**Name: Syed Aqib**

**Matriculation Number: 1185953**

**Program: Master of Engineering in Information Technology**

**Submitted To:**

**Prof. Andreas Pech   
& Damir Dobric**

Implementation of IOT APi using OPC ua

Contents

[Internet of Things 2](#_Toc487401131)

[History 3](#_Toc487401132)

[Application of IOT 4](#_Toc487401133)

[Home Automation 4](#_Toc487401134)

[Smart Cities 5](#_Toc487401135)

[Industrial Internet of Things and its Benefits 9](#_Toc487401136)

[The Industrial IoT in Connected Logistics and Transportation 10](#_Toc487401137)

[The Industrial Internet of Things in context: Industry 4.0 and Industrial Internet 11](#_Toc487401138)

[OPC Unified Architecture 13](#_Toc487401139)

[Overview and Concepts 15](#_Toc487401140)

[Security Model 15](#_Toc487401141)

[Address Space 15](#_Toc487401142)

[Services 16](#_Toc487401143)

[Information Model 16](#_Toc487401144)

[Mappings 16](#_Toc487401145)

[Data Access 16](#_Toc487401146)

[Alarms and Conditions 17](#_Toc487401147)

[Programs 17](#_Toc487401148)

[Historical Access 17](#_Toc487401149)

[Discovery 17](#_Toc487401150)

[Aggregates 17](#_Toc487401151)

[Information Model 18](#_Toc487401152)

[Object Types 20](#_Toc487401153)

[Variable Types 21](#_Toc487401154)

[Data Type: 21](#_Toc487401155)

[Reference types 22](#_Toc487401156)

[State Machines 22](#_Toc487401157)

[Integration of OPC UA in IOT API 24](#_Toc487401158)

[Functionality of the Project 24](#_Toc487401159)

[OPC Connector 24](#_Toc487401160)

[Unit Test 24](#_Toc487401161)

[References 26](#_Toc487401162)

# Internet of Things

The **Internet of things** (**IoT**) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data[1]. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies” and for these purposes a "thing" is "an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks".The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities[2]. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of about 30 billion objects by 2020.

Typically, IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications. The interconnection of these embedded devices (including smart objects), is expected to usher in automation in nearly all fields, while also enabling advanced applications like a smart grid, and expanding to areas such as smart cities.

"Things", in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "Things" as an "inextricable mixture of hardware, software, data and service".

These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices. Current market examples include home automation (also known as smart home devices) such as the control and automation of lighting, heating (like smart thermostat), ventilation, air conditioning (HVAC) systems, and appliances such as washer/dryers, robotic vacuums, air purifiers, ovens, or refrigerators/freezers that use Wi-Fi for remote monitoring.

As well as the expansion of Internet-connected automation into a plethora of new application areas, IoT is also expected to generate large amounts of data from diverse locations, with the consequent necessity for quick aggregation of the data, and an increase in the need to index, store, and process such data more effectively.[2] IoT is one of the platforms of today's Smart City, and Smart Energy Management Systems.

## History

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine learning, commodity sensors, and embedded systems. This means that the traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things (IoT).

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance,able to report its inventory and whether newly loaded drinks were cold.Mark Weiser's seminal 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IoT. In 1994 Reza Raji described the concept in *IEEE Spectrum* as "[moving] small packets of data to a large set of nodes, to integrate and automate everything from home appliances to entire factories". Between 1993 and 1996 several companies proposed solutions like Microsoft's at Work or Novell's NEST. However, only in 1999 did the field start gathering momentum. Bill Joy envisioned Device to Device (D2D) communication as part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.[3]

The concept of the Internet of things became popular in 1999, through the Auto-ID Center at MIT and related market-analysis publications. Radio-frequency identification (RFID) was seen by Kevin Ashton (one of the founders of the original Auto-ID Center) as a prerequisite for the Internet of things at that point. Ashton prefers the phrase "Internet *for* Things." If all objects and people in daily life were equipped with identifiers, computers could manage and inventory them. Besides using RFID, the *tagging* of things may be achieved through such technologies as near field communication, barcodes, QR codes and digital watermarking.

In its original interpretation, one of the first consequences of implementing the Internet of things by equipping all objects in the world with minuscule identifying devices or machine-readable identifiers would be to transform daily life. For instance, instant and ceaseless inventory control would become ubiquitous. A person's ability to interact with objects could be altered remotely based on immediate or present needs, in accordance with existing end-user agreements. For example, such technology could grant motion-picture publishers much more control over end-user private devices by remotely enforcing copyright restrictions and digital rights management, so the ability of a customer who bought a Blu-ray disc to watch the movie could become dependent on the copyright holder's decision, similar to Circuit City's failed DIVX.

## Application of IOT

There are a diverse set of areas in which intelligent applications have been developed. These applications are not yet readily available; however, preliminary research indicates the potential of IoT in improving the quality of life in our society. Some uses of IoT applications are in home automation, fitness tracking, health monitoring, environment protection, smart cities, and industrial settings.

### Home Automation

Smart homes are becoming more popular today because of two reasons. First, the sensor and actuation technologies along with wireless sensor networks have significantly matured. Second, people today trust technology to address their concerns about their quality of life and security of their homes.

In smart homes, various sensors are deployed, which provide intelligent and automated services to the user. They help in automating daily tasks and help in maintaining a routine for individuals who tend to be forgetful. They help in energy conservation by turning off lights and electronic gadgets automatically.

Prediction algorithms are used to predict the sequence of events in a home. A sequence matching algorithm maintains sequences of events in a queue and also stores their frequency. Then a prediction is made using the match length and frequency. Other algorithms used by similar applications use compression based prediction and Markov models.

