et al., 2018)

The SSN has a horizontal and vertical modularization architecture as depicted in Figure 37. It includes SOSA (Sensor, Observation, Sample and Actuator) for its basic classes and properties. SSN and SOSA can support a wide range of applications and use cases from large-scale scientific monitoring, industrial and home infrastructures, ontology engineering, to the Web of Things. (Semantic Sensor Network Ontology W3.org, 2017)

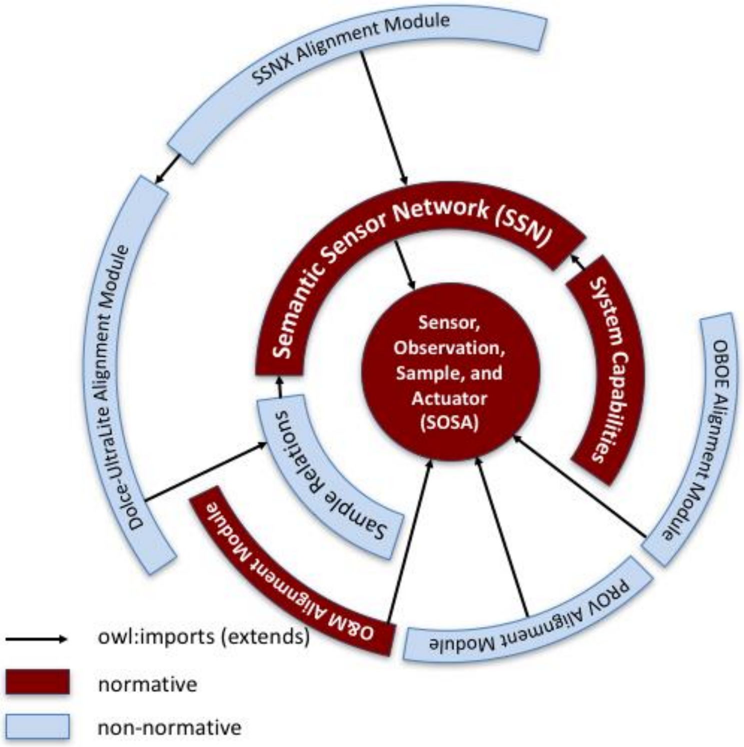


Figure 37: The SOSA and SSN ontologies and their vertical and horizontal modules

(Source: Semantic Sensor Network Ontology W3.org, 2017)

The SSN ontology is organized conceptually into ten modules as depicted in Figure 38. Sensors, their capabilities, observations and methods can be described by the SSN ontology. The ontology includes parts of the specification for sensors, like concepts for operating and the survival ranges. The deployment lifetime and sensing purpose of the deployed macro instrument are explained via structure for field distributions. Related, but not sensor specific specifications such as units of measurement, locations, hierarchies of sensor types, and feature and property hierarchies were excluded from the ontology. External ontologies can also be linked to the definitions via sub-classing or equivalence relations. (Compton et al., 2012)

[51] https://www.w3.org/
[52] https://www.opengeospatial.org/
[53] http://www.w3.org/ns/sosa
[54] http://www.w3.org/ns/ssn

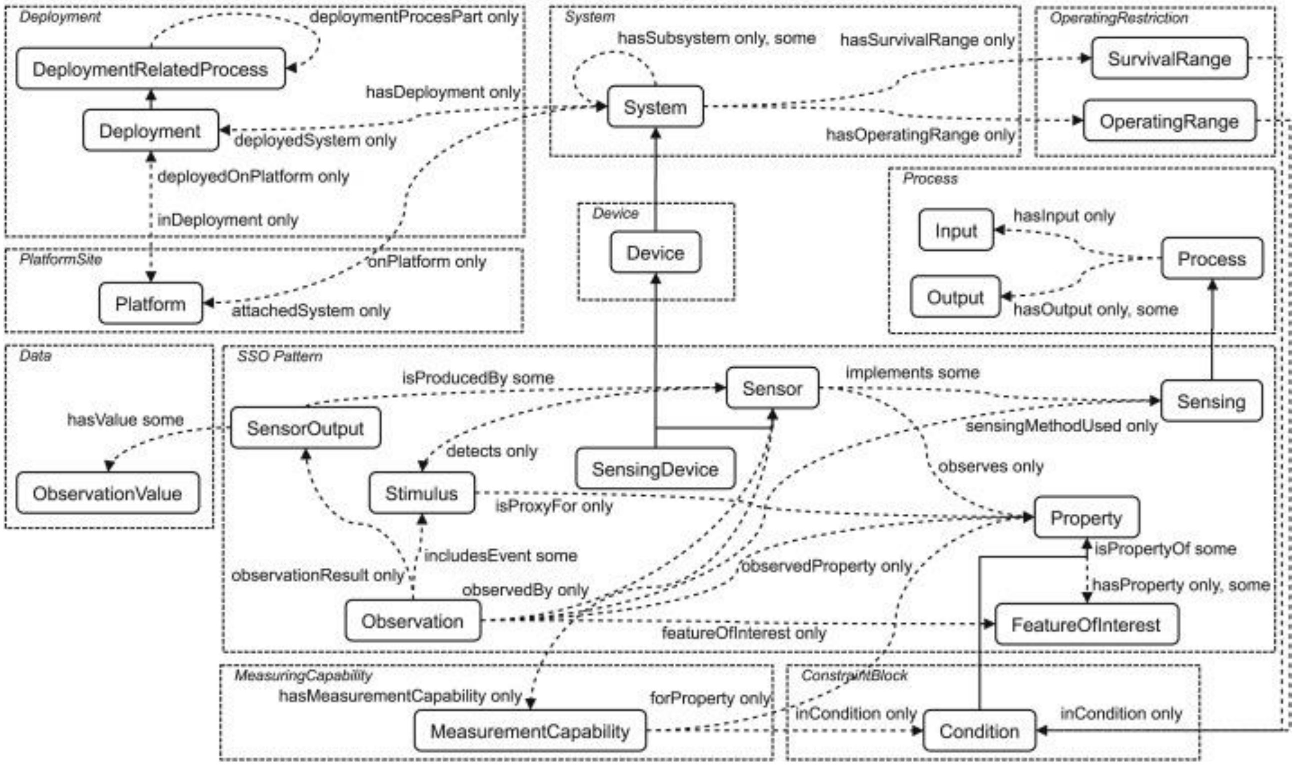


Figure 38 The SSN ontology, key concepts and relations, split by conceptual modules. (Source: Compton et al., 2012)

4.1.5. Observations and Measurements

Observations and Measurements (O&M[55]) is another international standard based on Geographical Markup Language (GML) that provides a common basis for location-based information. The O&M model as shown in Figure 39 can be used for modelling observation events and defining their relationships, measured properties and measurement procedure, and data from these observation events. It can also be used to model predicted information such as weather forecasts or to capture time-based changes to objects of interest. O&M offers system-independent, internet-enabled ways of data exchange between different parts of the sensor network and other systems using captured sensor information together with other open standards including SensorML and Sensor Observation Service (SOS). (O&M, 2012)

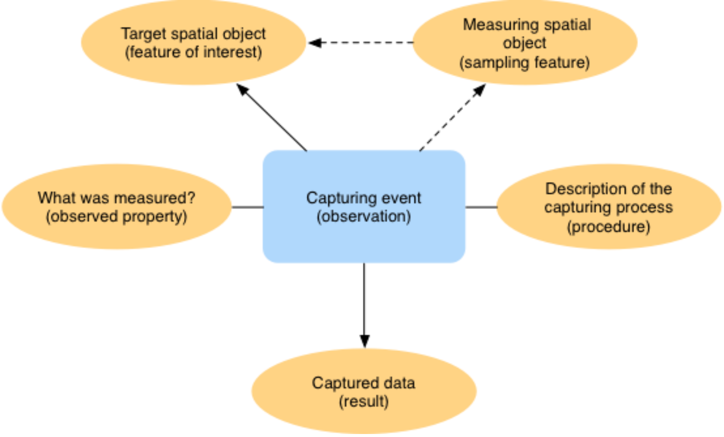


Figure 39 O&M definition of properties for an observation (Source: O&M, 2012)

[55] https://www.opengeospatial.org/standards/om

The XML[56] implementation of the O&M model is also standardized by OGC and a JSON[57] data transmission encoding model is added.

4.1.6. Sensor Model Language (SensorML)

The SensorML[58] standard, accepted by The Open Geospatial Consortium, includes generic models and an XML encoding to define sensors and measurement processes. A wide variety of sensors, including both dynamic and stationary platforms as well as in-situ and remote sensors can be represented with SensorML (SensorML, 2019).

Supported functionalities are:

- sensor discovery
- sensor geolocation
- processing of sensor observations
- a sensor programming mechanism
- subscription to sensor alerts

The principal goal of SensorML is to enable interoperability at syntactic and semantic level so that sensors and processes can be better understood by machines, utilized automatically in dynamic workflows, and easily exchanged among smart sensor web nodes (OGC® SensorML: Model and XML Encoding Standard, 2014).

All components including physical processors and hardware are modelled as processors that receive input and, through the application of an algorithm, generate output. Consequently, all such components may engage in process networks. Aggregate processes with their own inputs, outputs and parameters are themselves processes. SensorML process specifications are agnostic of environment and its models and encodings have been broken down into several conformance classes that facilitate the use of SensorML within specific frameworks. Therefore, the code must only comply with non-physical process requirements when SensorML is used for calculations. Likewise, a piece of software may only define internal processes using SensorML by adhering to the Simple Process conformance class. Nonetheless, all derived SensorML-based models and encodings must follow the key SensorML principles, whether they are solely associated with non-physical computational processes or with sensor systems (OGC® SensorML: Model and XML Encoding Standard, 2014).

4.1.7. OGC SensorThings API

Observations and Measurements standard (O&M) is the foundation of the SensorThings API[59] that establishes an interoperable conceptual model for observations and the functionalities in the sampling of observations. The features are modelled as *Things, Sensors* and *Features of Interest*. The SensorThings API provides two profiles: The *sensing part* and the *tasking part*. The sensing part offers a generic way of managing and retrieving observations and metadata from heterogeneous IoT sensor systems while the tasking part offers a standardized way to parameterize IoT devices. The sensing part is based on the O&M framework which enables IoT devices and apps to *Create, Write, Edit, and Delete* IoT information and metadata in a SensorThings network. (SensorThings API, 2019)

Figure 40 depicts the UML diagram of the SensorThings API data model. An IoT device can be modelled as a *Thing* entity that uses *Location* to describe the current location and *HistoricalLocation* for past trajectory. A *Thing* can have one or more *Datastreams*, where each *Datastream* is a logical combination of sensor *Observations* of an *ObservedProperty* produced by the same *Sensor*. *Observations* may observe different *FeaturesOfInterest* (Source: Huang and Chen, 2019).

[56] http://portal.opengeospatial.org/files/?artifact_id=41510
[57] https://portal.opengeospatial.org/files/64910
[58] https://www.opengeospatial.org/standards/sensorml
[59] https://www.opengeospatial.org/standards/sensorthings

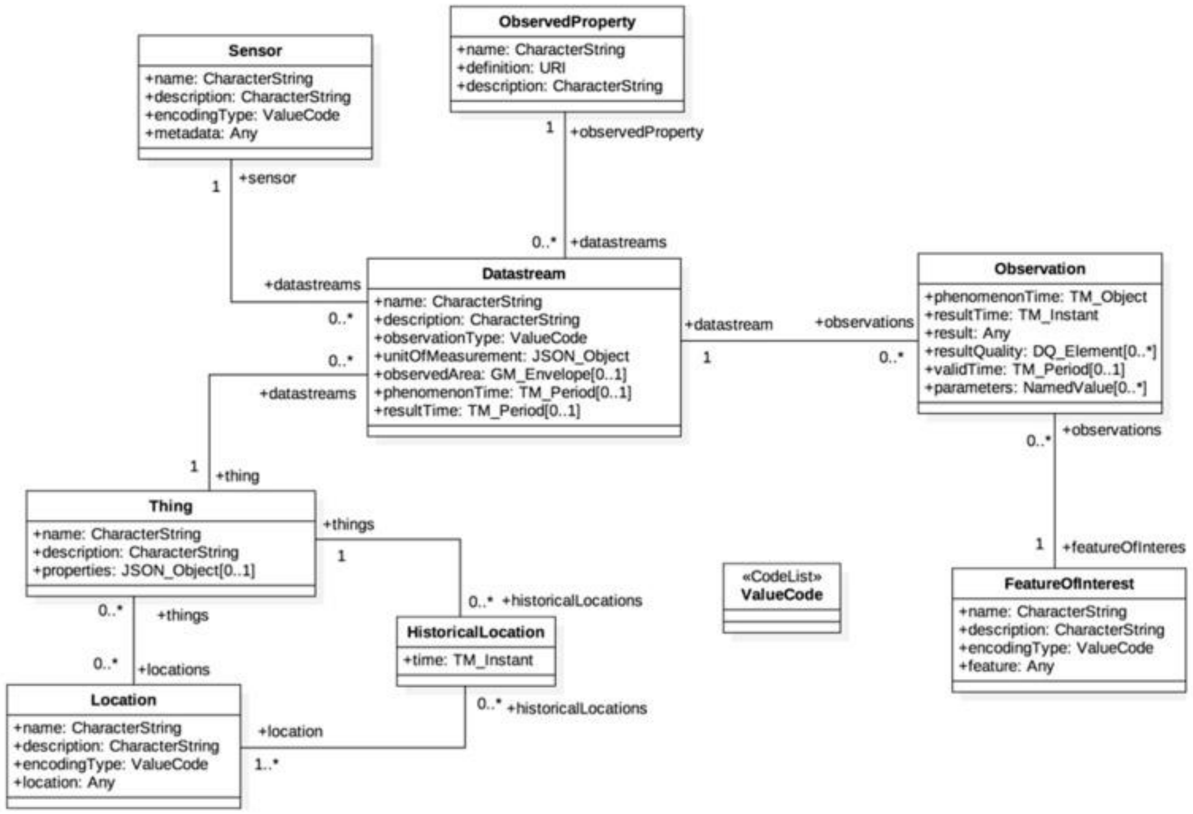


Figure 40 The SensorThings API UML diagram (Source: Huang and Chen, 2019)

4.1.8. Example. Systems that work with Interoperability: RIS

River Information Services[60] (RIS) are modern traffic management systems enhancing a swift electronic data transfer between water and shore through in-advance and real-time exchange of information. An EU framework directive provides minimum requirements to enable cross border compatibility of national systems.

Figure 41: RIS logo

Its objective is the deployment of interoperable, harmonized solutions. This includes services such as fairway information, traffic information, traffic management, calamity abatement support, statistics and customs services, waterway charges and port dues.

Actions to be included:

- The definition and implementation of harmonized concepts of data exchange between authorities, inland waterway transport operators and users.
- The interconnection of national systems and interconnection with systems at union level.
- Reduction of administrative burden and elimination of paper flow documents, establishing solutions to facilitate machine to machine communication.

[60] http://www.ris.eu/projects

4.2. Research Interoperability Projects

IoT-EPI[61] is a European Initiative for IoT platform development. It was formed to build a sustainable IoT-ecosystem in Europe, maximizing the opportunities for platform development, interoperability and information sharing.

Figure 42: IoT-EPI

Among the research and innovation projects risen from IoT-EPI are:

- symbIoTe
- bIoTope
- BIG IoT
- VICINITY
- INTER-IoT

These projects address in some components the specific problem of interoperability.

4.2.1. symbIoTe

symbIoTe aims to create a framework for interoperability between existing and future IoT platforms as shown in Figure 44. The framework will enable the discovery and sharing of resources for rapid cross-platform application development.

Figure 43: symbIoTe logo

To achieve this, symbIoTe will provide:

- An abstraction layer for a "unified view" on the resources of various platforms.
- symbIoTe will enable federation among IoT platforms, so that they can securely interoperate, collaborate and share resources for the mutual benefit.

[61] https://iot-epi.eu/

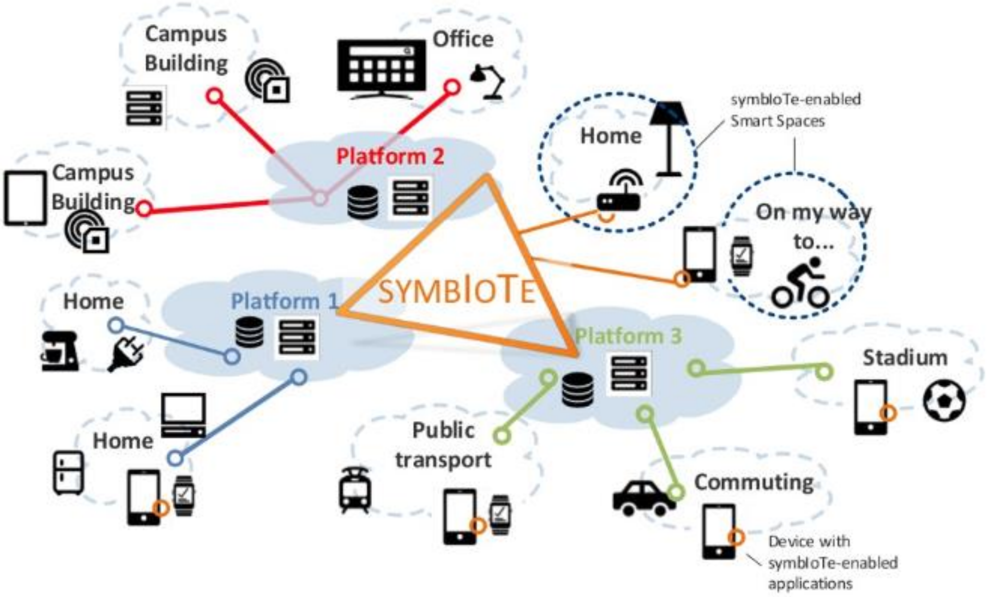


Figure 44: symbIoTe project
(Source, https://www.symbiote-h2020.eu/)

4.2.2. bIoTope

bIoTope[62] (Building an IoT open innovation ecosystem for connected smart objects) brings the opportunity to create open innovation ecosystems. It provides a platform where companies can easily create new IoT systems, reducing costs and changing how services are sold and consumed.

Figure 45: bIoTope logo

For this purpose, an advanced System-of-Systems (SoS) capability for Connected Smart Objects is used as depicted in Figure 46 .

