

Master Thesis

“Security and trust in IoT/M2M – Cloud based platform”

Radostin Stefanov Stefanov

Master in Innovative communication technologies and entrepreneurship

Submission date: June 2013

Supervisor: Bayu Anggorojati, ICTE

Aalborg University

Department of Electronic Systems

**ABSTRACT**

This thesis work considers Machine to Machine (M2M) services platform on the local cloud infrastructure concept. The main objectives of the thesis are to analyze security needs of M2M services and based on this requirement, access control method in such platform will be designed.

In this new approach for local cloud infrastructure different access methods are analysed to determine their security aspects. It is important to understand new message protocols that are used for M2M communications. They have specific requirements and security aspects. The techniques used to secure local cloud model may be implemented by means of network access, policies, authorization and authentication technologies or a combination from all of these. That is why security must be considered on every level of local network. The system also must communicate with outside environment and must be connected to the internet. That is why the connections must made by a proprietary or standard technology that provides interoperability of data and applications.

Typical protection using security certificates and cryptographic algorithms are not enough to ensure the necessary security level in the cloud. When we talk about machine-to-machine communications sometimes small embedded devices have no capabilities to support this type of certificates. That brings new challenges to the security of M2M/IoT environment. Security mechanisms must give users a high level of protection and in the same time they must be not so hard to implement in small embedded devices and easy to manage for users that create they own local cloud.

Trust is the main concern of end users, service providers and different stakeholders in the cloud environment. Because of complex scenario the trust is dividing in three major groups. The first one is the trust in human and how we can be sure that human interaction with the system is correct. The second one is the trust in M2M and the third one is the network system. The idea here is to check the system and give some trust level on different type of devices, connections and services. The system and the user must be sure that the deployed application it is not a threats for the environment and normal work of the other services and the local cloud.

Table of Contents

[INTRODUCTION 7](#_Toc358277089)

[1.1 Motivations 7](#_Toc358277090)

[1.2 Problems statements 9](#_Toc358277091)

[1.3 Objectives 9](#_Toc358277092)

[1.4 Scope and limits 10](#_Toc358277093)

[1.5 Organization of the Thesis 10](#_Toc358277094)

[MACHINE TO MACHINE (M2M) COMMUNICATION 11](#_Toc358277095)

[2.1 Background 11](#_Toc358277096)

[2.2 Standards Developing Organizations involved in Internet of Things/M2M standards and protocols 12](#_Toc358277097)

[2.3 Protocols 13](#_Toc358277098)

[2.3.1 MQ Telemetry Transport 13](#_Toc358277099)

[2.3.2 Advanced Message Queuing Protocol 14](#_Toc358277100)

[2.3.3 Micro M2M Data Access 14](#_Toc358277101)

[2.3.4 Supervisory Control And Data Acquisition 14](#_Toc358277102)

[2.3.5 Universal Plug and Play 14](#_Toc358277103)

[2.4 Platforms and EU projects 15](#_Toc358277104)

[2.5 Basic modules of M2M Service platform 18](#_Toc358277105)

[2.5.1 Data and device management 18](#_Toc358277106)

[2.5.2 M2M Application services 18](#_Toc358277107)

[2.5.3 Security 20](#_Toc358277108)

[2.4.3 Access Control 25](#_Toc358277109)

[2.4.4 XACML 26](#_Toc358277110)

[2.6 Tools and theory 29](#_Toc358277111)

[2.6.1 FUZZY 29](#_Toc358277112)

[2.6.2 MCDA/MAUT 29](#_Toc358277113)

[2.7 Wellness approach 31](#_Toc358277114)

[SYSTEM MODEL 34](#_Toc358277115)

[3.1 Requirements 34](#_Toc358277116)

[3.2 Clouds model 35](#_Toc358277117)

[3.2 Example scenario 35](#_Toc358277118)

[3.3 Detailed scenario 36](#_Toc358277119)

[PROPOSED METHODS 39](#_Toc358277120)

[IMPLEMENTATION AND RESULTS 43](#_Toc358277121)

[5.1 Fuzzy system for device connection 43](#_Toc358277122)

[5.1.1 Linguistic variables 43](#_Toc358277123)

[5.1.2 Membership functions 43](#_Toc358277124)

[5.1.3 Rules of the fuzzy system 45](#_Toc358277125)

[5.1.4 FIS Evaluation 46](#_Toc358277126)