Energy conservation in smart homes is typically achieved through sensors and context awareness. The sensors collect data from the environment (light, temperature, humidity, gas, and fire events). This data from heterogeneous sensors is fed to a context aggregator, which forwards the collected data to the context aware service engine. This engine selects services based on the context. For example, an application can automatically turn on the AC when the humidity rises. Or, when there is a gas leak, it can turn all the lights off.

Smart home applications are really beneficial for the elderly and differently abled. Their health is monitored and relatives are informed immediately in case of emergencies. Floors are equipped with pressure sensors, which track the movement of an individual across the smart home and also help in detecting if a person has fallen down.[4] In smart homes, CCTV cameras can be used to record events of interest. These can then be used for feature extraction to find out what is going on.

In specific, fall detection applications in smart environments are useful for detecting if elderly people have fallen down.

### Smart Cities

#### Smart Transport

Smart transport applications can manage daily traffic in cities using sensors and intelligent information processing systems. The main aim of intelligent transport systems is to minimize traffic congestion, ensure easy and hassle-free parking, and avoid accidents by properly routing traffic and spotting drunk drivers. The sensor technologies governing these types of applications are GPS sensors for location, accelerometers for speed, gyroscopes for direction, RFIDs for vehicle identification, infrared sensors for counting passengers and vehicles, and cameras for recording vehicle movement and traffic. There are many types of applications in this area:

1. Traffic surveillance and management applications: vehicles are connected by a network to each other, the cloud, and to a host of IoT devices such as GPS sensors, RFID devices, and cameras. These devices can estimate traffic conditions in different parts of the city. Custom applications can analyze traffic patterns so that future traffic conditions can be estimated. Yu et al. implement a vehicle tracking system for traffic surveillance using video sequences captured on the roads. Traffic congestion detection can also be implemented using smartphone sensors such as accelerometers and GPS sensors. These applications can detect movement patterns of the vehicle while the user is driving. Such kind of information is already being collected by Google maps and users are using it to route around potentially congested areas of the city
2. Applications to ensure safety: smart transport does not only imply managing traffic conditions. It also includes safety of people travelling in their vehicles, which up till now was mainly in the hands of drivers. There are many IoT applications developed to help drivers become safer drivers. Such applications monitor driving behavior of drivers and help them drive safely by detecting when they are feeling drowsy or tired and helping them to cope with it or suggesting rest. Technologies used in such applications are face detection, eye movement detection, and pressure detection on the steering (to measure the grip of the driver’s hands on the steering). A smartphone application, which estimates the driver’s driving behavior using smartphone sensors such as the accelerometer, gyroscope, GPS, and camera, has been proposed by Eren et al. It can decide whether the driving is safe or rash by analyzing the sensor data.
3. Intelligent parking management (see Figure 9: in a smart transportation system, parking is completely hassle free as one can easily check on the Internet to find out which parking lot has free spaces. Such lots use sensors to detect if the slots are free or occupied by vehicles. This data is then uploaded to a central server.
4. Smart traffic lights: traffic lights equipped with sensing, processing, and communication capabilities are called smart traffic lights. These lights sense the traffic congestion at the intersection and the amount of traffic going each way. This information can be analyzed and then sent to neighboring traffic lights or a central controller. It is possible to use this information creatively. For example, in an emergency the traffic lights can preferentially give way to an ambulance. When the smart traffic light senses an ambulance coming, it clears the path for it and also informs neighboring lights about it. Technologies used in these lights are cameras, communication technologies, and data analysis modules. Such systems have already been deployed in Rio De Janeiro.
5. Accident detection applications: a smartphone application designed by White et al. detects the occurrence of an accident with the help of an accelerometer and acoustic data. It immediately sends this information along with the location to the nearest hospital. Some additional situational information such as on-site photographs is also sent so that the first responders know about the whole scenario and the degree of medical help that is required.

#### Smart Water Systems

Given the prevailing amount of water scarcity in most parts of the world, it is very important to manage our water resources efficiently. As a result most cities are opting for smart solutions that place a lot of meters on water supply lines and storm drains. A good reference in this area is the paper by Hauber-Davidson and Idris.[5] They describe various designs for smart water meters. These meters can be used to measure the degree of water inflow and outflow and to identify possible leaks. Smart water metering systems are also used in conjunction with data from weather satellites and river water sensors. They can also help us predict flooding.

#### Examples of Smart Cities

Barcelona and Stockholm stand out in the list of smart cities. Barcelona has a CityOS project, where it aims to create a single virtualized OS for all the smart devices and services offered within the city. Barcelona has mainly focused on smart transportation and smart water. Smart transportation is implemented using a network of sensors, centralized analysis, and smart traffic lights. On similar lines Barcelona has sensors on most of its storm drains, water storage tanks, and water supply lines. This information is integrated with weather and usage information. The result of all of this is a centralized water planning strategy. The city is able to estimate the water requirements in terms of domestic usage and industrial usage and for activities such as landscaping and gardening.

Stockholm started way back in 1994, and its first step in this direction was to install an extensive fiber optic system. Subsequently, the city added thousands of sensors for smart traffic and smart water management applications. Stockholm was one of the first cities to implement congestion charging. Users were charged money, when they drove into congested areas. This was enabled by smart traffic technologies. Since the city has a solid network backbone, it is very easy to deploy sensors and applications. For example, recently the city created a smart parking system, where it is possible to easily locate parking spots nearby. Parking lots have sensors, which let a server know about their usage. Once a driver queries the server with her/his GPS location, she/he is guided to the nearest parking lot with free slots. Similar innovations have taken place in the city’s smart buildings, snow clearance, and political announcement systems.