[62] https://iot-epi.eu/project/biotope/

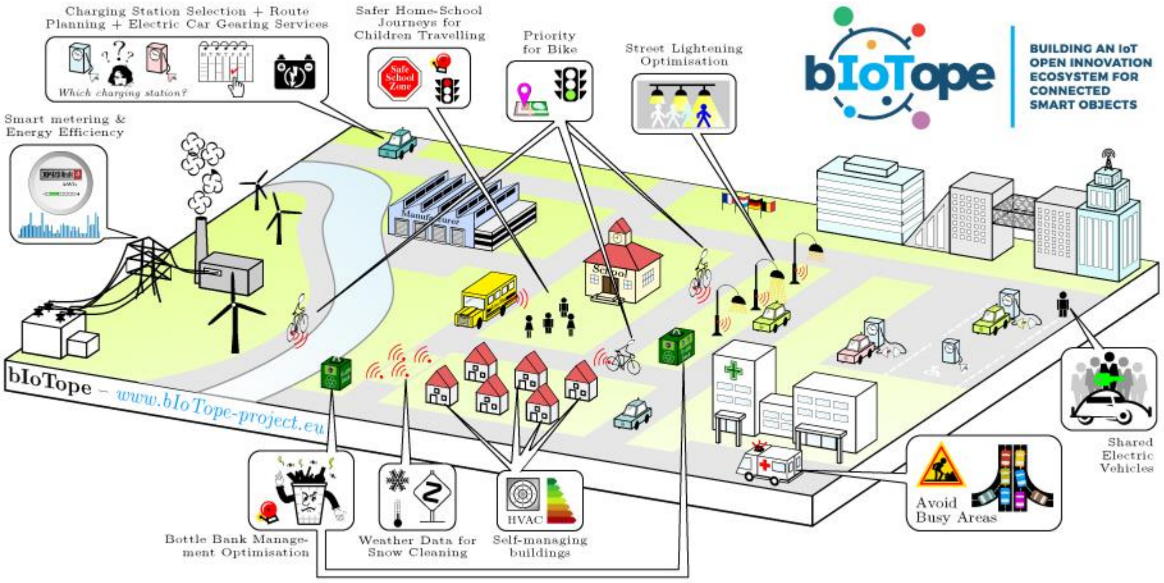

Charging Station Selection + Route Planning + Electric Car Gearing Services
Which charging station?
Safer Home-School Journeys for Children Travelling
Safe School Zone
Priority for Bike
Street Lightening Optimisation
bIoTope
BUILDING AN IoT OPEN INNOVATION ECOSYSTEM FOR CONNECTED SMART OBJECTS
Smart metering & Energy Efficiency
Manufacturer
School
bIoTope – www.bIoTope-project.eu
Bottle Bank Management Optimisation
Weather Data for Snow Cleaning
HVAC
Self-managing buildings
Avoid Busy Areas
Shared Electric Vehicles

Figure 46: bIoTope project

4.2.3. BIG IoT

BIG IoT[63] will address the interoperability gap by defining a generic, unified Web API for smart object platforms called the BIG IoT API. Interoperability is realized through the API (as a de-facto standard).

Figure 47: BIG IoT logo

The establishment of a marketplace where platform, application, and service providers can monetize their assets will introduce an incentive to grant access to formerly closed systems and lower market entry barriers for developers.

[63] https://iot-epi.eu/project/big-iot/

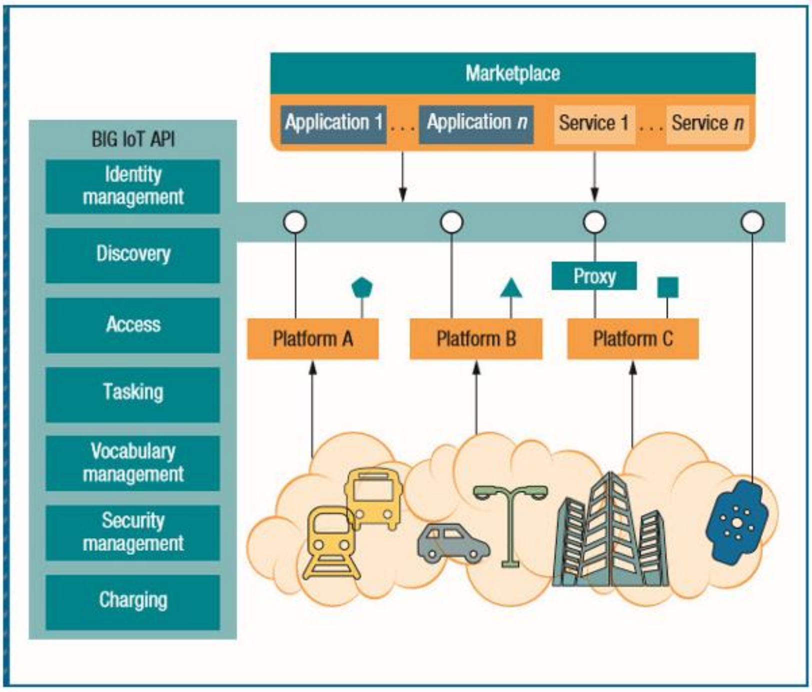


Figure 48: BIG IoT project

4.2.4. VICINITY

The **VICINITY**[64] project will build and demonstrate a platform and ecosystem that provides "interoperability as a service" for infrastructures in the Internet of Things.

Figure 49: VICINITY logo

To achieve this interoperability it will make use of semantics between domains.

[64] https://iot-epi.eu/project/vicinity/

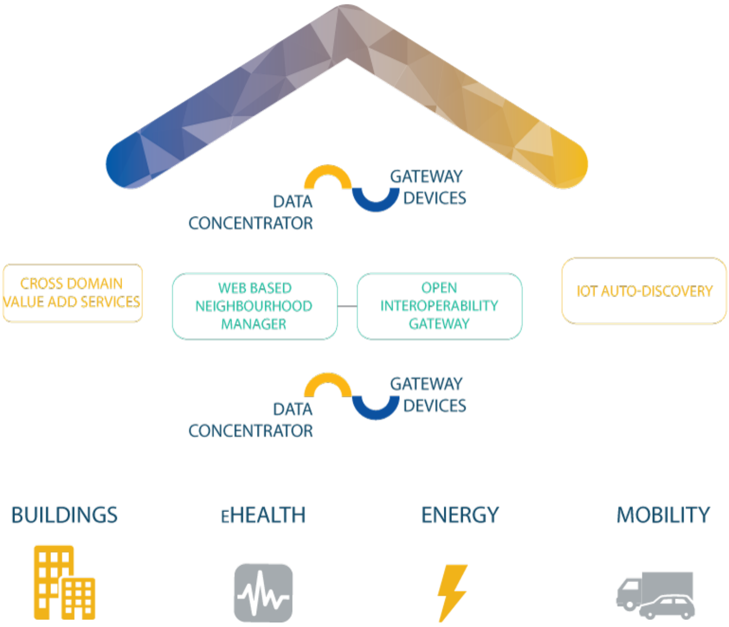


Figure 50: VICINITY project (Source, https://vicinity2020.eu/vicinity/)

4.2.5. INTER-IoT

The goal of **INTER-IoT**[65] is to design, implement and test a framework that will allow interoperability among different IoT platforms.

Figure 51: INTER-IoT logo

The proposal will allow effective and efficient development of adaptive, smart IoT applications and services, atop different heterogeneous IoT platforms, spanning single and/or multiple application domains.

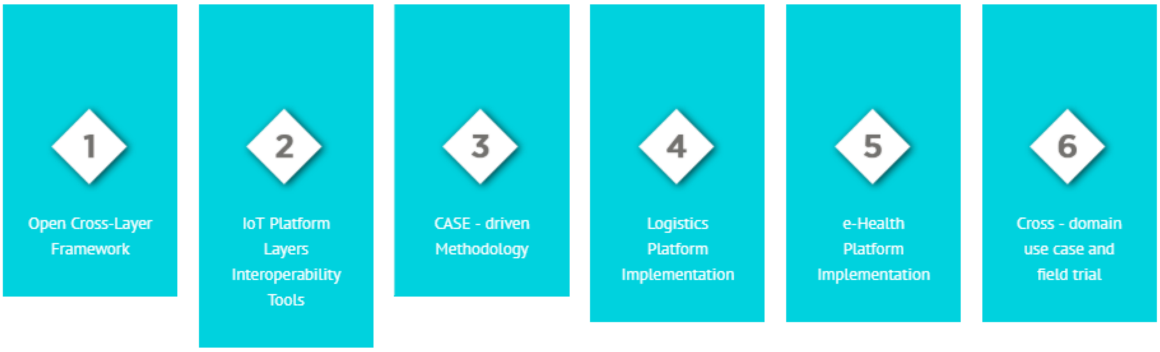

1
Open Cross-Layer Framework
2
IoT Platform Layers Interoperability Tools
3
CASE - driven Methodology
4
Logistics Platform Implementation
5
e-Health Platform Implementation
6
Cross - domain use case and field trial

Figure 52: INTER-IoT objectives (Source, https://inter-iot.eu/objectives)

Figure 53 depicts the layers involved in the INTER-Framework and the interaction among them.

[65] https://iot-epi.eu/project/inter-iot/

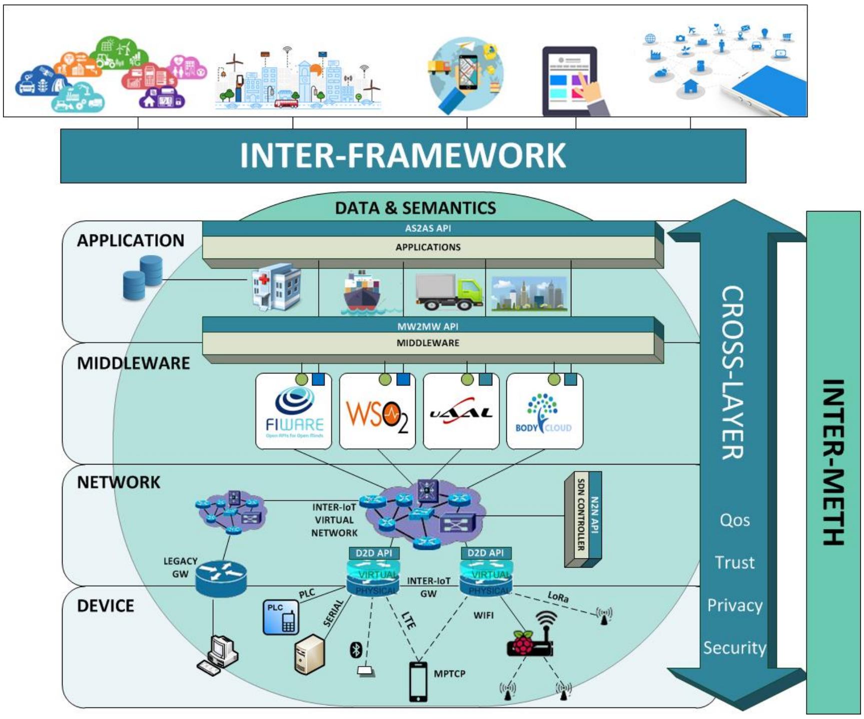


Figure 53: INTER-Framework

5. IoT Data Management

“Data is what enables the integration of two worlds (physical and cyber)” (Raptis, 2019).

The field of Industrial IoT is influenced by the fields of logistic engineering, computer science and business engineering. Therefore (Raptis, 2019) proposes the distinction between data enabling technologies and data centric services. This fits easily to the traditional automation pyramid.

Data enabling technologies provide the technological foundation and data centric services operate on the technological foundation to derive or provide higher-level business values from that foundation.

5.1. Data Properties of I4.0 Use Cases

(Raptis, 2019) has identified four properties for describing IoT data characteristics, based on an extensive literature review of real use cases. These four properties are data volume, diversity, traffic, and criticality:

Data volume. This property describes the amount of data that is required by an use case. The authors propose three values for this property: “small”, “medium” and “large” data volume, and provide the following examples: an use case with small data volume only processes sensor measurements, an use case with medium data volume processes images and sound files and an use case with large data volumes processes videos and 3D representations.

Data traffic. This property describes the diversity of the traffic patterns of an use case. They propose two values for this property, namely “intense” (use case requires measurement and/or analysis of a large volume of data in little time) and “mild” (use case has no such requirements).

Data criticality. This property indicates how important it is for the system to react quickly to new data. The authors define two possible values for this property: “low importance” and “high importance”. An example of an use case with a high data critically is the real-time monitoring of chemical pipelines. A system to monitor for a leakage of the pipeline must process and react to new data in a very short time. In contrast, a system that monitors the daily flow through a pipeline only requires a mild data critically.

Data variety. The diversity of the data used in a use-case. They propose the two values “diverse” and “uniform”. A use case with “diverse” data variety uses multiple types of data (for example CAD models, video data and acceleration measurements). A use case with “uniform” data variety processes only data which is semantically very similar.

5.2. Critical Characteristics of Data Management for Transactional Data

Laney's analysis of three critical characteristics for managing large amounts of transactional data (D. Laney, 2001) has shaped a concept of data management that has become popular with the term Big Data. These three critical characteristics are data volume, data speed and data diversity.

Laney has identified the criticality of these characteristics by analysing e-commerce applications. A description and the proposed management approach for the characteristics follows.

Data volume. The amount of data that must be stored, processed, and that is exchanged in total. Data volume is also considered through a transaction. The obvious approach to process an increasing amount of data is to increase the data storage resources. To limit the increase and need for resources, Laney suggests focusing on the value of a data source to the business process, eliminating duplicates in the data sources, and focusing on an unbiased sample of available data.

Data speed. Data velocity is the property of the velocity of a system to process data and to influence business processes. To address this characteristic, Laney proposes to distinction between a transactional view of data and an integrated and reorganized view of the same data and to further establish direct access between applications which requires a high data velocity and data sources.

Data variety. Data variety denotes the challenge that a system has to handle data with different semantics, structures and data access interfaces. For the first two challenges, Laney proposes to adapt data sources to a common data structure (e.g. XML or JSON), to establish a management of the metadata of data sources and more sophisticated indexing strategies. For the last challenge, Laney proposes to use a data access middleware, using distributed data queries and message queues.

5.3. Comparison of both sets of Data Characteristics

This section describes how the characteristics for IoT data management defined by (Raptis, 2019) and the set of characteristics for processing transactional data by Laney map to each other.

The notion of data volume in both sets is very similar.

For IoT data, (Raptis, 2019) splits the notion of data speed in two parts, data traffic and data criticality. Data traffic denotes the velocity in which an use case processes new data and data criticality denotes the velocity in which an use case has derived a response to new data.

This distinction reflects the differences in the criticality of the response time of both application domains: production control and e-commerce.

Raptis focusses the notion of data variety on data semantics, while Laney extends the notion to data structure management and metadata management.

To some extent, data management concepts and technologies to process transaction data fits to process IoT data in the I2Panema project.

5.4. IoT Data Characteristics for the I2PANEMA Project

In summary, the following characteristics of IoT data must be considered by IoT data management.

An important aspect for the architectures of I2PANEMA business cases is that external data can be ingested on both layers. The shop floor layer focusses on ingestion and pre-processing of data from devices in the field. Therefore, the IoT platform must be able to pre-process the data before it is forwarded to the office-floor layer. This pre-processing can include both the standardization of the data semantics and the appending and managing of metadata to the device data.

Most I2PANEMA business cases require only a small to medium data volume with a low data variety. The business cases are probably subject to intensive data traffic patterns.

A response time in the order of minutes is sufficient for most business cases, with the notable exception of the Nuremberg business case and the revised Wesel business case. A response time in the microsecond range is required for these business cases.

Therefore, the IoT platform must offer the possibility of low-latency routing of data to and from a software component at the shop-floor layer. Further, the software component must process and respond to new data with a very low latency.

On the office floor layer, the I2PANEMA platform must provide a more unified data access with more condensed data structures and semantics. Further, it requires the ability to execute data intensive workflows and routing the data to a large variety of ICT-services of very different scale.

Further, the office floor layer should provide a centralized persistent data storage.

5.5. Technological Choices

For the shop floor layer previous sections have discussed technological choices in details, i.e. IoT communication technologies (section 3) and IoT interoperability (section 4). Therefore, this section focuses on technological choices for shop floor level computation and office floor data routing.

5.6. Office Floor Layer Data Management Architecture

The office floor architecture for data management consists of several parts conceptually divided into three groups of components responsible for the business logic in I2PANEMA:

Data Warehouse. This group of components is responsible for data collection from the shop floor layer and other ICT systems, allowing users to perform advanced analysis on the data received. In the back-end, data analysis techniques will be implemented. The results obtained by these algorithms will be shown by means of the corresponding data visualization interfaces. This component consists of three parts:

- Collection and pre-processing of the data captured by the devices or stored in other ICT systems.
- Storage of information in unstructured databases (NoSQL databases) or a distributed file system.
- Routing and distribution of data between all components of the I2PANEMA platform

Data Analytics. This group of components derives knowledge from data and provides predictions. This component is the value-adding part in a business case and should be strongly adapted to an use-case needs.

Data Visualization. This group of components visualizes the data provided by the data warehouse and the prediction and knowledge provided by the data analytics.

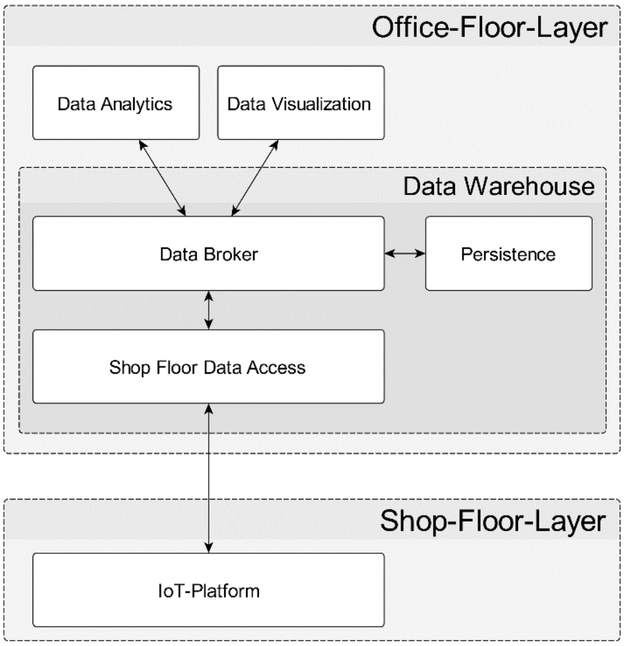


Figure 54: Office Floor Layer Architecture for Data Management

5.6.1. Container Engine as Runtime Environment

It is recommended to use a container engine as runtime environment for all parts of the office floor layer. The concept of containerization denotes the idea to bundle an application and all its runtime dependencies into a single file, to be able to deploy an application in a fully automated manner and in an isolated environment. The currently most common used container engine is *Docker*.

In most use cases, multiple applications distributed each in an individual container must interact to achieve a business outcome. Therefore, we use *Kubernetes* as a container management and composition system focusing on container scaling and fail-safe provisioning. It can deploy multiple instances of the same container on different hosts and manages the network and routing to those container instances. For I2PANEMA, *Kubernetes* provides an execution environment to deploy components and services on clusters of hosts and has the ability to configure scaling and high-availability of the hosts. Another interesting solution is *Rancher*, also an open source solution for application container management.