#### Social Life and Entertainments

Social life and entertainment play an important role in an individual’s life. Many applications have been developed, which keep track of such human activities. The term “opportunistic IoT” refers to information sharing among opportunistic devices (devices that seek to contact other devices) based on movement and availability of contacts in the vicinity. Personal devices such as tablets, wearables, and mobile phones have sensing and short range communication capabilities. People can find and interact with each other when there is a common purpose.

Circle Sense is an application, which detects social activities of a person with the help of various types of sensor data. It identifies the social circle of a person by analyzing the patterns of social activities and the people present in those activities. Various types of social activities and the set of people participating in those activities are identified. It uses location sensors to find out where the person is and uses Bluetooth for searching people around her. The system has built in machine learning algorithms, and it gradually improves its behavior with learning.

Affective computing is a technology, which recognizes, understands, stimulates, and responds to the emotions of human beings. There are many parameters, which are considered while dealing with human affects such as facial expressions, speech, body gestures, hand movements, and sleep patterns. These are analyzed to figure out how a person is feeling. The utterance of emotional keywords is identified by voice recognition and the quality of voice by looking at acoustic features of speech.

One of the applications of affective computing is Camy, an artificial pet dog, which is designed to interact with human beings and show affection and emotions. Many sensors and actuators are embedded in it. It provides emotional support to the owner, encourages playful and active behavior, recognizes its owner, and shows love for her and increases the owner’s communication with other people. Based on the owner’s mood, Camy interacts with the owner and gives her suggestions.[6]

#### Health and Fitness

IoT appliances have proven beneficial in the health and wellness domains. Many wearable devices are being developed, which monitor a person’s health condition.

Health applications make independent living possible for the elderly and patients with serious health conditions. Currently, IoT sensors are being used to continuously monitor and record their health conditions and transmit warnings in case any abnormal indicators are found. If there is a minor problem, the IoT application itself may suggest a prescription to the patient.

IoT applications can be used in creating an Electronic Health Record (EHR), which is a record of all the medical details of a person. It is maintained by the health system. An EHR can be used to record allergies, surges in blood sugar and blood pressure.

Stress recognition applications are also popular. They can be realized using smartphone sensors. Wang et al. describe an application, which measures the stress level of a college student and is installed on the student’s smartphone. It senses the locations the student visits during the whole day, the amount of physical activity, amount of sleep and rest, and her/his interaction and relationships with other people (audio data and calls). In addition, it also conducts surveys with the student by randomly popping up a question in the smartphone. Using all of this data and analyzing it intelligently, the level of stress and academic performance can be measured.[5]

In the fitness sector, we have applications that monitor how fit we are based on our daily activity level. Smartphone accelerometer data can be used for activity detection by applying complex algorithms. For example, we can measure the number of steps taken and the amount of exercise done by using fitness trackers. Fitness trackers are available in the market as wearables to monitor the fitness level of an individual. In addition, gym apparatus can be fitted with sensors to count the number of times an exercise is performed. For example, a smart mat can count the number of exercise steps performed on it. This is implemented using pressure sensors on the mat and then by analyzing the patterns of pressure, and the shape of the contact area.

#### Smart Environment and Agriculture

#### 

Environmental parameters such as temperature and humidity are important for agricultural production. Sensors are used by farmers in the field to measure such parameters and this data can be used for efficient production. One application is automated irrigation according to weather conditions.

Production using greenhouses is one of the main applications of IoT in agriculture. Environmental parameters measured in terms of temperature, soil information, and humidity are measured in real time and sent to a server for analysis. The results are then used to improve crop quality and yield.

Pesticide residues in crop production are detected using an Acetylcholinesterase biosensor. This data is saved and analyzed for extracting useful information such as the sample size, time, location, and amount of residues. We can thus maintain the quality of the crop. Moreover, a QR code can be used to uniquely identify a carton of farm produce. Consumers can scan the QR code and check the amount of pesticides in it (via a centralized database) online before buying.[4]

#### Supply Chain and Logistics

IoT tries to simplify real world processes in business and information systems. The goods in the supply chain can be tracked easily from the place of manufacture to the final places of distribution using sensor technologies such as RFID and NFC. Real time information is recorded and analyzed for tracking. Information about the quality and usability of the product can also be saved in RFID tags attached with the shipments.

Bo and Guangwen explain an information transmission system for supply chain management, which is based on the Internet of Things. RFID tags uniquely identify a product automatically and a product information network is created to transmit this information in real time along with location information. This system helps in automatic collection and analysis of all the information related to supply chain management, which may help examine past demand and come up with a forecast of future demand. Supply chain components can get access to real time data and all of this information can be analyzed to get useful insights. This will in the long run improve the performance of supply chain systems.

## Industrial Internet of Things and its Benefits

**The Industrial Internet of Things (IIoT) describes the Internet of Things as it is used across several industries such as manufacturing, logistics, oil and gas, transportation, energy/utilities, mining and metals, aviation and other industrial sectors.**

Just like the Internet of Things in general, the Industrial IoT covers many use cases, industries and applications. Initially focusing on the optimization of operational efficiency and rationalization/automation/maintenance, with an important role for the convergence of IT and OT, the Industrial Internet of Things opens plenty of opportunities in moving towards an on demand service model, new ways of servicing customers and the creation of new revenue models, often in unexpected ways.

Several so-called cross-industry IoT use cases are also present in the industrial Internet of Things. Examples include connected vehicles and smart buildings. The precise industries which are ‘covered’ under the Industrial Internet of Things can be broadened to smart cities, healthcare and so forth. Different industry bodies have different approaches as you will discover. The two main ones are ‘Plattform Industrie 4.0’ (Industry 4.0, originally restricted to manufacturing) and the Industrial Internet Consortium (covering more industries with its Industrial Internet view).[6]

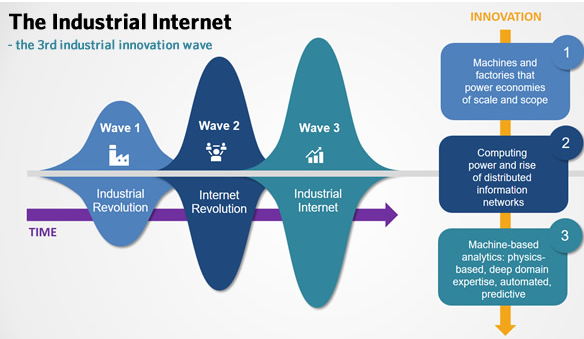
### **The Industrial IoT in Connected Logistics and Transportation**

**Transportation represent the second largest market from an Internet of Things spending perspective. Transportation and logistics (T&L) firms are looking to move up the value chain with advanced communication and monitoring systems, enabled by IoT.**

The transportation market reached an IoT spend of $78 billion and is poised to continue to grow rapidly, just as is the case for the IoT manufacturing market. The main use case in transportation is freight monitoring, good for a large majority of overall transportation IoT spend with a total of $55.9 billion and remaining a key driver in the market until 2020.

**If we look at the overall Industrial IoT evolutions in transportation and logistics, we see the growing emergence of a digital supply chain and connected logistics reality,** which is at the same time one of the challenges for the manufacturing industry and the T&L market as such as many players don’t have a digital strategy in place and are urged to speed up their digital transformation efforts. As you can read here, about 20 percent of digital transformation costs in T&L will be allocated to supply chain transformation. It’s clear that the Internet of Things plays an important role here.[5]

**This is also the case for the four pillars of a connected logistics system as Technavio defined them:** IT security, communication systems, supply chain monitoring systems and vehicle/transport tracking. Along with the cloud and analytics, the Industrial Internet of Things is a driver in the connected logistics landscape and freight monitoring leads the pack.



### **The Industrial Internet of Things in context: Industry 4.0 and Industrial Internet**

**We saw how the initial purpose of Industrial IoT projects typically is to automate, save costs and optimize in often rather siloed and ad hoc ways and how it’s important to have a more holistic view and strategy, whereby there is a shift towards goals of innovation, better customer-centric service offerings, leveraging new sources of data-driven revenues, building ecosystems of value and ecosystem-wide digital transformation goals.**

As mentioned, the Industrial Internet of Things enables industries to rethink business models. Generating actionable information and knowledge from Industrial IoT devices, for instance, enables the creation of a data sharing ecosystem with new revenue streams and partnerships.

The other way around, aggregated and real-time data from sensors and from information sources which can be ‘consulted’ via built-in capabilities also lead to the development of robots which can take specific actions because of these built-in capabilities whereby Industrial IoT becomes a driver of ‘decision-making’ devices. This already happens in some warehouses and is called the Internet of Robotic Things (IoRT). It is used in an Industrial IoT context but also a consumer IoT context.

**The usage of the Industrial Internet of Things, within a broader context, ultimately leads from specific projects and ‘smart’ Industrial IoT use cases to connected ecosystems.**

Supply chains become connected supply chains, factories become connected factories and so forth. In this sense, the connectedness stretches far beyond the simple connectedness – and data-driven results – of devices and industrial assets to a more connected ecosystem, whereby the extended enterprise gains a new meaning.

Therefore, the Industrial Internet of Things is mainly used in the context of Industry 4.0, the Industrial Internet and related initiatives across the globe, which all have their own names, from smart production to smart factory or intelligent industry. Industry 4.0 (which we tackle more in depth in an Industrial IoT context below) describes a new industrial revolution with a focus on automation, innovation, data, cyber-physical systems, processes and people. On top of Industrial IoT, Industry 4.0 also is about other technologies; which are related with it.

Examples include robotics, cloud computing and the evolutions in operational technology (OT). In the Industrial Internet of Things, IT and OT need and meet each other. Industry 4.0 further refers to [cyber-physical](https://www.i-scoop.eu/industry-4-0/#The_building_blocks_of_Industry_40_cyber-physical_systems) production systems (CPPS) and typical embeds the so-called [third platform](https://www.i-scoop.eu/digital-transformation/the-third-platform/) technologies and accelerators of what is known as the digital transformation or [DX economy](https://www.i-scoop.eu/the-acceleration-of-third-platform-innovation-here-comes-the-dx-economy/).

**The Industrial Internet of Things is also strongly related with what GE called the Industrial Internet or the third industrial innovation wave and is comparable to Industry 4.0.**

In fact, both are the same and the Industrial Internet Consortium and Industry 4.0 Platform work together. Whereas Industry 4.0, a concept which found its roots in Europe (Germany mainly) says we are entering the fourth industrial evolution, the Industrial Internet approach speaks about the third industrial innovation wave. No matter, the exact number, both express the same realities with, among others, machine-based analytics, CPPS (bridging digital and physical) and the essence of the Internet of Things in an industrial context, to name a few. More about the Industrial Internet of Things in the scope of Industry 4.0, the Industrial Internet, industry bodies and standardization.