In the past, *Docker Swarm* was also an important component to run multiple instances of different application containers. Furthermore, it could create and manage clusters of *Docker* runtime. The development has been stopped and the solution was officially replaced by *Kubernetes* as the default container orchestration solution of *Docker*.

5.7. Data Warehouse Components

This section describes components of the data warehouse. These components are based on the SMACK technology stack for a data warehouse. In this section frameworks used by the SMACK technology stack are described.

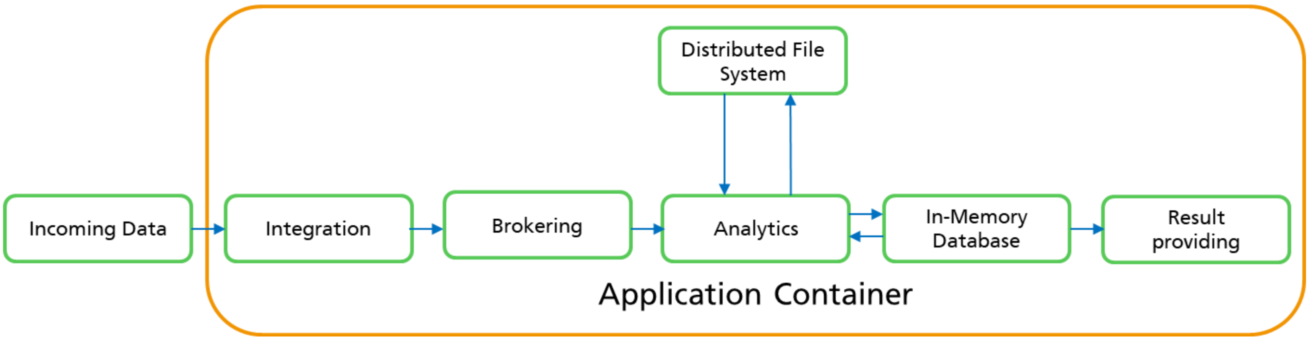


Figure 55: Data Warehouse Components

5.7.1. Integration Component

The integration component (and result providing) is used to integrate and add data to a data warehouse. Additionally, the same component provides any results and data to a service or IT-System. Today, the standard technology is called Akka. Akka is a toolkit for actor-based concurrency focused on message-based processing for communication between multiple threads instead of global memory and locks. Akka is an implementation of the actor-based concurrency on the Java Virtual Machine (JVM) with an uniform communication approach between actors on one host and between actors on different hosts.

Akka could provide the building block for the data access component as well as the data persistence connector. Further, Akka might be a good choice for a service that derives new actor settings on the shop floor layer for low response time.

5.7.2. Apache Kafka as the Broking Component for the Data Warehouse

Apache Kafka is a distributed message-oriented middleware which stores and forwards messages. Kafka offers advanced data distribution patterns such as the grouping of several instances of the same service that process all messages of a certain type or certain guarantees for data delivery. It is highly flexible and supports different types of message deliveries.

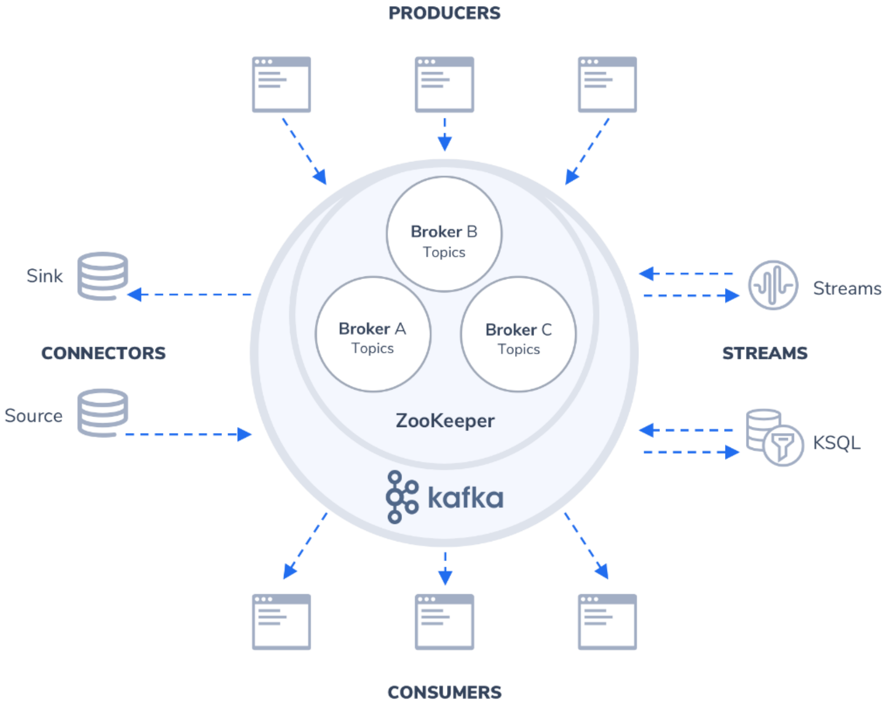


Figure 56: Apache Kafka
Taken from coralogix.com/log-analytics-blog/a-complete-introduction-to-apache-kafka

Kafka uses a publish-subscribe messaging system based on topics. At a Kafka instance you subscribe to a topic. Kafka streams read data from a topic after their subscription. Furthermore, Kafka has an SQL engine called KSQL to create queries on data streams. Finally, Kafka Connect is a framework that can be used to import data into Kafka.

In our proposed architecture, Kafka is used as the data broker that connects and transfers data between all parts in the components above or below.

5.8. Data Persistence

A data warehouse needs two kinds of data persistence. A storage component for vast amounts of data, that are stored for a long term and a fast short-term storage. Today, there a two primary solutions for long term storage.

First a distributed file system (e.g. HDFS – Hadoop File System), that combines several servers (with a lot of hard drive for storing data) into one big data storage system. The second solution is to use a modern database. For IoT data management, NoSQL databases are the most appropriate technology. Cassandra and MongoDB are two common NoSQL databases. MongoDB is a database that directly manages documents and allows clients to submit data and query existing documents similar to the JavaScript Object Notation (JSON). Especially, MongoDB is built to store data in JSON-like structures. Any sensor or IoT data can be transferred to JSON and stored as-is in a MongoDB. Apache Cassandra is a column-oriented database with a row-wise partition. This combination makes Cassandra clusters highly scalable without a single point of failure.

The advantage of NoSQL databases is that no structure must be defined to store and manage data. All data is stored as-is. A second advantage is (this does not count for every NoSQL database, but for MongoDB and Cassandra it does) you can run the databases as “In-Memory” databases. That means, the data are stored in the memory (RAM) of a server and not on a hard drive. This leads to a fast data read and write access, providing a short-term storage technology. For IoT data, small datasets and small messages are common, and for any near real-time usage (data is processed within seconds) a NoSQL database is the right technology to store the data during the processing. Data in “In-Memory” databases is only stored for a short term, but the data can be added to the storage component for long term storage.

6. IoT Operational Workflow

This layer is responsible for managing the operational workflows among the different actors involved in port activities (e.g. any task or activity in the port environment). Workflows are defined by the usual structures for specifying tasks, such as status information (in progress, confirmed, completed, etc.) but they can also have a specific part according to the actual task they describe (such as loading time, ship arrival, validated temperature, etc.). This layer will provide support to automate collaborative information flows and to connect to devices capable of receiving instructions in real time, facilitating the implementation of ad-hoc processes.

6.1. Introduction

6.1.1. Challenges in logistics and port environment

As a result of globalisation, ports have become central hubs for cargo. Numerous parties and stakeholders are involved in highly networked and complex logistics structures and work together on cargo handling, processing, and transport. The same applies to the large amount of data processed and exchanged among the numerous actors in day-to-day port operations.

External pressure in terms of process improvement, of the ability to provide information and of the significantly grown demand for more flexible co-operation with service providers and other players is enormously demanding ports and logistics companies. Especially within this logistics sector, there exist large improvement potential w.r.t. data exchange among the actors along the maritime supply chain. In addition, manual operations, outdated data, data loss and inconsistency, various software and unstandardized data interfaces complicate the information flow and weaken the whole supply chain.

The use of new technologies for IT-supported port processes can thus ensure the seamless data exchange in the port environment for all involved parties.

6.1.2. Digitalization of processes as a solution: actor’s network

The aim of the I2PANEMA project is to implement IoT technologies in the port environment. With the obtained "intelligent" sensor data, consistent analyses, evaluations and, if necessary, control systems can be derived. Additionally, it is important to forward, evaluate and process securely the outgoing data, as a basis for decision-making, to the subsequent systems. Decisions may be, thereby, prepared or even made by AI means. This involves the data exchange and decision making, both, at the level of an actor and of the entire cooperation network. For this, a consistent and secure data networking of all actors is mandatory.

Logistics networks are known for their high number of different actors, and those in port areas, are no exception. The actors involved in the port environment are, for example, port operators, logistics service providers, railway undertakings (RUs) and transport companies, shippers, and port authorities, as well as consignees.

A possible communication structure among various actors is outlined in Figure 57. The complexity of the vertical and horizontal integration can be flexibly adjusted.

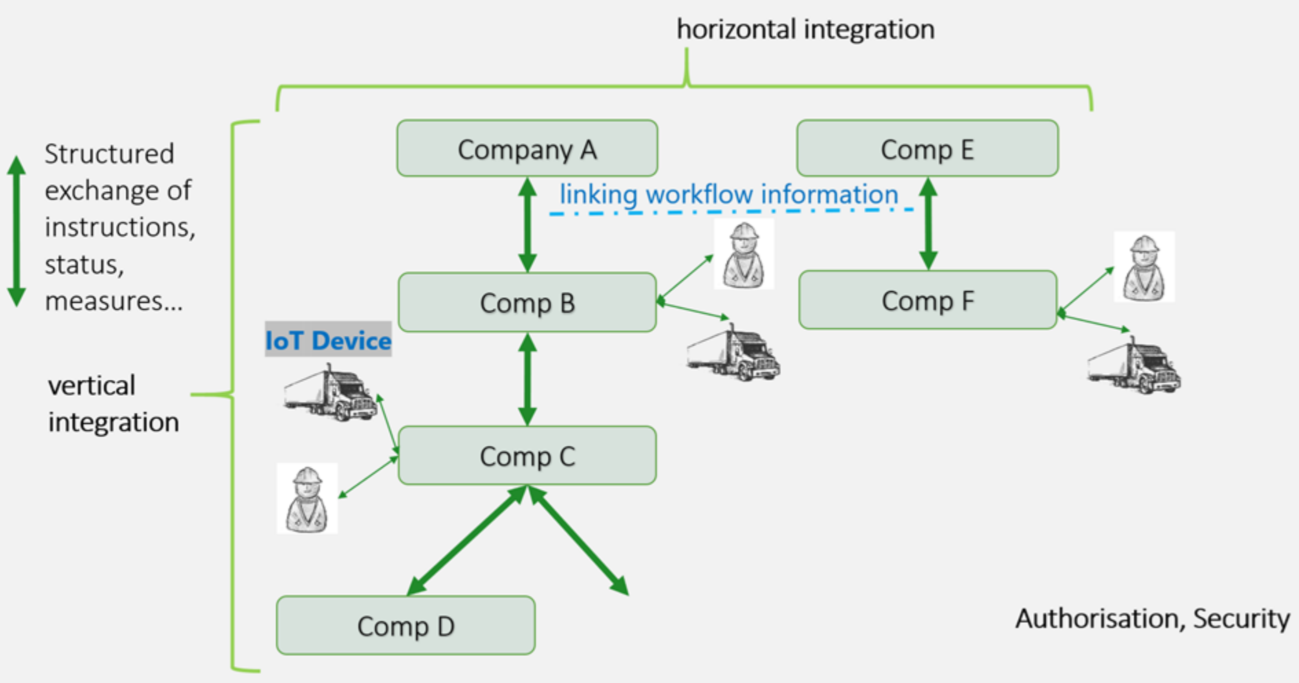
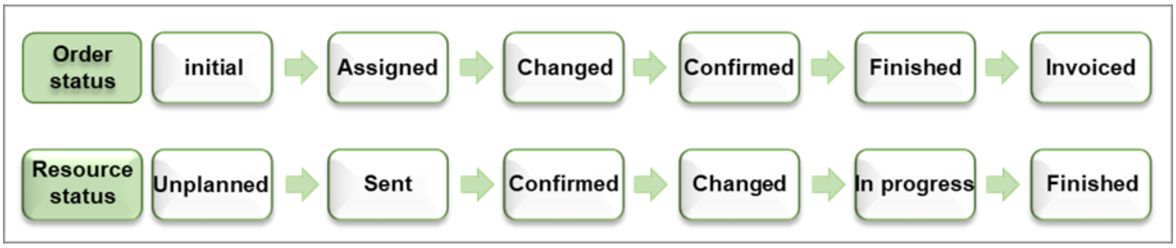


Figure 57: Operational Workflows in a maritime network

(Source: catkin GmbH)

Flow diagrams can represent typical use cases in logistics by illustrating the information flow between two or more players. As a rule, a task is planned on a time axis, instructions are sent to a contractor, the contractor reports status information, and measurements back to a planner.

A possible usage in an IoT environment is, that on the contractor`s side a device is assigned to a workflow instance. It receives instructions and reports measurements back. By way of these measurements progress information can be propagated, in form of a concrete status value (i.e. “arrived” or “loaded”) (see Figure 58 status information of an order and a resource as used in the catkin workflow management system).



Order status	initial	Assigned	Changed	Confirmed	Finished	Invoiced
Resource status	Unplanned	Sent	Confirmed	Changed	In progress	Finished

Figure 58: A status information of an order and resource

(Source: catkin GmbH)

This way, in the context of use cases, typical task situations can be tested, monitored and task structures can be refined while testing.

The challenges mentioned above demonstrate the importance of linking work processes and the information, exchanged in them, with the actors, who are involved in these processes. This can be realized by workflow management systems. The next chapter gives a theoretical introduction to this topic and an overview of some solutions, available on the market.

6.2. Operational Workflows

6.2.1. Collaborative Workflows: definition

A workflow is a "work-sharing and mostly recurring business process which defines the tasks, processing units, and their relationship network within a process".[66] An Operational Workflow defines the business processes for managing account, user, and business partner user entities.[67]

A collaborative workflow links these terms and describes work processes in which two or more participants work together. Collaborative workflows enable therefore the exchange of relevant process data and order information among actors of a network.

As the I2PANEMA project is focused on logistic processes and, above all, on those in maritime logistics, it is important to mention their complexity and high dynamics. In transport logistics, especially, in a port environment such operational collaborative workflows can be, for example:

- Transportations or replacements of a load unit
- A turnover on a terminal
- Check-in process, etc.

Workflows also play an important role w.r.t, the use of IoT systems. Due to their modularity and versatility, they bring important aspects to the IoT environment, such as: [68]

- State of a process in real time
- Fast reaction based on the information transmitted from various IoT- devices
- Fast information exchange due to the "low level IoT messages"
- Storage of the data collected from the IoT – devices
- Predictive algorithms can also be used

It is therefore obvious that the use of IoT devices can achieve much higher efficiency by connecting workflows to these IoT systems. This can be realized via the so-called workflow management systems, which can execute the aspects mentioned above.

In the next chapter, an overview on some workflow management tools existing on the market is provided.

6.3. Overview on Existing Workflow Management Systems

At the point in time when the importance and relevance of a company's resources were recognized, the first software solutions for resource management were developed. As for connecting the manpower resources with a company's processes and data management further software solutions were needed. Furthermore, with the growing usage of IoT, the management of other mobile resources (such as trucks, trains, or ships in the logistics supply chain) has also become essential. This was the impetus for the development of software solutions that combine all these functions ensuring the workflow management of a company.

[66] https://wirtschaftslexikon.gabler.de
[67] https://www.ibm.com/support/
[68] https://www.scitepress.org/Papers

In order to get an understanding, what kind of software solutions are available on the workflow management market, what features they have and what industries they focus on, an overview of some selected workflow management solutions is provided below

6.3.1. Low-code platform Mendix®

Using the low-code technology, Mendix[69] (Figure 59**Fehler! Verweisquelle konnte nicht gefunden werden.**) offers for example a platform for visual software development of processes.

This means, Mendix enables uncomplicated visual development of workflow solutions, independent from an application area.

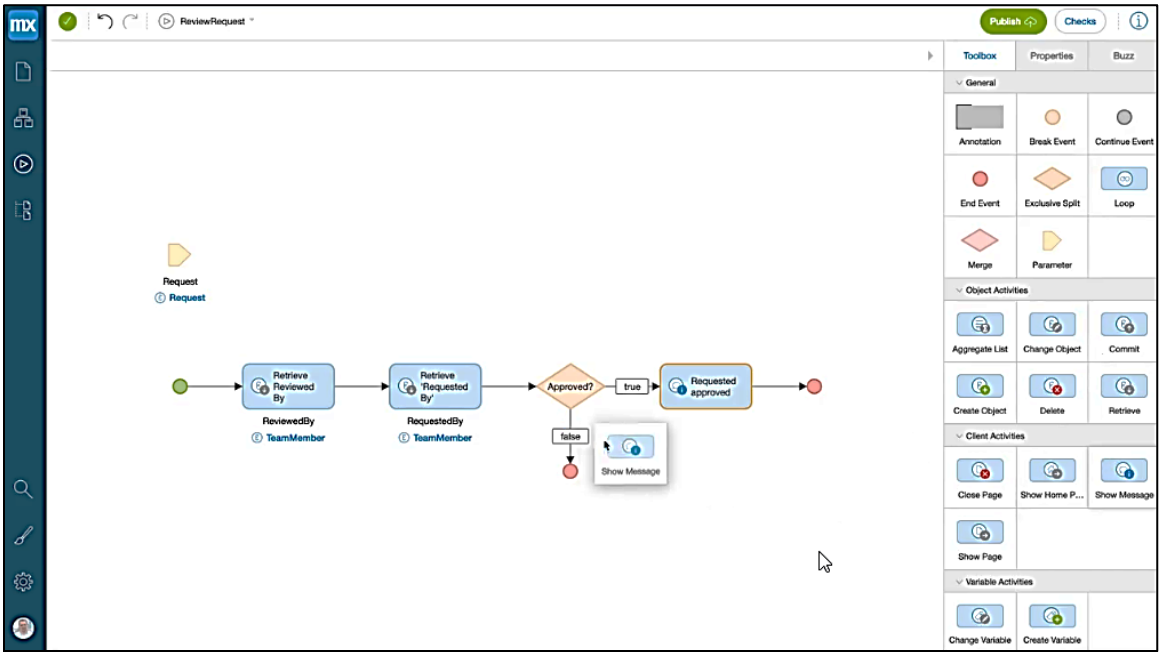


Figure 59: Workflows by Mendix

The most important functions of Mendix include following features:

- Visual development of applications by using the graphical elements of the software
- No programming with code
- Development of logic in a visual language
- Data storage in the cloud

However, as Mendix is a low-code platform designed for development, basic knowledge of programming is required. Mendix is not focused on any industry and is therefore not specialized in the logistics.