## OPC Unified Architecture

OPC(Open Platform Communication) Unified Architecture , shortlyOPC UA , is an industrial M2M - communication protocol  As the newest of all OPC specifications of the*OPC Foundation* to OPC UA is significantly different from its predecessors, particularly by the ability to machine data ( control values , measured values , parameters , etc.) not only to transport, but also machine-readable, semantically describe.[11]

After more than three years of specification work and a year of prototype implementation, the first version of the Unified Architecture was adopted in the autumn of 2006. In February 2009, a revised version of Parts 1 to 5 and 8 and the first version of Parts 6 and 7 were published.

After more than three years of specification work and a year of prototype implementation, the first version of the Unified Architecture was adopted in the autumn of 2006. In February 2009, a revised version of Parts 1 to 5 and 8 and the first version of Parts 6 and 7 were published.

Introduction OPC Unified Architecture (OPC UA) is described in a layered set of specifications broken into parts. It is purposely described in abstract terms and only in selected parts coupled to existing technology on which software can be built. This layering is intentional and helps isolate changes in OPC UA from changes in the technology used to implement it.[12]

The OPC UA specifications are organized as a multi-part document combined in the following sets:

• Core specification   
• Access type specification   
• Utility specification

The first set specifies core capabilities of OPC UA. Those core capabilities define the concept and structure of the Address Space and the services that operate on it. The access type set applies those core capabilities to specific models of data access. Like in OPC Classic, there are distinguished: Data Access (DA), Alarms and Conditions (A&C) and Historical Access (HA). A new access mode is provided because of introducing the programs concept and aggregation mechanisms. This set also specifies the UA server discovery process.[13] Those mechanisms are not directly dedicated to support data exchange, but play a very important role in the whole interoperability process. The core set contains the following specifications:

• Part 1 – Overview and Concepts: presents the concepts and overview of OPC Unified Architecture.   
• Part 2 – Security Model: describes the model for securing interactions between OPC UA clients and servers.   
• Part 3 – Address Space Model: describes an object model that servers use to expose underlying real-time processes to create an OPC UA connectivity space.   
• Part 4 – Services: specifies the services provided by OPC UA servers.   
• Part 5 – Information Model: specifies information representations - types that OPC UA servers use to expose underlying real-time processes.   
• Part 6 – Mappings: specifies transport mappings and data encodings supported by OPC UA.   
• Part 7 – Profiles: introduces the concept of profiles and defines available profiles that are groups of services or functionality.

The access type set contains the following specifications:   
  
• Part 8 – Data Access: specifies the use of OPC UA for data access.   
• Part 9 – Alarms and Conditions: specifies the use of OPC UA support for accessing alarms and conditions.  
 • Part 10 – Programs: specifies OPC UA support for accessing programs.   
• Part 11 – Historical Access: specifies the use of OPC UA for historical access. This access includes both historical data and historical events.   
  
The utility specification parts contain the following specifications:  
  
• Part 12 – Discovery: introduces the concept of the Discovery Server and specifies how OPC UA clients and servers should interact to recognize OPC UA connectivity.   
• Part 13 – Aggregates: describes ways of aggregating data.

Overview and Concepts   
  
This part describes the goal of OPC UA and introduces the following models to achieve it:   
• Address Space and information model to represent structure, behavior, semantics, and infrastructure of the underlying real-time system.   
• Message model to interact between applications.   
• Communication models to transfer data over the network.   
• Conformance model to guarantee interoperability between systems.   
• Security model to guarantee cyber security addressing client/server authorization, data integrity and encryption.

### Security Model

This part describes the OPC UA security model. OPC UA provides countermeasures to resist threats that can be made against the environments in which OPC UA will be deployed. [15]It describes how OPC UA relies upon other standards for security. The proposed architecture is structured in an application layer and a communication layer. Introduced security policies specify which security mechanisms are to be used. The server uses security policies to announce what mechanisms it supports and the client - to select one of those available policies to be used when establishing the connection.

### Address Space

There is no doubt that information technology and process control engineering must be integrated to benefit from macro optimization and synergy effect. To integrate them, we must make systems interoperable. It causes the necessity of exchanging information, but to exchange information, it must be represented as computer centric (save able in a binary memory) and transferable (a stream of bits) data. According to the specification, a set of objects that an OPC UA server makes available to clients as data representing an underlying real-time system is referred to as its Address Space. The breaking feature of the Address Space concept allows representing both real process environment and real-time process behavior - by a unique means, mutually understandable by diverse systems.

### Services

The OPC UA services described in this part are a collection of abstract remote procedure calls that is to be implemented by the servers and called by the clients. The services are considered abstract because no implementation is defined in this part. The part Mappings describes more specific mappings supported for implementation.[16] Separation of the service definition and implementation allows for harmonization with new emerging technologies by making new mappings.

### Information Model

To make the data exposed by the Address Space mutually understandable by diverse systems, the information model part standardizes the information representation as computer centric data. To promote interoperability, the information model defines the content of the Address Space of an empty OPC UA server. This content can be used as a starting browse point to discover all information relevant to any client. Definitions provided in this part are considered abstract because they do not define any representation on the wire. To make the solution open for new technologies, the representation mappings are postponed to the part Mappings. The solution proposed in this model is also open to defining vendor specific representations.

### Mappings

This part defines mappings between abstract definitions contained in the specification (e.g. in the parts: Information Model, Services, Security Model) and technologies that can be used to implement them. Mappings are organized into three groups: data encodings, security protocols and transport protocols. Different mappings are combined to create stack profiles. Profiles This part describes the OPC UA profiles as groups of services or functionality that can be used for conformance level certification. Individual features are grouped into conformance units, which are further grouped into profiles. All OPC UA applications shall implement at least one stack profile and can only communicate with other OPC UA applications that implement the same stack profile. Servers and clients will be tested against the profiles. Servers and clients must be able to describe which of the features they support.[15]

### Data Access

This part describes the information model associated with the Data Access (DA) mode. It particularly includes an additional definition of variable types and a complementary description of Address Space objects. This part also includes additional descriptions of node classes and attributes needed for DA, as well as DA specific usage of services to access process data.

### Alarms and Conditions

This part describes the representation of events and alarms in the OPC UA Address Space and introduces the concepts of condition, dialog, acknowledgeable condition, confirmable condition and alarm. To expose above information, it extends the information model defined in other parts and describes alarm specific uses of services.

### Programs

This part extends the notion of methods and introduces the concept of programs as a complex, state-full functionality in a server or underlying system that can be invoked and managed by a OPC UA client. The provided definitions describe the standard representation of programs as part of the OPC Unified Architecture information model. The specific use of services is also discussed.[16]

### Historical Access

This part describes an extension of the information model associated with Historical Access (HA). It particularly includes additional and complementary definitions of the representation of historical time series data and historical event data. Additionally, this part covers HA specific usage of services to detect and access historical data and events.

### Discovery

The main aim of this part is to address the discovery process that allows the clients to first find servers on the network and then find out how to connect to them. This part describes how UA clients and servers interact to exchange information on resources available on the network in different scenarios. To achieve this goal, there are introduced the concepts of a discovery server that is a placeholder of global scope information and a local discovery server, whose main task is to manage information vital to local resources. Finally, this part describes how to discover UA applications when using common directory service protocols such as UDDI and LDAP.

### Aggregates

This part specifies the information model associated with aggregates and describes how to compute and return aggregates like minimum, maximum, average etc. Aggregates can be used with base (live) data as well as historical (HA) data. This part also addresses the aggregate specific usage of services. Related articles

• OPC Unified Architecture - Main Technological Features (mpostol.wordpress.com) • OPC Books & Academic Articles ( [www.opcfoundation.org](http://www.opcfoundation.org))

• OPC UA Makes Smart Factory Possible (mpostol.wordpress.com) • OPC UA makes cloud computing possible. (mpostol.wordpress.com)

## Information Model

To make systems interoperable, i.e. empower common processing of information by variety of computer systems, the data transfer mechanism must be associated with a consistent information representation model. OPC UA uses an object as a fundamental notion to represent data and behavior of an underlying system. The objects are placeholders of variables, events and methods and are interconnected by references. This concept is like well-known object-oriented programming (OOP) that is a programming paradigm using "objects" – data structures consisting of fields, events and methods – and their interactions to design computer programs. The OPC UA Information Model provides features such as data abstraction, encapsulation, polymorphism, and inheritance. For unification of the information representation the producers (servers) and consumers (clients) use the type notion. The OPC UA object model allows servers to provide type definitions for objects and their components. Type definitions may be abstract, and may be inherited by new types to reflect polymorphism. They may also be common or they may be system-specific.[15] Object types may be defined by standardization organizations, vendors or end-users. Each type must have a globally unique identifier that can be used to provide description of the data semantics from the defining body or organization. Using the type definitions to describe the information exposed by the server allows for:

• Development against type definition.   
• Unambiguous assignment of the semantics to the data expected by the client.

Having defined types in advance, clients may provide dedicated functionality, for example: displaying the information in the context of specific graphics. It greatly improves reusability because of the possibility of designing a unique context for typical Real-time Processes. As an example, the section Adopting Companion Standard Models - Analyzer Devices Integration presents a case of unification of the model for chemical analyzers.

The OPC UA information modelling concept is based on domains. A domain is a named self-contained collection of definitions. Any domain name must be globally unique - it is an identification string that defines a realm of administrative autonomy and authority of responsibility. Type definition from one domain may inherit from type definitions located in other domains. To avoid circular references domains should be organized in layers, which expand step by step the basic model provided by the OPC UA Specification.

Type definitions are exposed in the OPC UA Address Space using the specialized NodeClasses: ObjectType, DataType, ReferenceType, VariableType. The main role of the types represented by the above NodeClasses is to provide a description of the Address Space structure and to allow clients to use this knowledge to navigate to desired information in the Address Space exposed by the server. In other words, this way the clients obtain the definition of the data (metadata) using the following two concepts:

1. NodeClass – as a formal description of the node defining the allowed attributes and references.

2. Type – as a formal description of the node defining values of the allowed attributes and references.   
  
The OPC UA Information Model concept provides a set of predefined types and rules that must be applied to expand it. Even though the OPC UA specification contains a rich set of predefined types, the type concept allows designers to freely define types according to the application needs.[16] New types are derived from the existing ones. The derived types inherit all features from the base types but can include modifications to make the new types more appropriate for information the designers are representing. To expand the standard model, independent domains must be defined. This new information model covered by the domain may be the subject of a companion specification or proprietary release. In any case the definitions must be uniquely named and self-contained except for external type references. All not predefined types (not belonging to the standard domain) must be exposed in the Address Space.