6.3.2. Workflow - based software Nintex®

Nintex [70] is an IT provider of workflow-based software (**Fehler! Verweisquelle konnte nicht gefunden werden.**) and offers an IT solution which enables designing of the workflows within a company and the information exchange among them.

[69] https://www.mendix.com/de/plattform/
[70] https://www.nintex.com/process-automation/

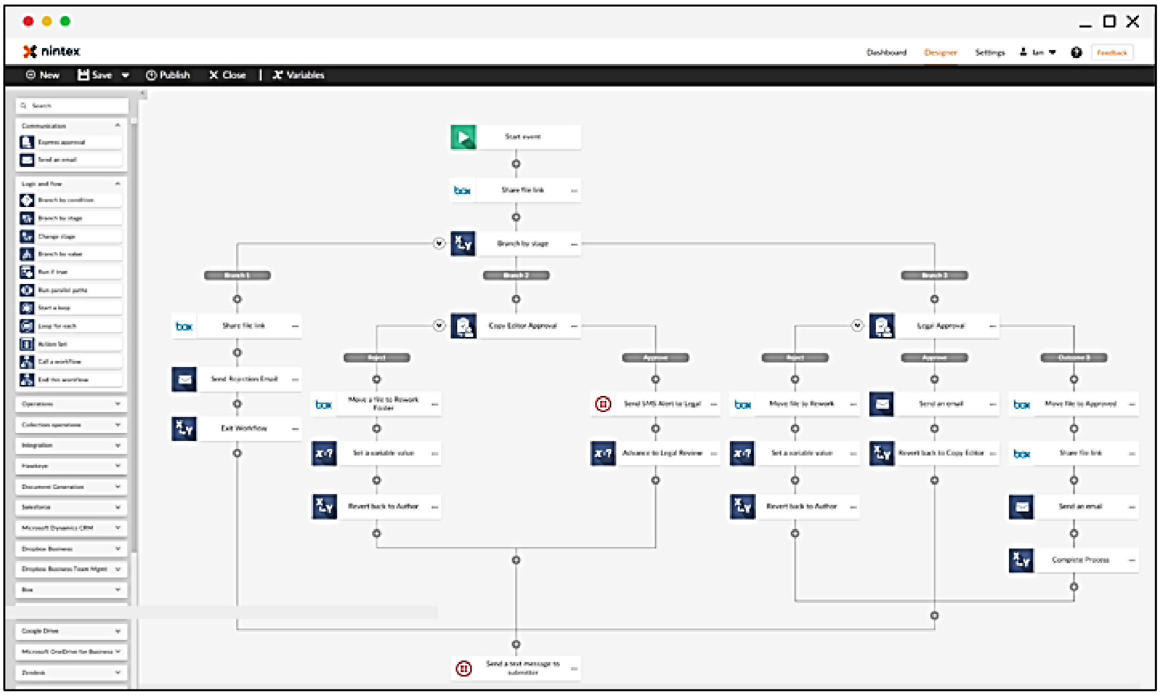


Figure 60: Workflows by Nintex®

Nintex is a process management platform and offers a list of features in this area. The most important of them can be summarized as follows:

- virtual capturing and storing of the information
- visualising of repetitive tasks with workflows
- document management
- integrating of existing tools via built-in connectors

Nintex is not specialised in one industry, like for example, logistics and can be used as a platform in a wide range of areas.

6.3.3. Workflow visualization kissflow®

This solution is intended for collaboration in creative teams. With the help of the kissflow[71] platform workflows can be visualized for any collaboration processes taking place in creative teams. The software environment is shown in the Figure 61.

[71] https://kissflow.com/offers

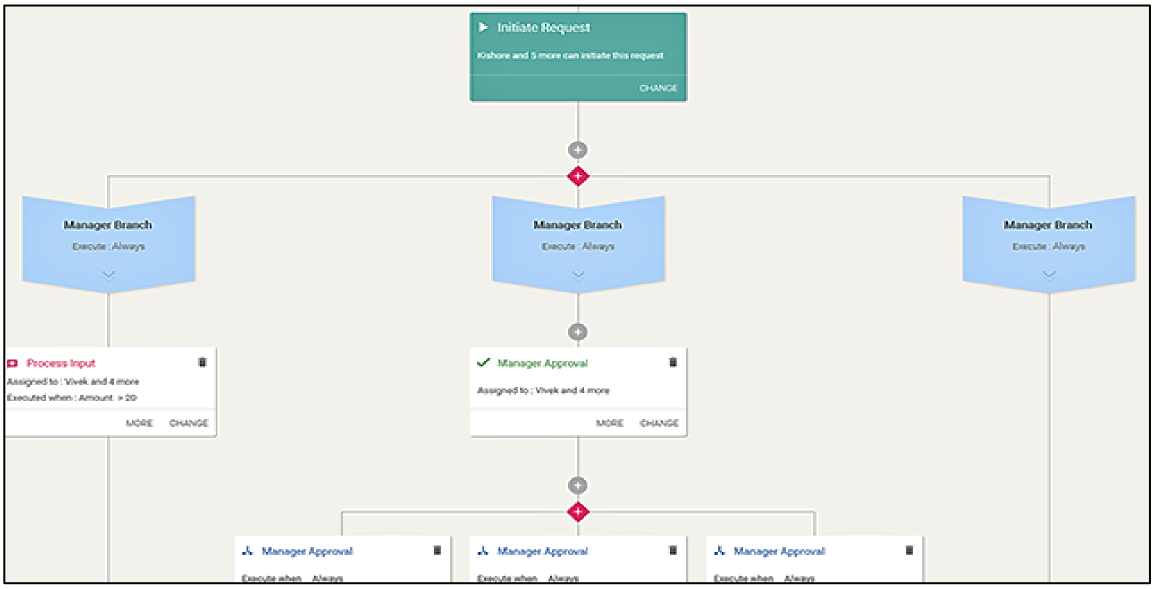


Figure 61: Workflows by kissflow[72]®

Like other mentioned solutions, the kissflow platform enables connecting processes via workflows and supports following features:

- Process management by creating visual workflows
- Project management by integrating team members
- Collaboration of departments and their members

This solution is aligned to the creative industry and focuses mainly on project and team management.

6.3.4. Web-based solution for workflows visualization Panflow®

Panflow (Figure 62) is another web-based solution that allows the visualization of workflows. By connecting processes and team relations document management and status overview can be exchanged.[73]

[72] https://gsuite.google.com/marketplace/

[73] https://www.panvision.de/de/panflow/

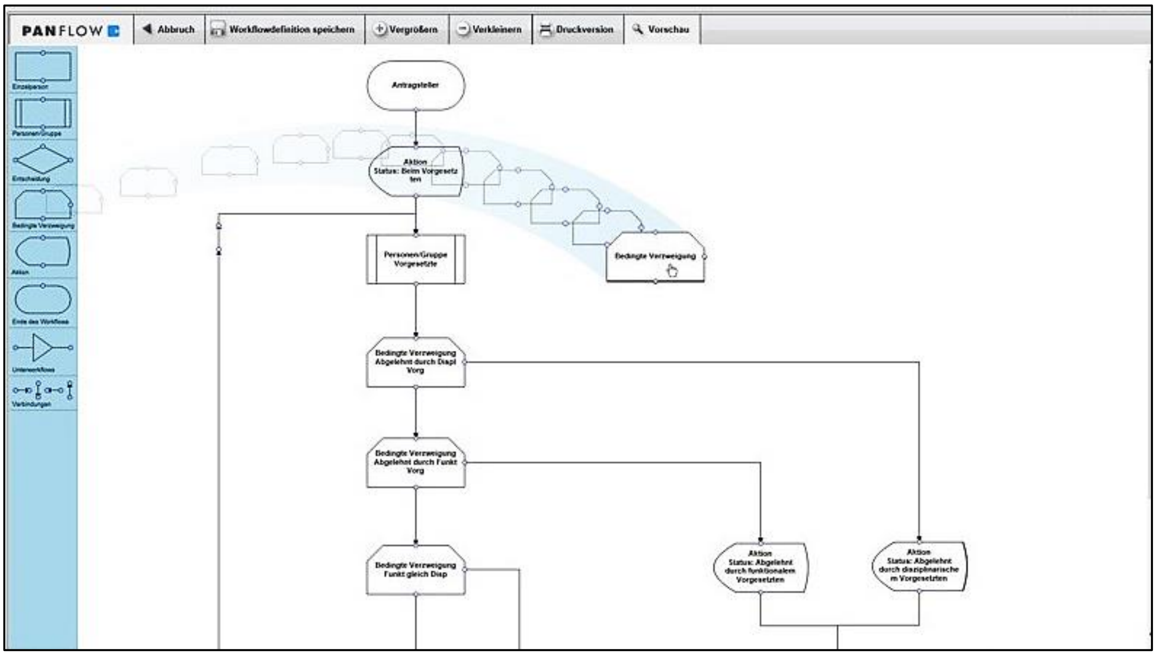


Figure 62: Workflows by Panflow [74]®

The main functionalities of this solution include among others:

- Management of individual forms
- Status overview
- Customisable overviews

A limited number of universal functionalities are offered in this software, so that they can be used industry-independent with the focus on document management.

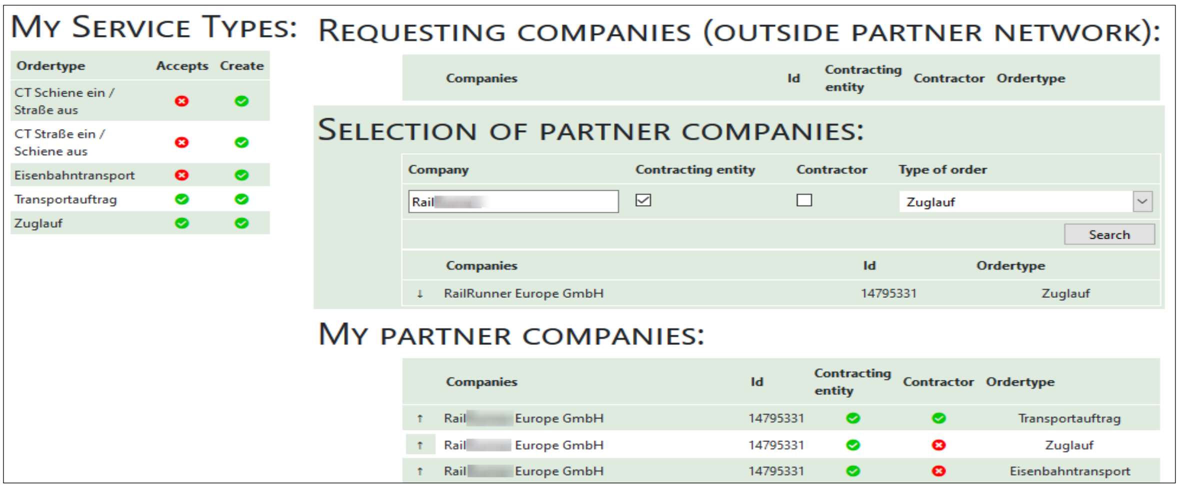
6.3.5. Operational workflows for data exchange and process organization with the logistics platform catkin®

The catkin platform is mainly focused on the custom development and collaboration among companies. It can be used for workflow management in any field but specializes on collaboration processes in (port) logistics.

This solution is a relational database system for creation and configuration of workflows for collaborative processes. As such, the processes can be described, which enable cooperation of teams across departments, companies, and countries. [75]

[74] https://www.softguide.de/programm/panflow-business-process-management

[75] https://www.catkin.eu/en/ueber-uns/



My Service Types:

Ordertype	Accepts	Create
CT Schiene ein / Straße aus	✗	✓
CT Straße ein / Schiene aus	✗	✓
Eisenbahntransport	✗	✓
Transportauftrag	✓	✓
Zuglauf	✓	✓

Requesting companies (outside partner network):

Companies	Id	Contracting entity	Contractor	Ordertype

Selection of partner companies:

Company	Contracting entity	Contractor	Type of order
Rail[redacted]	☑	□	Zuglauf

Search

	Companies	Id	Ordertype
↓	RailRunner Europe GmbH	14795331	Zuglauf

My partner companies:

	Companies	Id	Contracting entity	Contractor	Ordertype
↑	Rail[redacted] Europe GmbH	14795331	✓	✓	Transportauftrag
↑	Rail[redacted] Europe GmbH	14795331	✓	✗	Zuglauf
↑	Rail[redacted] Europe GmbH	14795331	✓	✗	Eisenbahntransport

Figure 63: Customized Operational Workflows by catkin®

(Source: catkin GmbH)

The focus of this solution is not the visualisation of the workflows, but the collaboration of different members of an organisation or a supply chain (as shown in the Figure 63) and the information exchange among them in the form of order processing.

For example, an order processing can consist of several information, such as:

- Order data, such as instructions or confirmations
- Status information (order confirmed, in process, finished, …) (Figure 64)
- Integration of resources (persons and devices)
- GPS
- Location
- Time

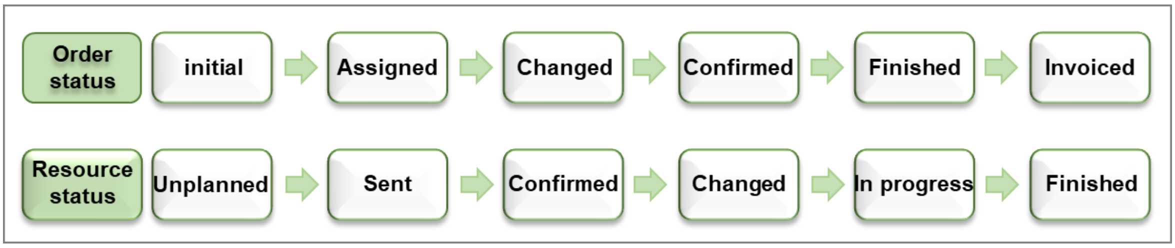


Figure 64: An example of status information: an order and a resource

The data and processes, mentioned above, build the main structure of operational workflows and can be linked with each other within orders. The actors involved in the supply chain get the order data from other actors and devices securely and according to their access role.

In summary, catkin is adapted to logistic processes and can link companies with their service providers, partners, mobile resources, and other supply chain participants. In addition to this, a flexible API can easily connect in.house-solutions with catkin, so the exchange of diverse data is supported.

6.4. Comparison and conclusion

The solutions mentioned above can be characterized as the visualization of process and workflow management and are used in a wide range of areas. The use of these solutions enables easy documentation and project management. The offered interfaces, additionally, provide a simple connection to existing software. In the following Table 5, the described solutions are listed and compared.

Table 5: Comparison of Workflow Management Solutions (Source: catkin GmbH)

	Mendix	Nintex	kissflow	Panflow	catkin

Visualized workflows	x	x	x	x	
Cloud-based	x	x	x	x	x
Process/ resource state	x	x		x	x
APIs	x	x		x	x
Document management	x	x			x
Collaboration intern	x	x	x	x	x
Collaboration extern logistics					x

Table 5 gives an overview of the solutions and important features, mentioned above. It is apparent, that all providers offer the possibility to support the designing of modular internal and external networks, containing workflows of customers and suppliers. These workflows can contain any relevant information and, additionally, the states not only of the processes but, also, of the resources.

However, some solutions have their focus on a specific industry, such as logistics, for example. This allows the creation of specific workflows and relations that are characteristic for that one industry. In addition, if there is any information collected by the IoT devices, it can be used to enrich these workflows and, thus, make the processes more efficient.

Especially in the I2PANEMA project, which is focused on improving the efficiency of logistics processes in ports and the use of IoT devices, these two segments - workflow management systems for logistic processes and IoT data - can be well combined as a part of the global architecture.

As can be seen from the market overview presented above, and, from the tabular comparison, it is not possible yet, to say, how the use of such workflow management systems can affect processes in the port environment. In this way, these benefits can be evaluated in the I2PANEMA project, and, demonstrated then, using proof of concept and demonstrators.

7. IoT visualization

7.1. Introduction

Data visualization allows the analysis and presentation of heterogeneous data in abstract form using computer graphics and interactive technologies. Data collection methods have evolved from traditional wired transmission to open wireless technologies such as RFID (Radio Frequency Identification) labels and embedded sensor and actuator nodes. Driven by the recent development of a variety of enabling wireless technologies, the IoT is the technology that will create the fully integrated internet of the future.

One of the most noticeable trends of the past decade is the large amount of data of all sorts, we need to work with and make sense of. From car telemetry to grocery chains' sales statistics, enormous heaps of numbers and figures are gathered nowadays. Obviously, an important task is to make sense of the numbers, monitor them and make decisions accordingly and in time. Therefore, Comprehensive and straight-to-the-point visualization of data is a must for any IoT appliance. The task becomes ever harder with the flow of data increasing.

Not only are the acquisition, storage, analysis and implementation of IoT data rather costly, but also businesses often lack tools and expertise to interpret and present the data effectively. Therefore, the Internet of Things' future and commercial adoption largely depend on data visualization technologies. Without data presentation tools, it would be practically impossible to interpret sensor readings coming from multiple data points distributed over a network, filter out insignificant entities and timely identify changes in smaller datasets (which might denote a performance issue or a marketing trend).

7.2. Data Visualization Techniques and Tools

The ever-growing volume of IoT data and its importance for analysis, make data visualization an essential part of many industries' business strategies.

In the following, we provide a detailed view on data visualization techniques and instruments, the factors that affect the choice of visualizations and a review of the most widely-used data visualization tools used in the literature.

7.2.1. What determines Data Visualization choices

Visualization is the first step to make sense of any type data including IoT data. To transcript and present data and data correlations in a simple way, data analysts use a wide range of techniques – charts, diagrams, maps, etc. Choosing the right technique and its setup is often the best way to make data understandable. And vice versa, wrong tactics may fail to present the full potential of data or even make it irrelevant.

Five factors that influence data visualization choices are as follows:

Audience. It is important to adjust data representation to the target audience. If it is end customers who browse through their progress in a fitness app, then simplicity is the key. On the other hand, if data insights are intended for researchers or experienced decision-makers, more complex charts are required.

Content. The type of data determines the tactics. For example, if it is metrics that changes over time, line charts are more suitable to show the dynamics. To show the relationship between two elements, a scatter plot can be used. In turn, bar charts are perfect for comparison analysis.

Context. Different approaches for the graphs and charts are used depending on the context. To emphasize a certain figure, for example serious profit growth compared to other years, shades of one colour can be used and the bright one can be selected for the most significant element on the chart. On the contrary, to differentiate elements, contrast colours is preferable.

Dynamics. There are various types of data, and each of it implies a different rate of change. For example, financial results can be measured monthly or yearly, while time series and tracking data is constantly changing. Depending on the type of change, dynamic representation (steaming) or a static visualization can be considered.

Purpose. The goal of data visualization also has serious influence on the way it is implemented. In order to make a complex analysis of a system or combine different types of data for a more profound view, visualizations are compiled into dashboards with controls and filters. However, dashboards are not necessary to show a single or occasional data insight.

7.2.2. Data Visualization Techniques

Depending on these five factors presented in the previous section, different data visualization techniques can be used.

Charts. The easiest way to show the development of one or several data sets is a chart. Charts vary from bar and line charts, that show the relationship between elements over time, to pie charts, that demonstrate the components or proportions between the elements of one whole.

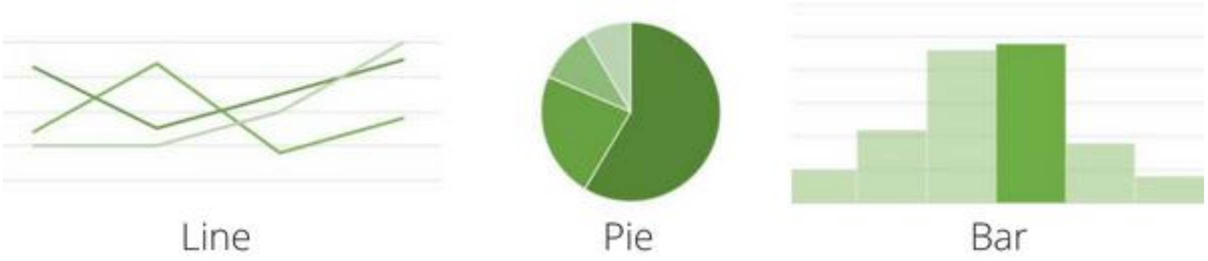


Figure 65: Charts

TPlots. Plots allow distributing two or more data sets over a 2D or even 3D space to show the relationship between these sets and the parameters on the plot. Plots also vary: scatter and bubble plots are the most traditional. Though when it comes to big data, analysts use box plots that allow visualizing the relationship between large volumes of different data.

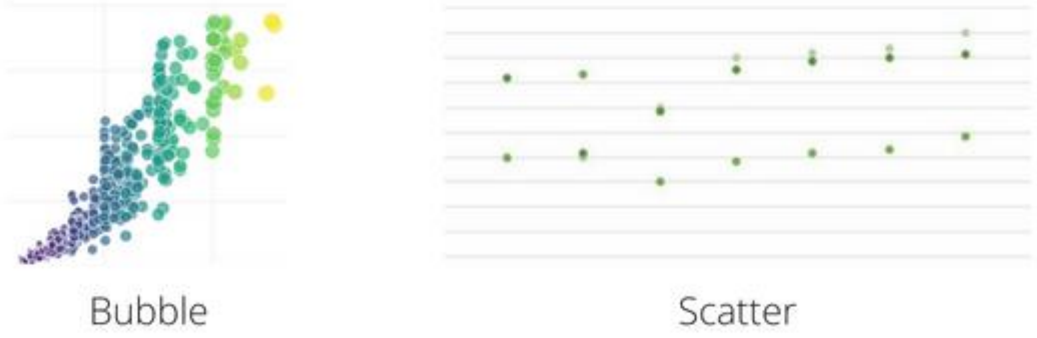


Figure 66: TPlots

Maps. Maps are widely used in different industries. They allow to position elements on relevant objects and areas – geographical maps, building plans, website layouts, etc. Among the most popular map visualizations are heat maps, dot distribution maps, and cartograms.

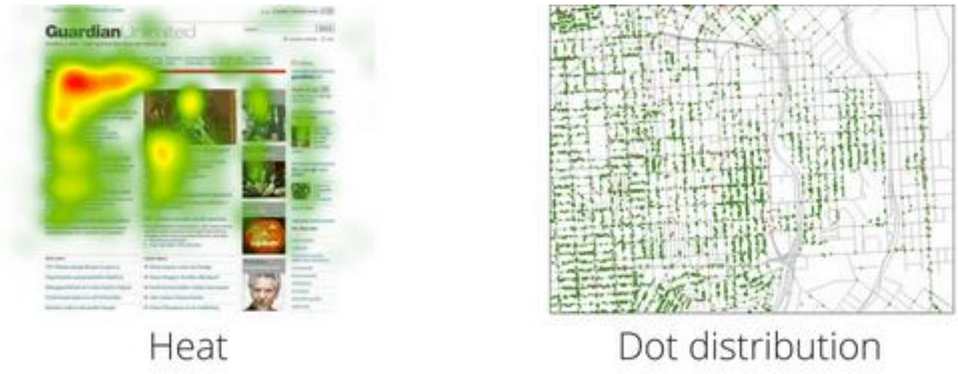


Figure 67: Maps

Diagrams and matrices. Diagrams are usually used to demonstrate complex data relationships and links and include various types of data in one visualization. They can be hierarchical, multidimensional, tree-like, etc. Matrix is a big data visualization technique that allows reflecting the correlations between multiple constantly updating (steaming) data sets.

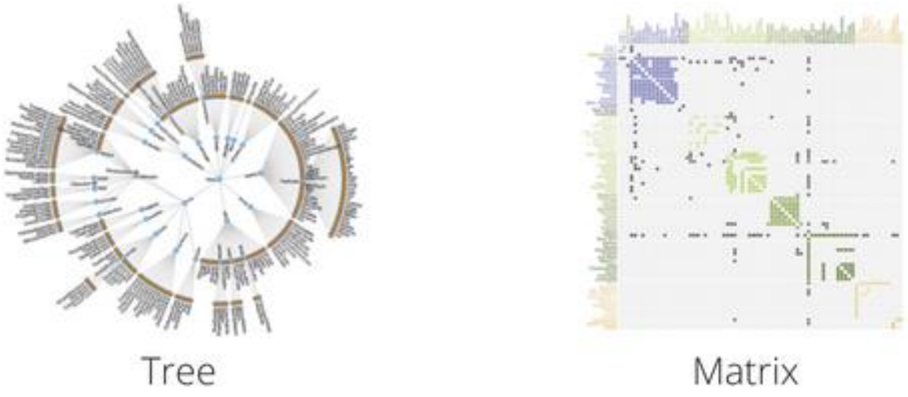


Figure 68: Diagrams and matrices

7.2.3. Data Visualization Tools

Together with the demand for data visualization and analysis, the tools and solutions in this area has developed fast and extensively. Novel 3D visualizations, immersive experiences and shared VR offices are getting common alongside traditional web and desktop interfaces. There are three categories of data visualization tools for different types of users and purposes.

1. **Data Visualization Tools For Everyone**

Tableau is one of the leaders in this field. Both newbies and professional analytics companies like Statista rely on this platform to create stories and derive meaning from their data.

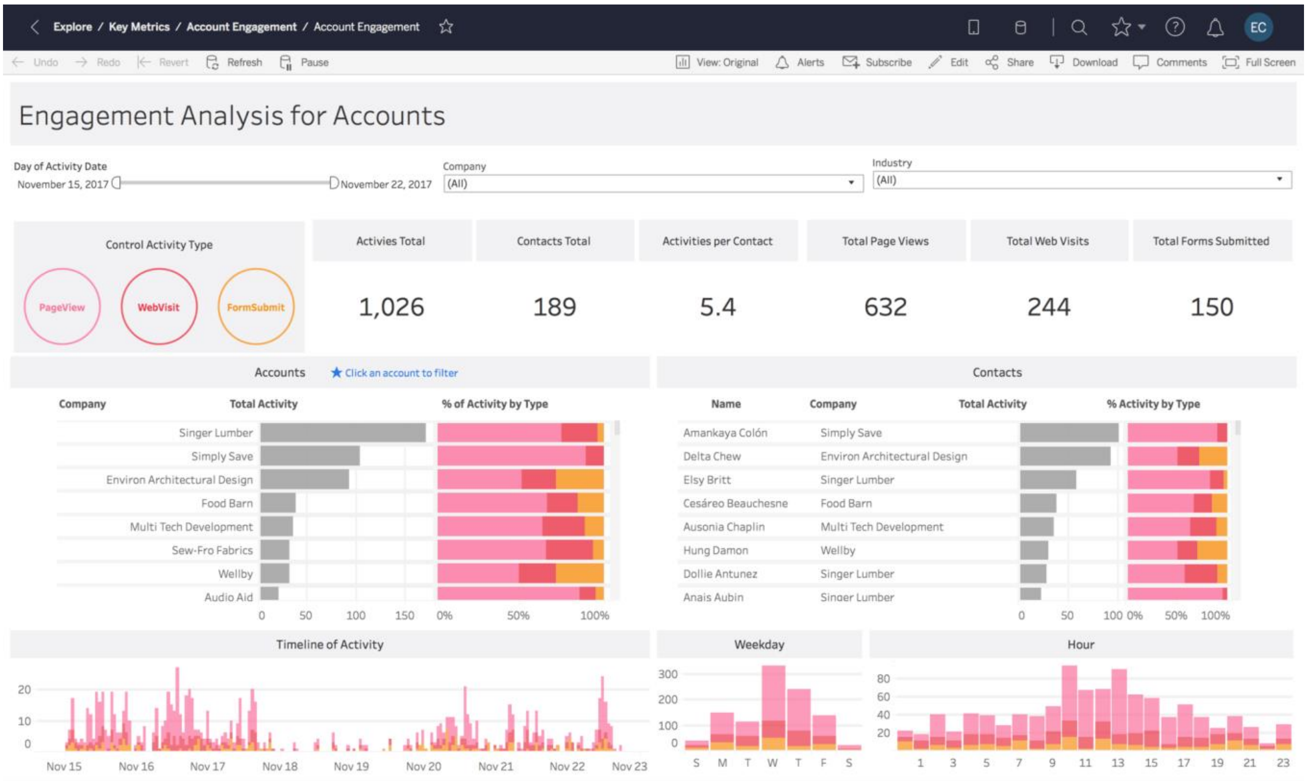


Figure 69: Tableau

The platform provides many integration options including MySQL, Teradata, Hadoop and Amazon Web Services. Therefore, it is a good tool for both occasional data visualizations and professional data analytics. The system can handle various types of data, including streaming big data and machine learning insights, and allows combining visualizations into smart dashboards.

Other Useful Tools for Everyone. Among other popular data visualization tools in this category are easy-to-learn **Visme**, **Fusioncharts** with varied integration capabilities, free and open source **Datawrapper** and **ZingChart** for JavaScript and HTML5 charts.

2. Data Visualization Tools for Coders

This category of tools includes more sophisticated platforms that enable not only data visualization, but also a certain level of data analytics.

Plotly is one of the most popular ones in this category. It is more complex than Tableau, however comes with analytics benefits. This visualization tool allows creating charts using R or Python, building custom data analytics web apps with Python, and even using and collaborating in rich open-source libraries for R, Python and JavaScript.

Sisense is another data visualization tool with full-stack analytics capabilities. This cloud-based platform has a drag-and-drop interface, can handle multiple data sources, and supports natural language queries. It seems that it is not good for beginners.

IBM Watson Analytics is known for its NLP capabilities. The platform enables conversational data control alongside strong dashboard building and data reporting tools. However, IBM Watson Analytics is not cheap and works best for serious data visualization and analytics tasks.

3. Tools for complex data visualization and analytics

Growing importance of connected technology, available data sources and fast-changing environments make companies deal with different types of multi-source data and search for more complex visualization and analytics solutions. This category includes three of them: Microsoft Azure Power BI, ELK stack Kibana and Grafana.

Power BI The Microsoft Power BI solution is designed to trace and visualize data gathered by a variety of sensors including temperature, sound, motion and location tags, as well as healthcare sensors, and promptly identify trends across IoT device installed base. The platform works in sync with Azure cloud-based analytics and cognitive services, and it has a drag and drop interface, short learning curve and large integration capabilities.

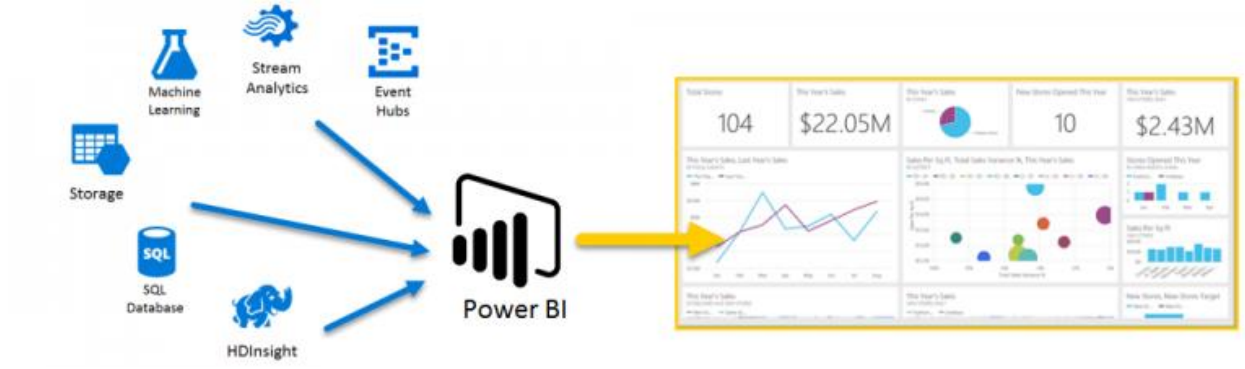
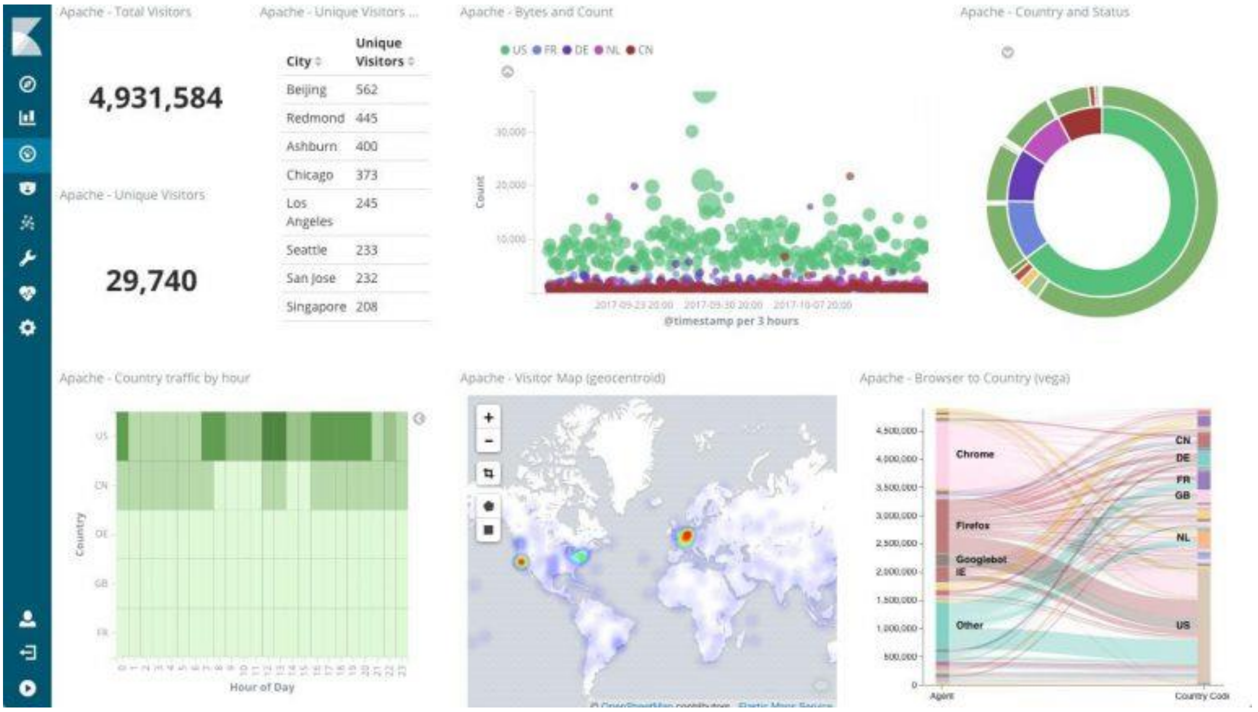


Figure 70: Power BI

With the integrated Azure services, Power BI today is one of the mostly used data visualization and analytics tool that can handle large amount of data.

The platform allows creating customized reports from different data sources. Furthermore, it also allows processing streaming real-time data and therefore it is suitable for streaming data analytics. In addition to the Azure and other Microsoft services, it can also connect to existing applications and drive analytics to custom systems.

Kibana is the graphical part of the Elastic Stack. It is built on and designed to work on Elasticsearch data only. However, this constraint does not prevent Kibana from being an effective data visualization tool for log data.



Apache - Total Visitors

4,931,584

Apache - Unique Visitors

29,740

Apache - Unique Visitors ...

City	Unique Visitors
Beijing	562
Redmond	445
Ashburn	400
Chicago	373
Los Angeles	245
Seattle	233
San Jose	232
Singapore	208

Figure 71: Dashboard of Kibana

The tool supports most of data visualization techniques – interactive charts, maps, histograms, etc. Moreover, Kibana goes beyond traditional dashboards for data visualization and analytics. The system allows building advanced analytics: combining visualizations from multiple sources to explore correlations between different insights and turn on machine learning features to reveal hidden relationships between data events. It has also a set of tools and APIs for developers.

Grafana is a data visualization and analytic tool that supports nearly 30 data sources, including AWS, Elasticsearch and Prometheus.

Even though Grafana is more flexible in terms of integrations compared to Kibana, each of the systems works best with its own type of data. In case of Grafana, it’s metrics. Therefore, it makes this tool popular among IoT data visualization solutions.

Grafana allows visualizing and compiling into complex dynamic dashboards different types of metrics data. It has a wide variation of admin roles which makes it suitable for complex monitoring and control systems.

Additionally, it enables alerts and notifications based on predefined rules. Finally, Grafana provides a set of benefits for fast data analytics, such as creating custom filters and making annotations – adding metadata to certain events on a dashboard.