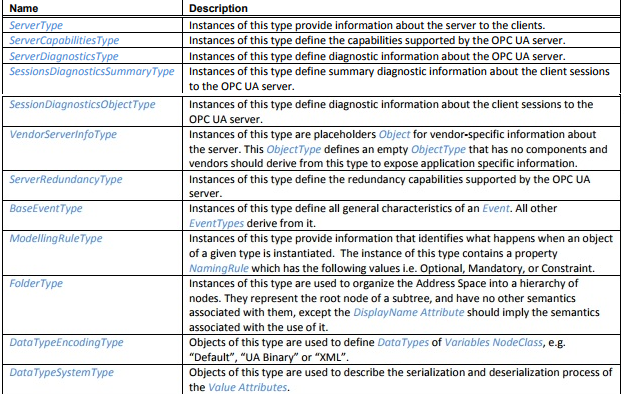
Types are called metadata since they describe the data structure and not the actual data values. Simplifying, we can say that a NodeClass plays a role similar to the shape of a puzzle piece and the represented information is similar to the picture on the piece. Both are needed to enable us to see the final picture. In the above simplification we have lost that the OPC UA Address Space is capable of displaying movies, and not just static pictures. From the above discussion we learn that before nodes making up the Address Space can be instantiated by the server, that Address Space must be designed first. Model designing is a process aimed at defining a set of types and their associations and, next, creating an Address Space representation in a format appropriate for implementation. More detailed description of this topic is captured in Information Model Life-cycle.

The Address Space concept based on types can be a foundation for exposing any information that is required. Clients understand the Address Space concept and have a browse service to navigate the Address Space. Since browsing is based on the incremental and relative passage among Nodes it is apparent that each path must have an entry point defined, so we must address the question as to “where to start". To meet this requirement, the Information Model includes definition to create a predefined structure containing well defined Nodes that can be used as anchors from which a client can discover the Address Space. Thus to design an Address Space instance using predefined new types, we must derive them from the existing ones. At the very beginning the only existing types are the standard ones defined by the OPC Foundation. The available standard types are briefly described in the next sections.

### Object Types

The Object NodeClass is used to define objects as parts involved in the underling real-time process. Each Object in the Address Space has an assigned ObjectType. The OPC UA specification has defined a BaseObjectType from which all other ObjectTypes shall either inherit, directly or indirectly

Many of these standard types are used for describing OPC UA Server functionality and to provide diagnostic information. The BaseEventType has many specialized subtypes to allow handling most common transient Events. System configuration changes, operator interaction and system errors are examples of Events. OPC UA Part 9 – Alarm and Conditions expands on this object type to define alarm and condition events.



### Variable Types

Variable NodeClass is dedicated to provide a value to the clients. To define a Variable two types must be provided.

* VariableType: which describes the type of a variable. A Variable node has a HasTypeDefinition reference to its type definition (depicted as double closed and filled arrows).
* DataType: which describes the type of the variable value. It is assigned to the DataType attribute.

### Data Type:

The type of data provided by the Variable Value attribute is defined by the associated DataType. DataType is pointed out by the DataType attribute of the Variables and VariableType nodes. The DataType attribute is of the NodeId type. In many cases, the value of the DataType attribute – called DataTypeId – will be well-known to clients and servers. Well-known DataTypeIds allow clients to use random addressing and interpret values without having to read the type description from the server. Therefore, servers may use well-known DataTypeIds without representing the corresponding DataType nodes in their Address Space.

DataType NodeClass is dedicated to describe types. In this case, the represented types have a special mission, because they describe data provided by the UA server to clients. For example, VariableTypes to represent items of data a node of the DataType can provide information to clients that the data has a numeric value and the clients reading it can use this knowledge to interpret and process the obtained value – stream of bits.

The BaseDataType is the root of the inheritance tree. The simplified inheritance hierarchy of the standard DataTypes is shown in where the whole sub-tree of built-in types are represented commonly by a single symbol.

Most of the built-in types are similar to those known in other IT systems, except the NodeId type. This type needs some comments, because it is intended to be used by the random addressing mechanism to represent information allowing clients to uniquely identify and access the nodes. This built-in DataType is a structure composed of:

1. namespaceIndex : numeric values used to identify namespaces,

2. identifierType: identifies the type of the NodeId, its format and its scope,

3. identifier: a unique identifier within the context of the namespace.

The namespace is a URI (Unique Resource Identifier) that identifies the naming authority responsible for assigning the identifier element of the NodeId. Namespace URIs are identified by BaseDataType Enumeration Built-in StructureStandard DataTypes inheritance hierarchy numeric values in OPC UA Services to permit a more efficient transfer and processing (e.g. table lookups). Depending on the application requirements, the identifierType field may have the following values:

• NUMERIC: numeric

• STRING: text string

• GUID: Globally Unique Identifier

• OPAQUE: namespace specific format

### Reference types

Reference types are used to create interconnections between nodes. They are not instantiated, i.e. a NodeClass representing a reference is not defined. Instead of instantiating the references, they are added to a collection associated with each node. NodeClass of the node and its type decide what references are allowed to be added to this collection. The base of all references is an abstract References type (Figure 8). There is no semantics associated with it. There are two disjoint sets of standard references:

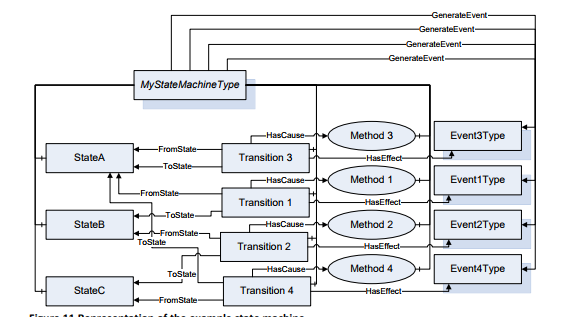
• HierarchicalReferences

• NonHierarchicalReferences

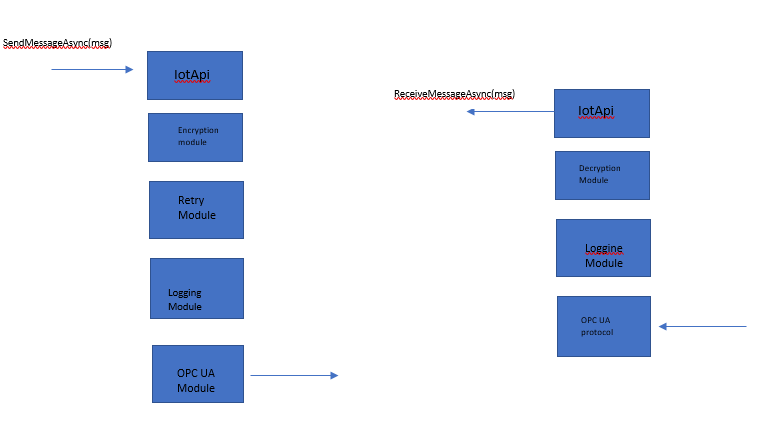
### State Machines

The information model provides constructs that can be used to model discrete object behavior in terms of the states an object can reside in and the transitions that can happen between those states. State machines are built as complex objects using dedicated ObjectTypes, VariableTypes and ReferenceTypes, whose behavior is governed by the rules that must be strictly observed. A state is a condition in which an object can be at some point during its lifetime, for some finite amount of time. A transition is a change of an object from one state (the source state) to another (the target state). The transition is triggered ("fires") when an event of interest – cause - to a given object occurs. According to the information model concept, causes are represented in the form of Methods that have to be called, but a vendor can define other items or have them be internal (i.e. nothing is listed causing the transition). There may also be an action associated with a triggered transition. This action called an “effect” is executed unconditionally before the object enters the target state. Effects are Events that are generated.

The simplified state machine model described above can be freely expanded to provide more complex functionality like sub-machines, parallel states, forks and joins, history states, choices and junctions, etc. State machines are represented in the Address Space as an object of a type derived from the StateMachineType that defines a single Variable of the StateVariableType, which represents the current state of the machine. An instance of the StateMachineType shall generate an event whenever a state change occurs. Transitions are represented as objects of the TransitionType. Each valid transition shall have exactly one FromState reference and exactly one ToState reference, each pointing to an object of the StateType. Using the above terminology we can represent any state machine from Figure. For this diagram it is assumed that MyStateMachineType is derived directly or indirectly from the StateMachineType. All states, transitions and methods are components of this type.



# Integration of OPC UA in IOT API



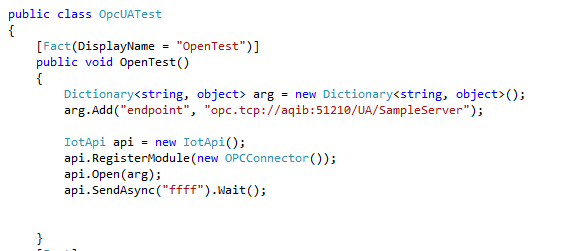
# Functionality of the Project

### OPC Connector

OPC Connector Consists of Client and Server developed using OPC UA. OPC UA uses and endpoint to get connected to OPC UA Server. OPC Clients creates all the necessary certificates and configurations to connect to OPC Server.

### Unit Test

I have wrote two unit tests one send string using api.SendAsync().Wait(). The other test send a list of the items using the same method.





# References

1. Scientific services of the German Bundestag: *Current term: Industrie 4.0.*26.09.2016 , available on 07.10.2016 (PDF, 237kB, German)
2. Scientific services of the German Bundestag: *Current term: Industrie 4.0.*26.09.2016 , available on 07.10.2016 (PDF, 237kB, German)
3. D. Uckelmann, M.-A. Isenberg, M. Teucke, H. Halfar, and B. Scholz-Reiter:*Autonomous Control and the Internet of Things: Increasing Robustness, Scalability and Agility in Logistic Networks* . In: DC Ranasinghe, QZ Sheng, S. Zeadally (eds.): Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Springer; Berlin 2010: 163-181.
4. D. Uckelmann, M.-A. Isenberg, M. Teucke, H. Halfar, and B. Scholz-Reiter:*Autonomous Control and the Internet of Things: Increasing Robustness, Scalability and Agility in Logistic Networks* . In: DC Ranasinghe, QZ Sheng, S. Zeadally (eds.): Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Springer; Berlin 2010: 163-181.
5. CERP-IOT: Internet of Things Strategic Research Roadmap (PDF file, 850 kB). Retrieved on April 8, 2011.
6. *Internet of Things - Networked Living and Working Worlds* . Website of the BMWi Technology Program, sighted on 8 April 2011.
7. http://www.commsvr.com/UAModelDesigner/ OPC Unified Architecture e-book, 2010.
8. Jürgen Lange, Frank Iwanitz, Thomas J. Burke. Von Data Access bis Unified Architecture. Hüthig Fachverlag, 2009.
9. Wolfgang Mahnke, Stefan Helmut Leitner, Matthias Damm. OPC Unified Architecture. Berlin: Springer, 2009.
10. OPC UA Specification: Part 1 – Concepts, Version 1.0 or later. OPC Foundation, 2009.
11. OPC UA Specification: Part 3 – Address Space Model, Version 1.0 or later. OPC Foundation, 2009.
12. OPC UA Specification: Part 4 – Services, Version 1.0 or later. OPC Foundation, 2009. Bibliography 38
13. OPC UA Specification: Part 5 – Information Model, Version 1.0 or later. OPC Foundation, 2009.
14. OPC Unified Architecture Companion Specification for Analyser Devices. OPC Foundation, 2009.
15. OPC Unified Architecture Companion Specification for Devices. OPC Foundation, 2009.
16. OPC UA Object Oriented Internet, Opc-ua-ooi open source project on GitHub,