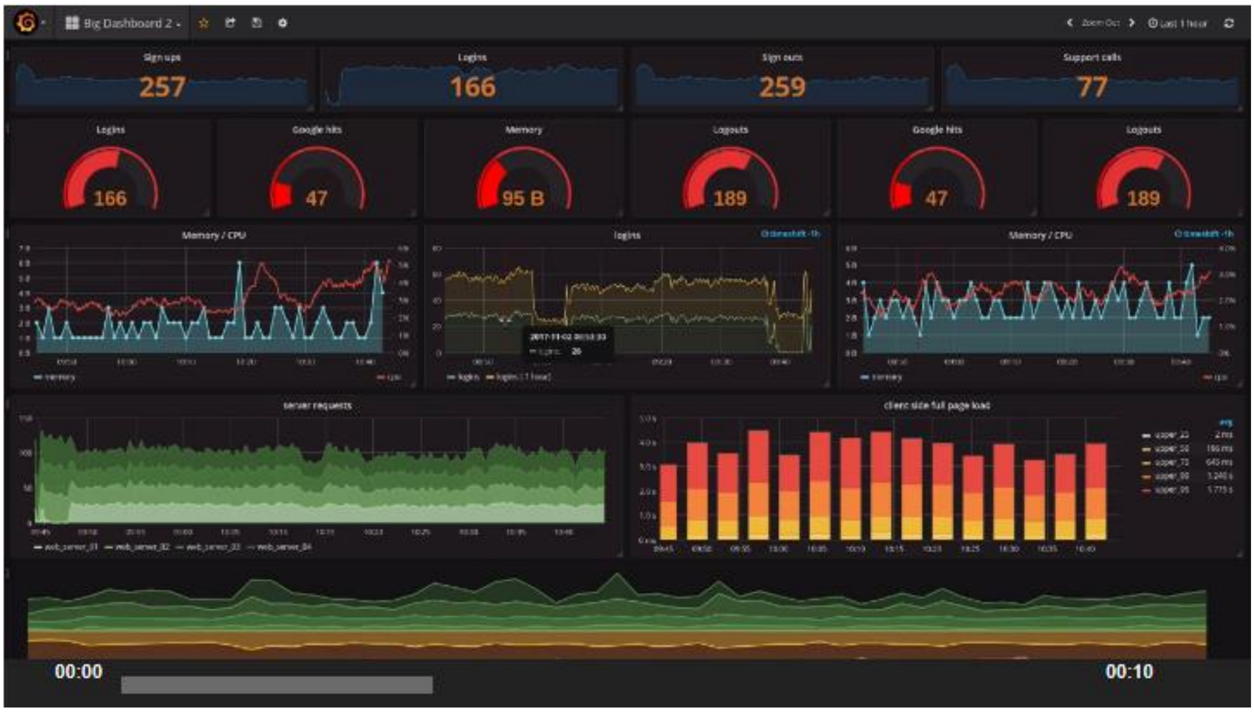


Figure 72: Dashboard of Grafana

ThingSpeak. Launched by the creators of Matlab, this is a flexible open source platform and API that allows developers and application designers to gather data from sensors and other sources, analyse, and visualize it.

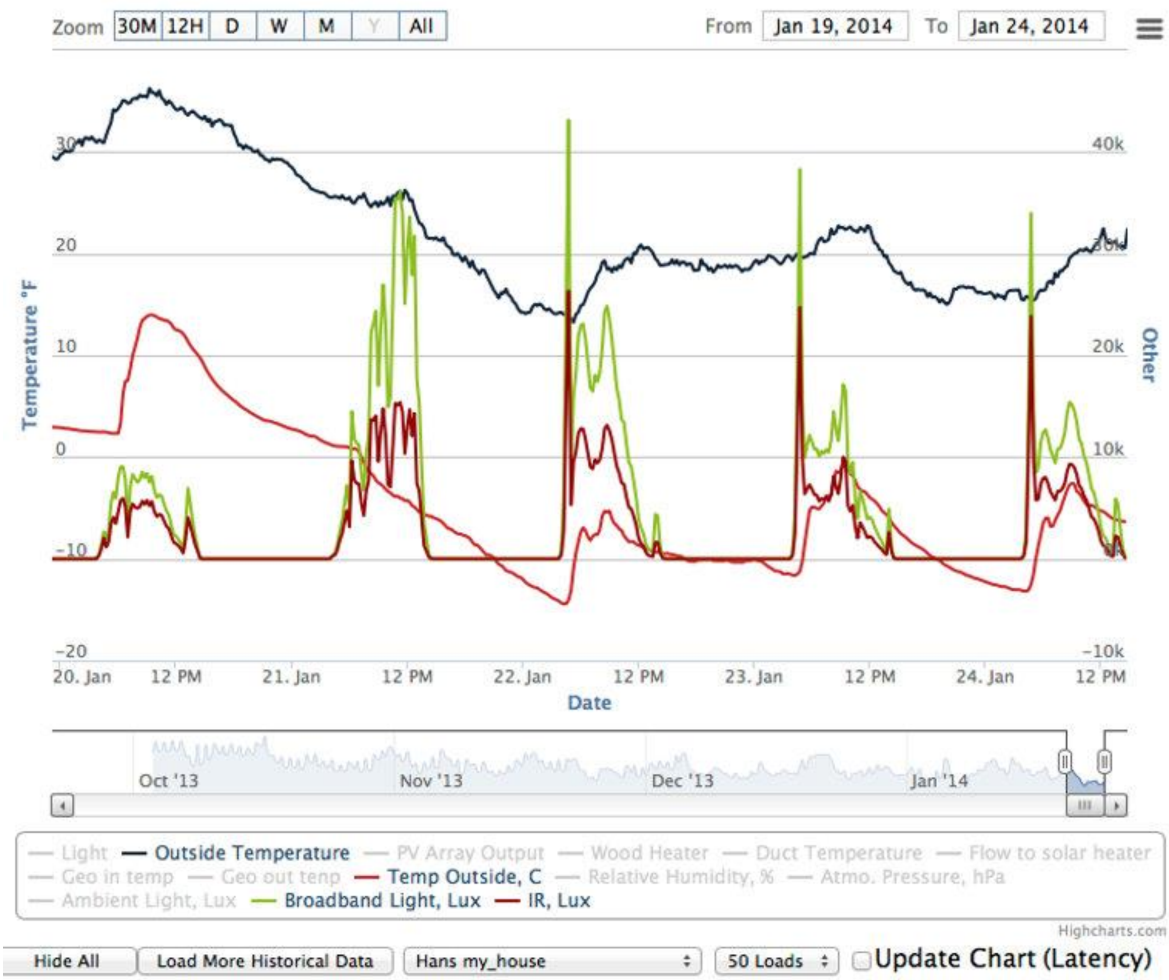


Figure 73: ThingSpeak's Dashboard

Freeboard is a purpose-built visualization tool for the IoT. It allows to create a dashboard full of different widgets and share it with other people. To collect data, Freeboard offers integration with the Dweet.io IoT messaging system but can also communicate with a web-based API. Widgets, which are the main building block on the platform, can be either created from scratch or chosen from a list of predefined ones. The platform is open source but not entirely free.

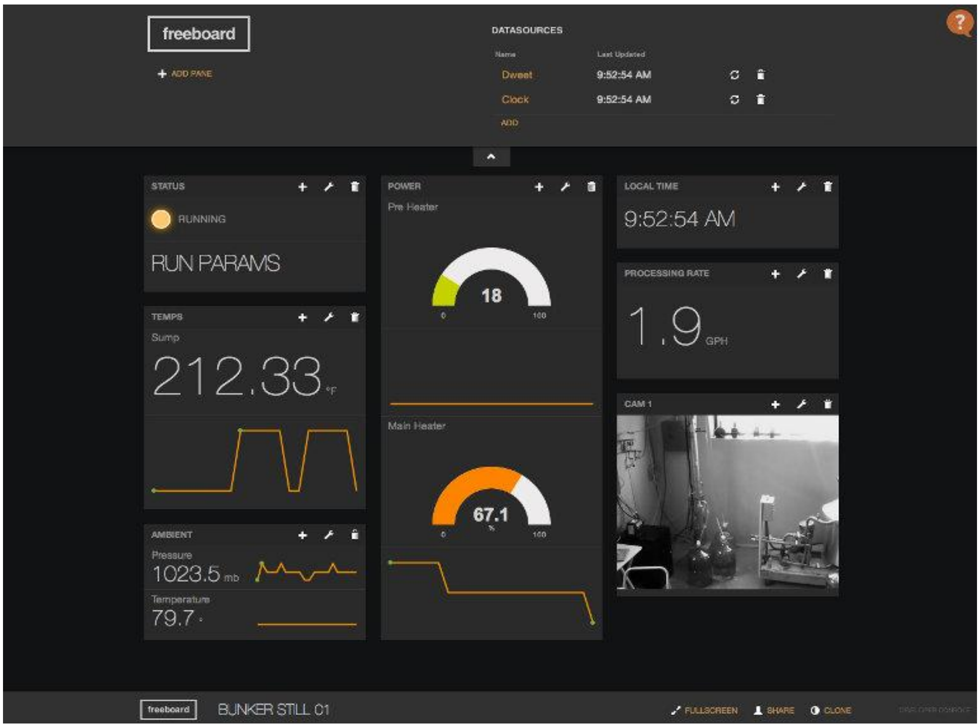


Figure 74: Freeboard's dashboard

IBM Bluemix. The centrepiece of the IBM Internet of Things Foundation works as a hub, to which sensors and other data sources can be connected. The data is sent via the MQTT lightweight messaging protocol. Being a universal hub for Watson-powered IoT solutions, the platform features built-in web applications and supports the integration with 3rd-party software via REST APIs. The service helps businesses create customizable minimalistic diagrams, graphs and tables displaying static and dynamic data, access device properties data and set up alerts.

A visualization example from Sogeti with an explanation is presented:

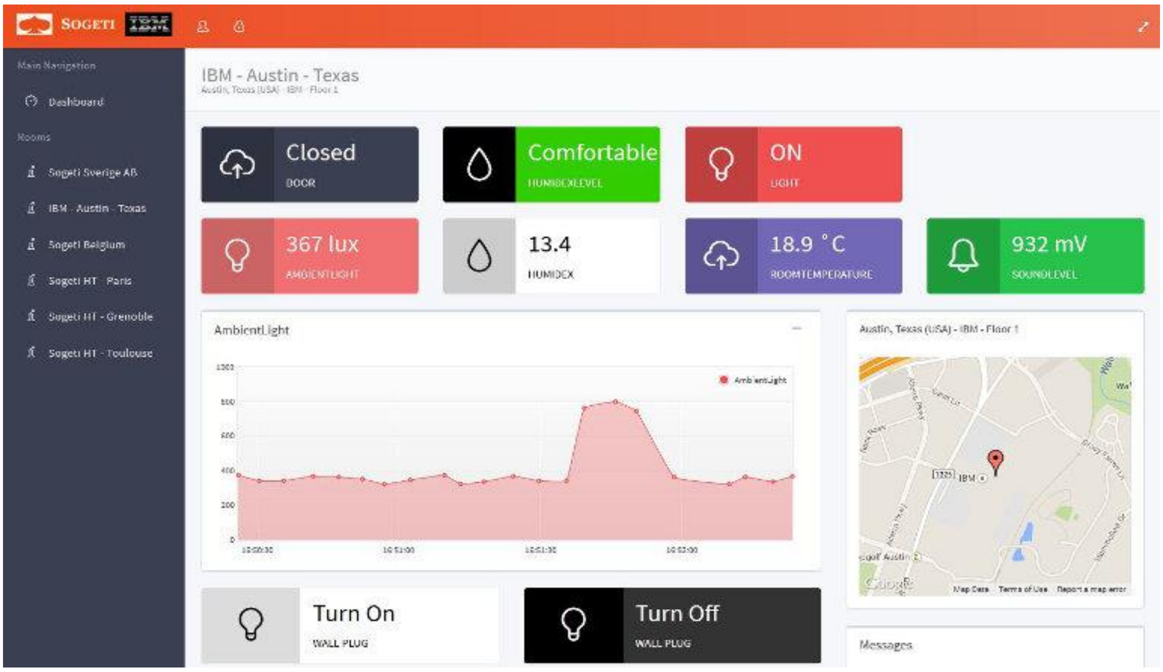


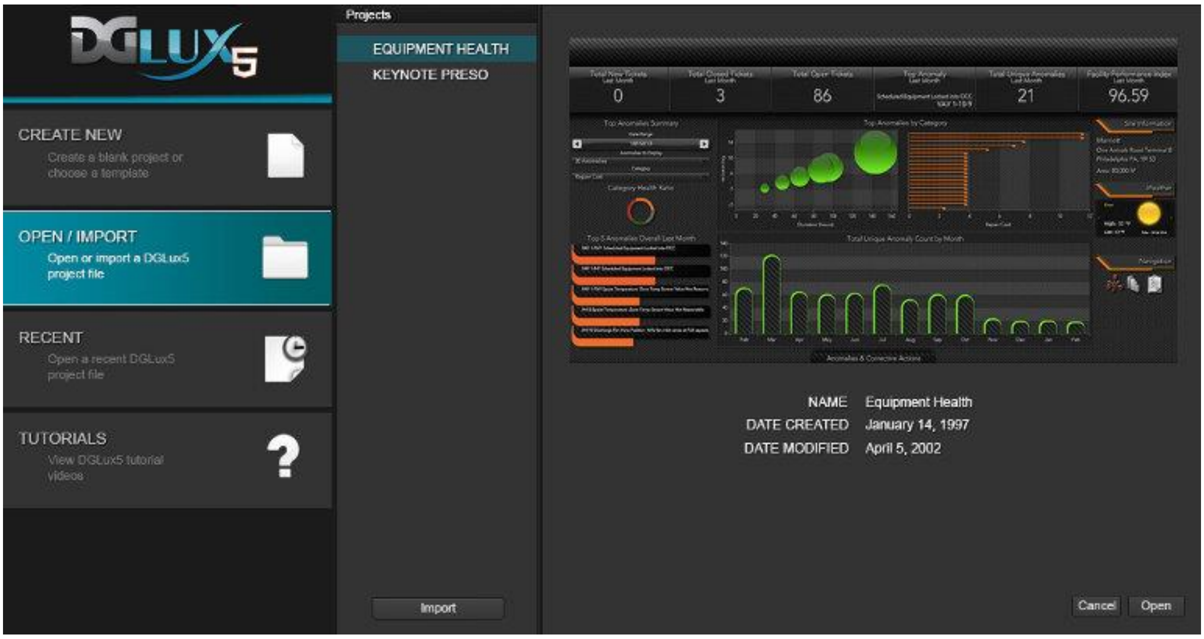
Figure 75: Dashboard of IBM Bluemix

"Data and events are forwarded as they occur by each gateway to IBM's IoT Cloud by sending JSON over the MQTT Protocol. A Bluemix application processes incoming messages based on a flow that can be configured separately per room. Some of the data is just captured and stored in

> *the historian. Some of the data is consumed by the live user interface. Some of the data can trigger alerts."*

IBM offers a flexible pricing system, where users could be paying depending on the number of instances and the amount of memory used.

DGLux. The IoT application platform by DGLogik provides a number of assets like animated widgets, background themes, patterns, effects and so on. The programming of the dashboard is also visual, which helps those less familiar with coding.


DGLUX5
Projects
EQUIPMENT HEALTH
KEYNOTE PRESO
CREATE NEW
Create a blank project or choose a template
OPEN / IMPORT
Open or import a DGLux5 project file
RECENT
Open a recent DGLux5 project file
TUTORIALS
View DGLux5 tutorial videos
Import
Cancel
Open

NAME Equipment Health
DATE CREATED January 14, 1997
DATE MODIFIED April 5, 2002

Figure 76: DGLux's Dashboard

Ripples IOT. For industrial condition monitoring, this IoT Platform consists of retrofit, wireless sensors, edge gateway, device management tools, and on-premises data visualization software with alerts and notifications. With this platform, it is possible to know about abnormal temperature levels, excess vibration in standard equipment, tracking of moving assets and inventory.

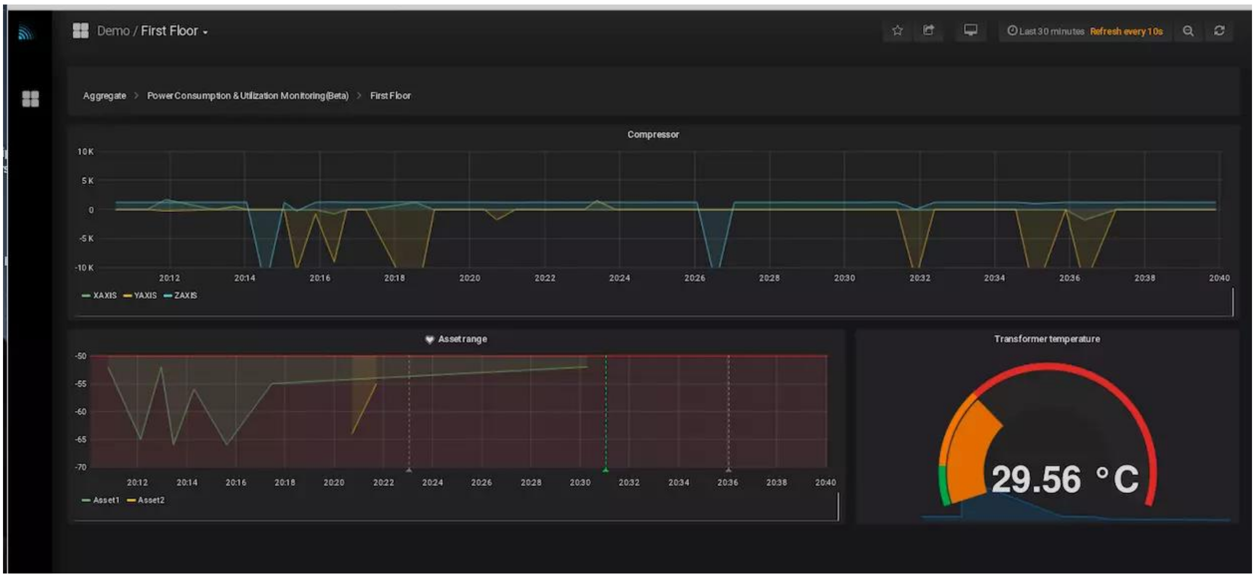
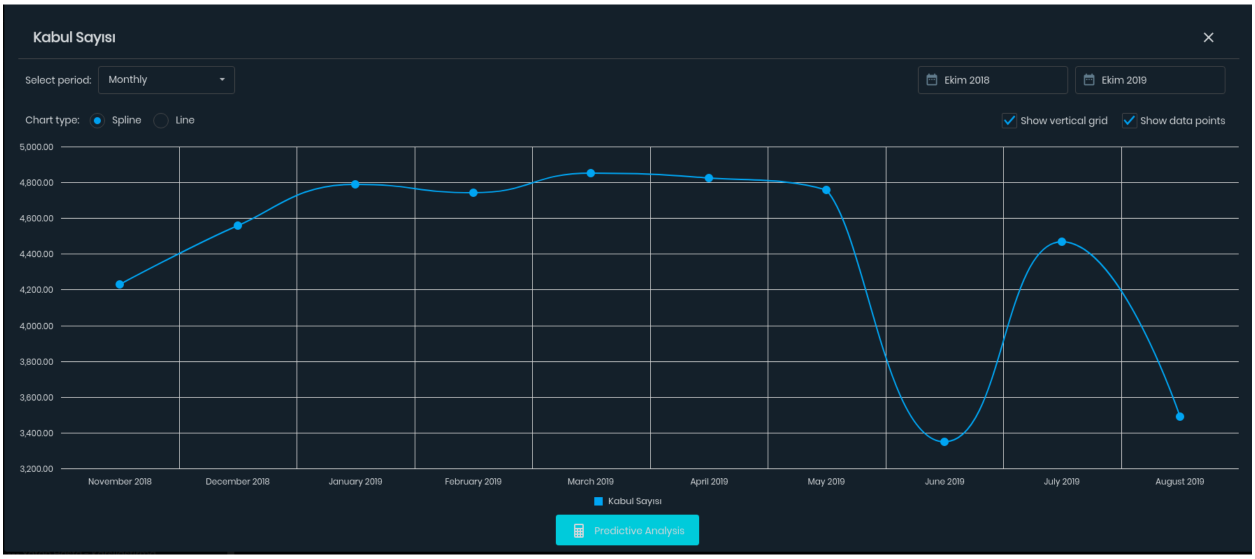


Figure 77: Ripples IoT Dashboard

MantIQ. The business intelligence tool of SRDC Corp. provides multiple visualization methods including map, charts and location maps. The tool allows users to define their Key Performance Indicators (KPI) and their Dimensions and create their own dashboards for these KPIs.

MantIQ supports continuous data flow. For this purpose it stores the data as time series data. The tool also has goal/notification service and predictive analysis tools.



Kabul Sayısı

Select period: Monthly

Ekim 2018 — Ekim 2019

Chart type: ● Spline ○ Line

☑ Show vertical grid ☑ Show data points

Figure 78: MantIQ Dashboard

8. IoT Security and Privacy

This layer will be a vertical layer to ensure adequate and controlled access to the platform’s data and operations. Much of the data collected by sensors or port management systems require restricted access (e.g. information about the goods a ship carries may be confidential for business reasons). The proliferation of “Connected Things” driven by the IoT has prompted the need for “Secure Things” that are critical in preventing the sophisticated system attacks. Given the security vulnerabilities of IoT Use Cases, the UC owners are primarily concerned about the following:

- What are the safest mechanisms to implement a secure data exchange among several systems?
- What is the most appropriate technology for identity management inside the company and the value chain?

8.1. Technological Choices

Before discussing the technological choices for a secure implementation of IoT systems, it is important to understand what security means in the context of IoT.

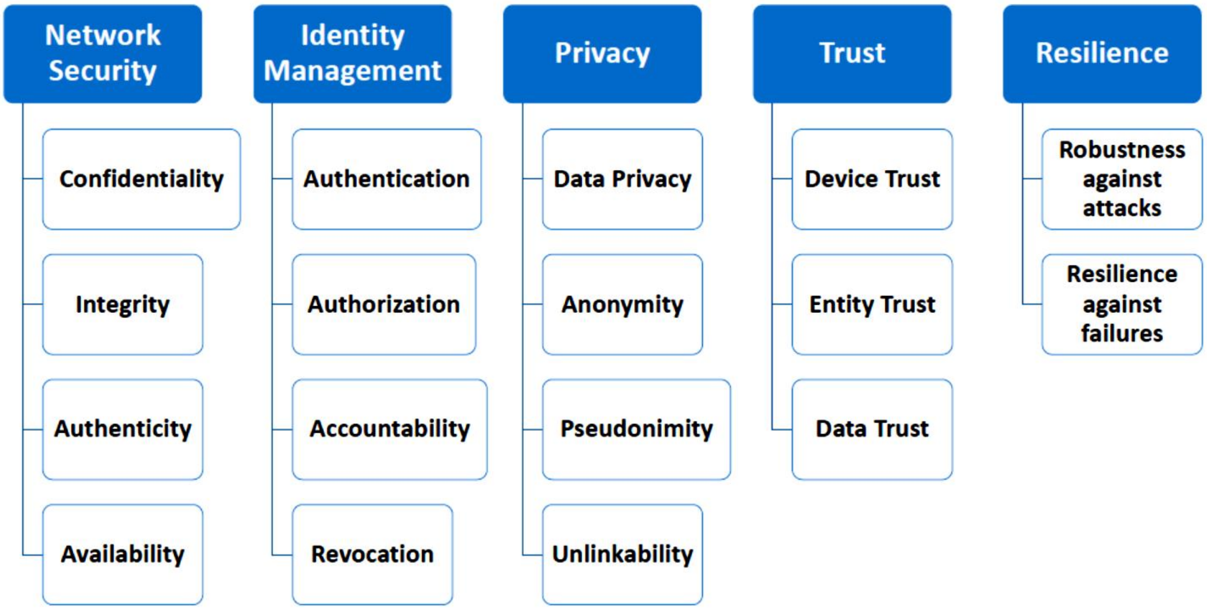


Figure 79: Security, Privacy & Trust aspects for secure IoT solution

The figure above briefly describes the important aspects that need to be addressed while designing a secure IoT solution. Security must be integral part of the system design from the start (i.e. security by design). Typical challenges encountered for designing a secure IoT solution are

- **Constraint resources**: IoT modules are usually low power and have less computational resources (RAM/ROM) etc. Therefore security countermeasures have to fit into these small footprints on top of the required functionalities.
- **Uncontrolled environment**: Typically IoT devices have to be 24x7 operational and their place of deployment are usually remote locations (not easy to access or even mobile or in open public area). This access to these locations may not be restricted by owners. The environment conditions can be harsh (cold, warm, humid, rainy etc.). Thus IoT devices have to be robust against these environmental factors.
- **Heterogeneous**: Usually IoT devices network or ecosystem is comprised of multiple vendors & solution providers. There is no uniform operation guidelines and interfaces may not be interoperable (often proprietary). Due to these heterogeneous device profiles, operation & maintenance for device owners is very difficult.
- **Scalability**: In addition to above mentioned challenges, scalability is another daunting aspect. Due to rapid change/improvement in technology, shorter time to market value, non-standard protocols & interfaces, IoT solutions are faced with shorter lifetimes. This means, customers cannot plan for a large scale deployment across multiple, wider geographical areas (due to the mentioned bottlenecks).

A typical security analysis of IoT solution encompasses detailed description of the Assets (i.e. what needs to be protected? What is its value), stakeholders (their roles & rights), possible threat scenarios (attack vectors & inherent vulnerabilities) followed by standard security countermeasures. For doing security analysis for IoT solution, Microsoft's STRIDE methodology is a very good option. NXPGE recommends the use cases in I2PANEMA to employ STRIDE methodology to do security analysis.

STRIDE is an abbreviation that stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privileges which are type of security threats described briefly in the below.

Table 6: STRIDE threat description

Threat	Security Property	Description
Spoofing	Authentication	Pretending to be something or someone other than yourself

Tampering	Integrity	Modifying something on disk, on a network or in a memory
Repudiation	Non-repudiation	Claiming that you didn’t do something, or where not responsible
Information disclosure	Confidentiality	Providing information to someone not authorized to see it: data leak or privacy breach
Denial of service	Availability	Absorbing resources needed to provide service
Elevation of privilege	Authorization	Allowing someone to do something they’re not authorized to do

STRIDE Analysis is a threat classification model developed by Microsoft which can be used in security analysis of any system which is briefly described below. More info about STRIDE Analysis can be found in the below links

1. https://en.wikipedia.org/wiki/STRIDE_%28security%29
2. http://resist.isti.cnr.it/free_slides/security/williams/RiskBasedSecurityTesting.pdf
3. http://www.cs.berkeley.edu/~daw/teaching/cs261-f12/hws/Introduction_to_Threat_Modeling.pdf

Details of this methodology and how to employ it on IoT use case shall be explained in Task 5.2 of Work Package 5.

8.2. Guidelines for Designing a Secure IoT Solution

While the STRIDE analysis provide Use Cases with an in-depth analysis of security threats, the results only become valuable if one addresses the findings accordingly and hence increases the overall security of the Use Case architecture. To do so, one must not only select the most secure hardware, it is also important to design effective processes and train the people that are responsible for implementing these processes.

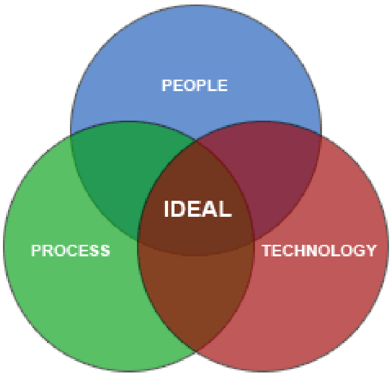


Figure 80: Information Security Management System

To secure your Use Case properly you need an *Information Security Management System* which addresses people, processes and technology in a single, cohesive package. If you have got the technology in place but you do not have proper processes and have not trained your staff properly then all your efforts to achieve a more secure system go in vain.

Below listed are the general best practices for IoT product development. Some of these ‘best practices’ can be used to reduce the danger of some of the threats.

Process:

- In case of non-anonymous data, a regulation on data management is implemented to guarantee the integrity and the safety of these data
- ‘Security through obscurity’ is not a good security doctrine

- From the beginning of the project, realize and keep updated a risk analysis of the product and define threat's mitigations (STRIDE analysis for IoT Use Cases)
- Choose and apply a cybersecurity standard
- The product has a secure process of software update
- No network protocol or encryption algorithm with known weakness is used (md5, sha1…)
- The source code is written according to coding rules (OWASP/CCERT/MISRA/internal…)
- The source code is verified by a third software (static and dynamic source code analysis tool) and should cover at least simple fault injection, simple logic attacks and side channel attacks
- All functionalities associated to cybersecurity are unitary tested
- The security bulletins of all third-party software and services used for the realization of the software are followed during the design and the product's life to react to the identification of a weakness (software provider website, specialized website: http://cve.mitre.org …)
- The firmware is verified by a series of antivirus before it's sent in production
- Every version of the product is validated by a Pen Test and/or fuzzing attack (authorized simulated attack)

People:

- Name a security manager who defines and guarantees best practice of design as well as the preservation of the product's security
- Do not design your own security mitigation technology, use standard & available ones. If you need a custom solution, ask experts.
- The people having access to the keys of encryption and to the firmware are clearly identified during the production

Technologies:

- All network protocols use an encryption (HTTPS, TLS, FTPS, SSH) and are protected against replay attacks (use of session keys)
- The secret keys used are unique for each product and stored on a protected memory.
- Strong password policies are enforced
- Activate a secure start/initialization of the product by secure booting/initialization (load of a trusted encrypted image of the software)
- Once the final software is ready to be released, all debug interfaces (i.e. JTAG etc.), all software debug functions are disabled or removed and the security fuses are blown to lock configuration.
- Some compilers embed security feature like the stack protection on ARM compiler, use these features after a careful read of the document
- The different data stores embed integrity and reverse engineering proof mechanism
- The unused network interfaces (bus + port) and unused services are deactivated
- On the final product's electronic board, the test points have no significant names
- The elements used for the security functionalities cannot be replaced during the final assembly
- The product can execute only numerically signed software (trusted source)
- During his run time, the software verifies periodically its integrity
- A secure element is used to store the encryption keys
- The software uses physical devices to encrypt/decrypt/hash data (easy to use and more difficult to tamper than software functions)
- An unique and not modifiable identifier is used to assure the identification of the product
- If an operating system is used, mechanisms of memory insulation and least privilege of execution are used (trust boundary between the different soft-ware's layers, 'onion design')

- If an operating system is used, with possibility to use privileged accounts, these accounts are only usable by the firmware, without possible remote access
- The default user password must be changed at the first use of the product
- If passwords must be stored, store the crypto hash values instead (passwords encryption) then compare hash values to control
- In case of failure of the password seizure, a delay must be respected before the next try (brute-force attack resistance)
- Data type, format, length, and range checks are enforced

In general, developers following security by design principles should ensure that below security properties are adequately addressed for the intended IoT solution they wish to create.

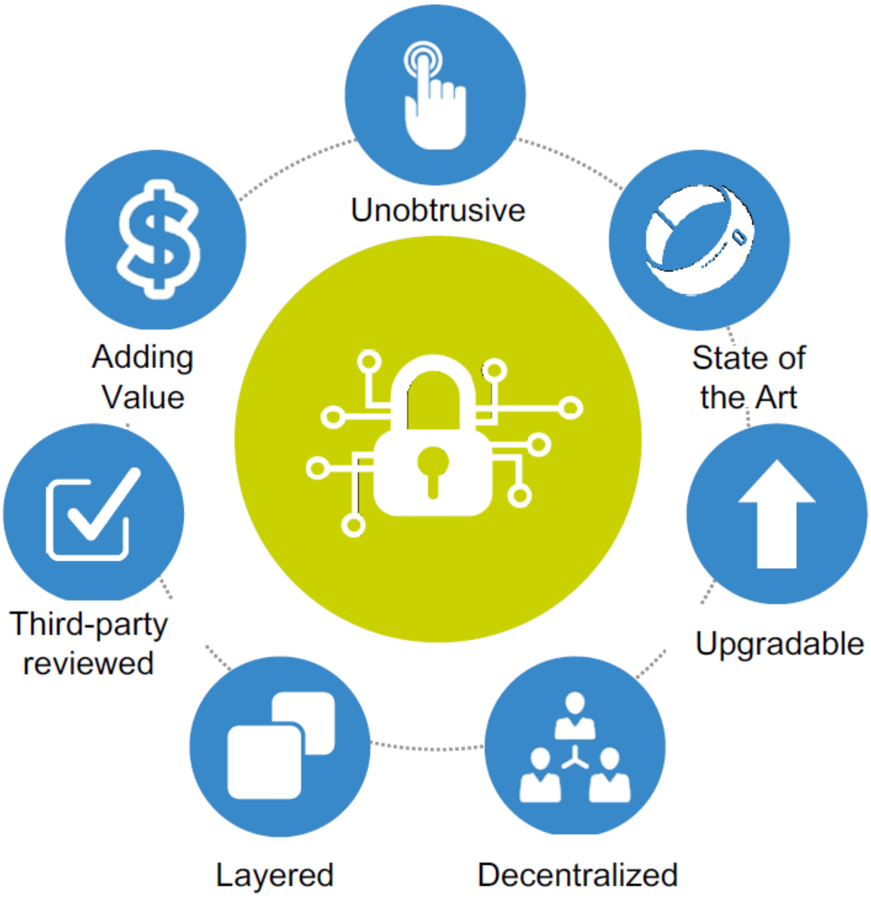


Figure 81: Security by Design principles

The standard mitigations against threats described in STRIDE model is shown below. It is important that well known and tested security policies are employed instead of inventing ad hoc security solutions.

Table 7: Standard mitigations for STRIDE threats

Threat	Security Property	Mitigation
Spoofing	Authentication	Challenge / Response schemes (PKIs like TLS & Certificates), Digital signatures, Messages Authentication Codes (MAC), Hashes
Tampering	Integrity	Access control lists (ACL), Digital signatures, CRC, MAC
Repudiation	Non-repudiation	Secure logging with timestamps, Digital signatures, trusted partners

Information disclosure	Confidentiality	Encryption, ACL
Denial of service	Availability	Quotas (dedicated slots), scheduling, Filters, Access control lists
Elevation of privilege	Authorization	Group / Role membership, privilege ownership, input validation, ACL, permissions

8.3. State of the Art for Security & Privacy

Unlike other domains, latest or cutting edge technologies doesn't always mean most secure security solutions. Radical innovations are inherently characteristic to be also least tested and often poses high risk due to undiscovered vulnerabilities. Typically security countermeasures are exhaustively tested over extended period of time and then certified by several international and national bodies before being recommended for mass deployment. As the security technologies advance, so does the hacking technologies and therefore it is always advisable for security solutions developers and integrators to deploy latest recommendations from international security certification bodies rather than inventing or building own/proprietary security solutions. Some of the well-recognized organizations are Bundesamt für Informationssicherheit or BSI from Germany, National Institute of Standards and Technology or NIST from USA, Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408, Federal Information Processing Standard or FIPS 140-2, ANSI/ISA IEC 62443 (Formerly ISA-99) Industrial Cyber security standards etc.

A typical IoT solution is always composed of one or more sensors, a processing unit and wired/wireless communication unit that sends/receives data to backend network. The backend composes of local/global servers and databases (may be running on cloud) and analyse current and past data and give feedback to device and end user.

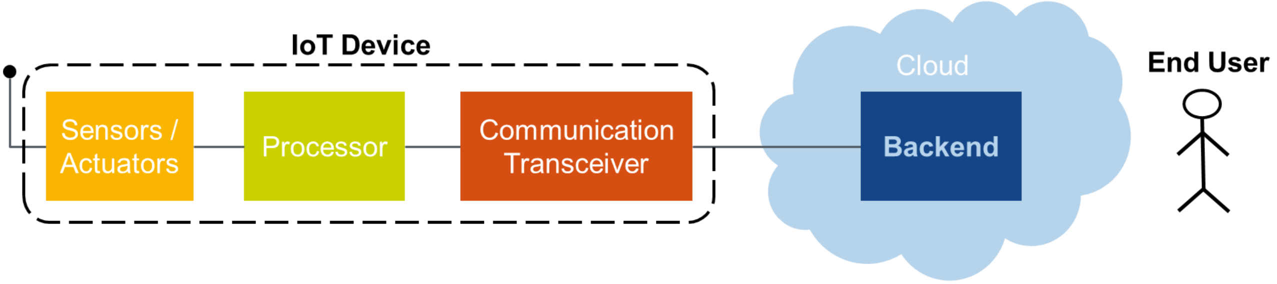


Figure 82: Model IoT system

Incorporating end to end security is vital for resilient IoT solution. Possible security countermeasure for every component of an IoT system is summarized in the below figure.

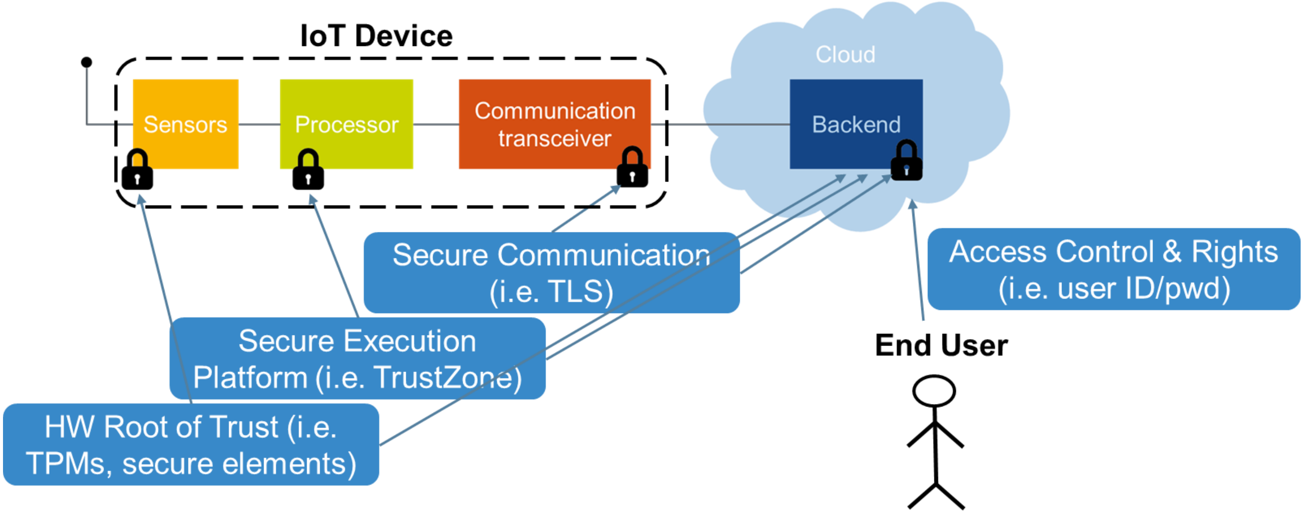

IoT Device
Sensors
Processor
Communication transceiver
Cloud
Backend
Secure Communication (i.e. TLS)
Access Control & Rights (i.e. user ID/pwd)
Secure Execution Platform (i.e. TrustZone)
End User
HW Root of Trust (i.e. TPMs, secure elements)

Figure 83: Secure IoT system

Some of the state of the art security solutions[76] in accordance to Figure 83and a brief description of security property they address are mentioned below.

- **Access Control:** These security products uniquely identify the each user or an entity in an IoT network and also define their access rights and privileges. Typically RFID/NFC & 2FA solution are employed for access control. Some examples are NXP MIFARE solutions

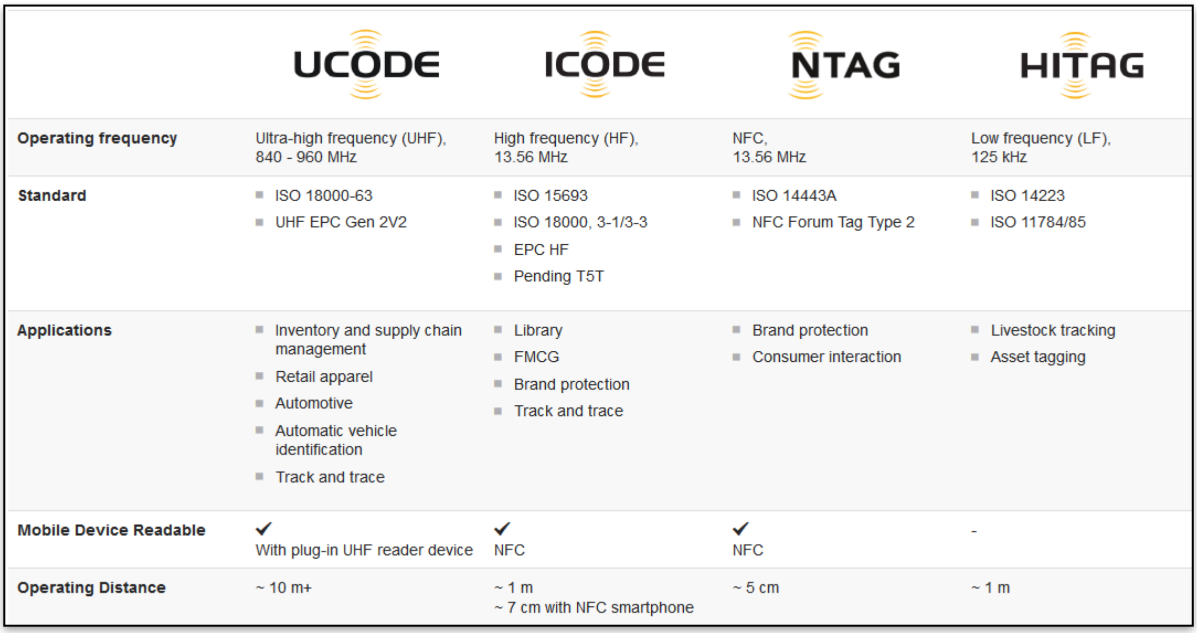
Figure 84: Family of secure MIFARE microchips used in contactless smart cards and NFC devices[77]

  - Contact/Contactless Readers & Smartcards
  - Proven, secure, scalable & most widely deployed solution worldwide
  - Certified security standards

- **Asset identification**: these solutions also uniquely identify the person or an entity similar to access control solution. However they are low cost & have limited security features. Typically RFID, Barcode, QR Code solutions are employed for asset ID. Some examples are NXP RFID Tags

[76] There are some NXP security products & solutions mentioned in the list. They are simply some examples (representative) of multitude of security solutions available in the market. System integrators are advised to select the appropriate one from any vendor of their choice that best fits their requirements & costs.

[77] https://www.nxp.com/products/rfid-nfc/mifare-hf:MC_53422



	UCODE	ICODE	NTAG	HITAG
Operating frequency	Ultra-high frequency (UHF), 840 - 960 MHz	High frequency (HF), 13.56 MHz	NFC, 13.56 MHz	Low frequency (LF), 125 kHz
Standard	▪ ISO 18000-63 ▪ UHF EPC Gen 2V2	▪ ISO 15693 ▪ ISO 18000, 3-1/3-3 ▪ EPC HF ▪ Pending T5T	▪ ISO 14443A ▪ NFC Forum Tag Type 2	▪ ISO 14223 ▪ ISO 11784/85
Applications	▪ Inventory and supply chain management ▪ Retail apparel ▪ Automotive ▪ Automatic vehicle identification ▪ Track and trace	▪ Library ▪ FMCG ▪ Brand protection ▪ Track and trace	▪ Brand protection ▪ Consumer interaction	▪ Livestock tracking ▪ Asset tagging
Mobile Device Readable	✓ With plug-in UHF reader device	✓ NFC	✓ NFC	-
Operating Distance	~ 10 m+	~ 1 m ~ 7 cm with NFC smartphone	~ 5 cm	~ 1 m

Figure 85: NFC/RFID tags, labels, and readers, featuring security, memory and interface options[78].

- **Root of Trust:** These are security solution which forms the foundation of any secure network. Usually every communication entity in a network derives its trust (i.e. authenticated) from this trust anchor to build a chain of trust in the network. Only authenticated entities should be allowed to communicate within any network. Typically secure elements and trusted platform module or TPMs are used as authentication solutions. Apart from establishing authenticity, they provide integrity, tamper resistance & identity to the system. Some examples could be NXP A1006, A71CH/CL/ SE050[79].

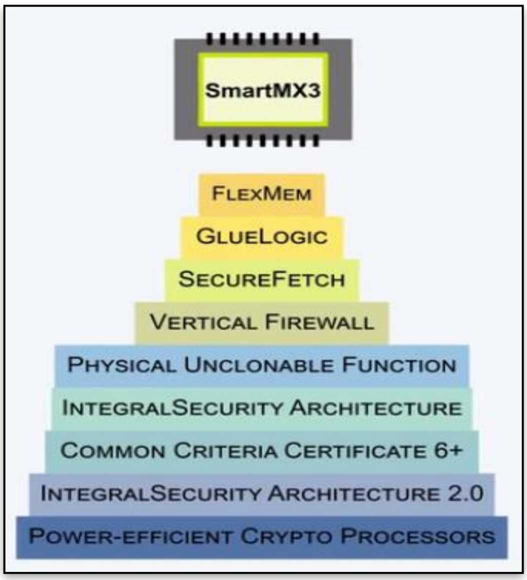

SmartMX3
FlexMem
GlueLogic
SecureFetch
Vertical Firewall
Physical Unclonable Function
IntegralSecurity Architecture
Common Criteria Certificate 6+
IntegralSecurity Architecture 2.0
Power-efficient Crypto Processors

Figure 86: NXP SmartMX Microcontroller family devices comprise security and flexibility as key features[80]

  - Certified security standards (CC EAL6+, GlobalPlatform, EMVCo)

[78] https://www.nxp.com/products/rfid-nfc:RFID-NFC
[79] https://www.nxp.com/products/security-and-authentication/authentication:MC_71548
[80] https://www.nxp.com/products/security-and-authentication/security-controllers:MC_71108

- Tamper resistant
- Contact & contactless mode of operation
- Turnkey solution for easy integration
- Typical applications: counterfeit protection, profile of service, and secure machine-to-machine communication & other IoT/I4.0 Applications

- **Secure execution platform**: These are composed on microcontrollers and microprocessors which can isolate security processes from regular applications. They provide sandboxed environment for security applications with logical/physical separation of computational and communication resources. Typical examples are ARM's TrustZone based Trusted Execution Environment or TEE. They are less secure than a secure element but offer high performance computation platform that processes huge amount of data and much faster rate. Some examples could be NXP i.MX series of processors[81]

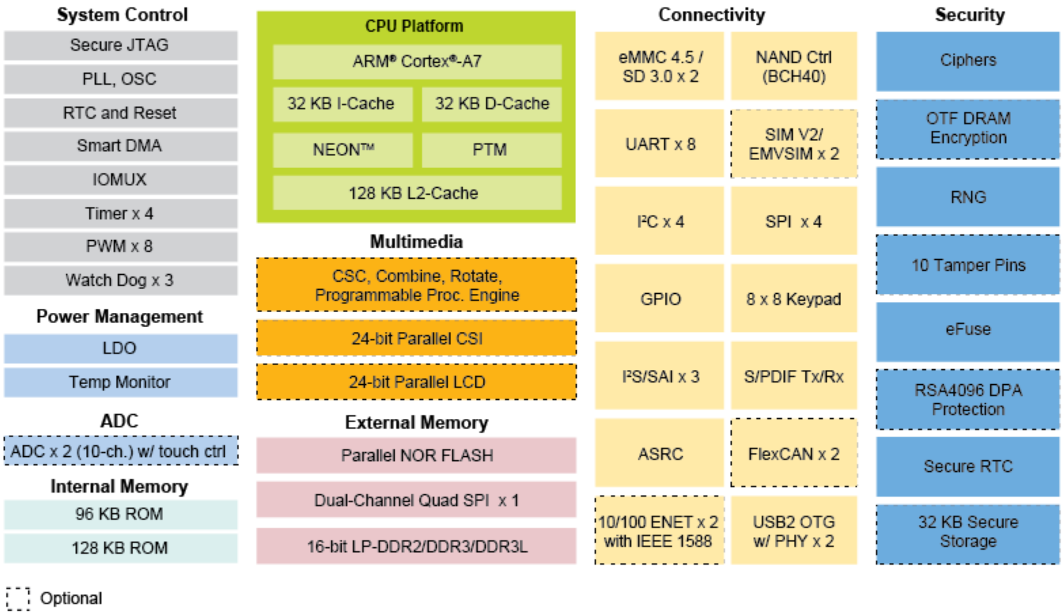


Figure 87: i.MX6 processor with security features (in blue)[82]

Provides advanced security capabilities:

- Anti-tamper peripherals
- Trusted Execution Environment (ARM TrustZone)
- Boot ROM to support encrypted firmware updates
- Automatic decryption from external serial flash
- HW support for public key cryptography
- Complies with the security standards

In addition, Table 6 summarizes standard security countermeasures based on the threat types. The conventional ICT systems and IoT solutions require that security & privacy are addressed in every layer of open systems interconnection or OSI model. The OSI model of IoT is scale down version of standard IT 7-layer model as shown below.

[81] https://www.nxp.com/products/processors-and-microcontrollers:MICROCONTROLLERS-AND-PROCESSORS

[82] https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i.mx-applications-processors:IMX_HOME

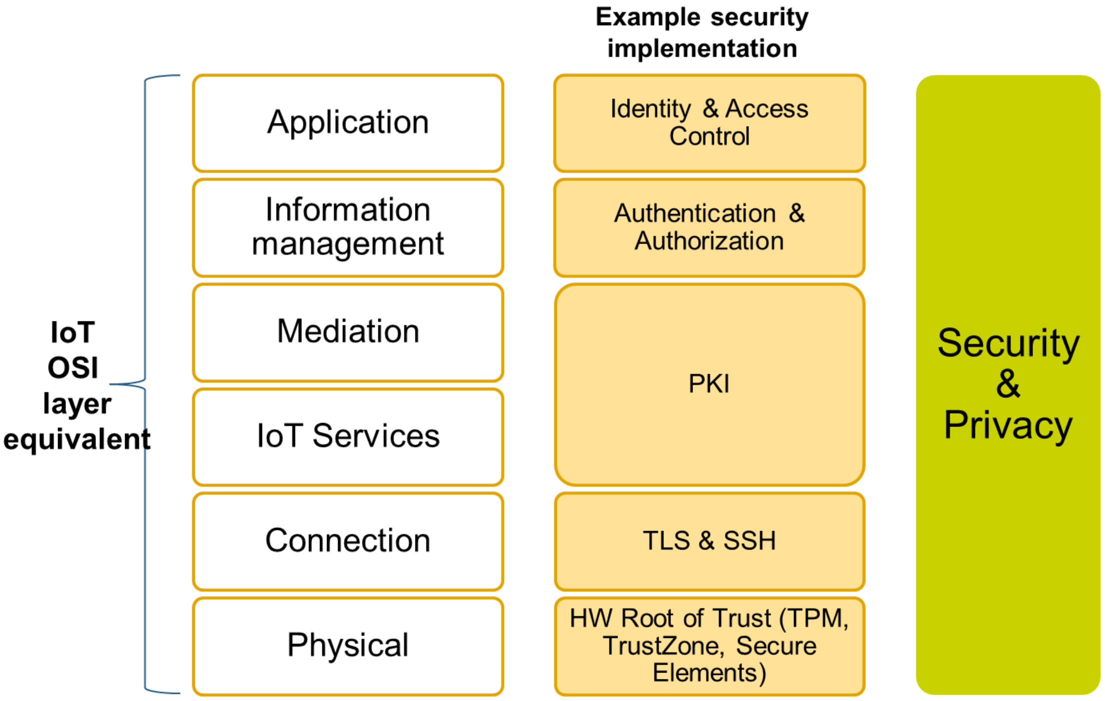

Example security implementation
IoT OSI layer equivalent
Application
Information management
Mediation
IoT Services
Connection
Physical
Identity & Access Control
Authentication & Authorization
PKI
TLS & SSH
HW Root of Trust (TPM, TrustZone, Secure Elements)
Security & Privacy

Figure 88: IoT OSI security & privacy representation

The above figure shows possible security countermeasures in each layer. PKI, OpenSSL & MBED-TLS are some examples of network security.

8.4. Privacy & Data Ownership

Any company planning to share data within its value chain with other companies, always raises major concerns about security and privacy of data. Some of the specific concerns are:

- When data is generated, collected and processed throughout the value chain, how can ownership of data be defined?
- How can a company protect its own specific data, or that of a specific partner, while still sharing some data with other partners?

In the wake of General Data Protection Regulation (GDPR)[83], now, there exists a defined set of data protection rules for all the companies operating in EU, regardless of their base location. Privacy by Design is one such concept that is enforced by GDPR. Privacy by Design entails the inclusion of data protection from the onset of the designing of systems, rather than as an addition. To be more specific, “the controller (e.g. company and / or its partners) shall implement appropriate technical and organizational measures in an effective way in order to meet the requirements of this regulation and protect the rights of data owners”[84].

Privacy includes not just anonymity but also unlink ability. I.e. identity of the user or device has to be kept secret as well as every transaction that a person or a device makes in the network should not be traceable to a person or past transactions. Some of the cryptographic solutions that help preserving privacy are homomorphic encryption and attribute based cryptography.

9. Conclusion and Future Work

9.1. Conclusion

This document includes an analysis of the current state of the art and alternative technologies for the architecture proposed in the I2PANEMA project.

[83] https://gdpr-info.eu/
[84] https://gdpr-info.eu/issues/privacy-by-design/

The result is a document that can be interpreted as a registry of the different design decisions that have been made during the period of work reported.

In a context where the information is a major asset and the systems, specially IT based, change and evolve rapidly, it is important to find a proper balance between detailed technical specifications and enough flexibility for future changes, adapting to new conditions in the markets, the emergence of technologies or variations in the approach of the ports that could be faced in the near future

9.2. Future Work

This is a State of the Art document. Therefore, is results are definitive and there will be no second version during the project. Its results should be taken into account in deliverables such as D1.3 (I2PANEMA Reference Architecture).

References

[1] Perry Lea. Internet of things for architects. IoT Architecture and Core IoT Modules. Birmingham: Packt Publishing Ltd; 2018. p. 26-38.

[2] Fremantle, Paul. (2015). A Reference Architecture for the Internet of Things. 10.13140/RG.2.2.20158.89922.

[3] Shanzhi Chen, HuiXu, DakeLiu, Bo Hu, and Hucheng Wang, " A Vision of IoT: Applications, Challenges and Opportunities with China Perspective ", IEEE Internet of Things Journal, Vol. 1, No. 4, August 2014.

[4] Atefeh Torkaman and M.A. Seyyedi. Analyzing IoT Reference Architecture Models. International Journal of Computer Science and Software Engineering (IJCSSE), Volume 5, Issue 6, August 2016. Page 154 – 160.

[5] Etsi.org. (2019). [online] Available at: https://www.etsi.org/deliver/etsi_ts/102600_102699/102690/01.01.01_60/ts_102690v010101p.pdf [Accessed 1 Oct. 2019].

[6] Gisfi.org. (2019). [online] Available at: https://www.gisfi.org/wg_documents/GISFI_IoT_201206218.doc [Accessed 1 Oct. 2019].

[7] AWS Industrial Predictive Maintenance. (2019). [online] Available at: https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/aws-industrial-PdM-ML-modbus-RA.pdf?did=wp_card&trk=wp_card.

[8] Microsoft Azure IoT Reference Architecture. (2018). [online] Available at: http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf.

[9] IBM Knowledge Center IBM.com. (2019). IBM Knowledge Center. [online] Available at: https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/overview/architecture.html [Accessed 1 Oct. 2019].

[10] CISCO, The Internet of Things Reference Model. (2019). [online] Available at: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf [Accessed 1 Oct. 2019].

[11] Internet of Things: Role of Oracle Fusion Middleware. (2019). [online] Available at: http://www.oracle.com/us/solutions/machine-to-machine/iot-wp-2190408.pdf [Accessed 1 Oct. 2019].

[12] "IDS Reference Architecture Model - Version 2.0"; Otto, Lohmann et.al.; International Data Spaces Association, Dortmund, 2018.

[13] T. P. Raptis, A. Passarella, und M. Conti, „Data Management in Industry 4.0: State of the Art and Open Challenges“, IEEE Access, Bd. 7, S. 97052–97093, 2019, doi: 10.1109/ACCESS.2019.2929296.

[14] D. Laney, „3D data management: Controlling data volume, velocity and variety“, META group research note, Bd. 6, Nr. 70, 2001.

[15] Definition of Workflow. (2019). [online] Available at: https://de.wikipedia.org/wiki/Arbeitsablauf

[16] IBM Knowledge Center, Operation Workflows. (2019). IBM Knowledge Center. [online] Available at: https://www.ibm.com/support/knowledgecenter/SSRMWJ_6.0.0.5/com.ibm.isim.doc_6.0.0.5/planning/cpt/cpt_ic_wkflo_opwkflo.htm