[5.2 Fuzzy system for the protocols evaluation 46](#_Toc358277127)

[5.2.1 Design the inputs 46](#_Toc358277128)

[5.2.2 Membership functions 47](#_Toc358277129)

[5.2.3 Rules of the fuzzy system 49](#_Toc358277130)

[5.2.4 FIS Evaluation 49](#_Toc358277131)

[5.3 Fuzzy system for the brokers evaluation 50](#_Toc358277132)

[5.3.1 Design the inputs 50](#_Toc358277133)

[5.3.2 Membership functions 50](#_Toc358277134)

[5.3.3 Rules of the fuzzy system 51](#_Toc358277135)

[5.3.4 FIS Evaluation 51](#_Toc358277136)

[5.4 Policy model 51](#_Toc358277137)

[CONCLUSIONS 54](#_Toc358277138)

[6.1 Findings 54](#_Toc358277139)

[6.2 Future work 54](#_Toc358277140)

[REFERENCES 56](#_Toc358277141)

[APPENDIX 59](#_Toc358277142)

[The FIS editor 59](#_Toc358277143)

[The membership function editor 59](#_Toc358277144)

[The rule editor 60](#_Toc358277145)

[The rule viewer 61](#_Toc358277146)

[The surface viewer 61](#_Toc358277147)

[FIS Evaluation 62](#_Toc358277148)

[The FIS Structure 63](#_Toc358277149)

**List of tables**

[Table 1. Summary of Per Device Usage Growth, MB per Month [3] 8](#_Toc358223481)

[Table 2. The context information about each participant in the scenario 37](#_Toc358223482)

[Table 3. Range of inputs for Bluetooth 43](#_Toc358223483)

[Table 4. Range of inputs for Wireless 44](#_Toc358223484)

[Table 5. Level of Security 45](#_Toc358223485)

[Table 6. Range of inputs for Latency 47](#_Toc358223486)

[Table 7. Range of inputs for bandwidth 47](#_Toc358223487)

[Table 8. Range of inputs for performance 50](#_Toc358223488)

[Table 9. Range of inputs for security 50](#_Toc358223489)

[Table 10. Range of inputs for interoperability 51](#_Toc358223490)

**List of figures**

[Figure 1. The cross-certification trust model [29] 23](#_Toc358223513)

[Figure 2. Third-party certification model [29] 24](#_Toc358223514)

[Figure 3. Data-flow diagram 27](#_Toc358223515)

[Figure 4. Cloud scenario 35](#_Toc358223516)

[Figure 5. Simple algorithm 40](#_Toc358223517)

[Figure 6. Adaptable security algorithm / Access control design concept 41](#_Toc358223518)

[Figure 7. Membership function of bluetooth 44](#_Toc358223519)

[Figure 8. Membership function of wireless 44](#_Toc358223520)

[Figure 9. Membership function of security level 45](#_Toc358223521)

[Figure 10. Membership function of latency level 47](#_Toc358223522)

[Figure 11. Membership function of bandwidth level 48](#_Toc358223523)

[Figure 12. Membership function of scalability level 48](#_Toc358223524)

[Figure 13. Fuzzy rules for message protocols security 49](#_Toc358223525)

[Figure 14. Example of XACML Policy 53](#_Toc358223526)

[Figure 15. The FIS editor 59](#_Toc358223527)

[Figure 16. The Membership Function editor 60](#_Toc358223528)

[Figure 17. The Rule editor 60](#_Toc358223529)

[Figure 18. The Rule viewer 61](#_Toc358223530)

[Figure 19. The Surface viewer 62](#_Toc358223531)

[Figure 20. FIS structure 63](#_Toc358223532)

**List of Abbreviations**

ABAC Attribute Based Access Control

AC Access Control

CoAP Constrained Application Protocol

DAC Discretionary Access Control

H2H Human to Human

H2T Human to Thing

H2M Human to Machine

HSM Hardware Security Module

IoT Internet of Things

ITMP Identity and Trust based Model for Privacy

M2M Machine to Machines

MAC Mandatory Access Control

MAUT Multi-Attribute Utility Theory

MCDA Multi-Criteria Decision Analysis

P3P Platform for Privacy Preferences

PDP Policy Decision Point

PET Privacy Enhancing Technologies

PGP Pretty Good Privacy

PIM Privacy-enhancing Identity Management

RBAC Role Based Access Control

PRIME Privacy and Identity Management for Europe

RRIM Role- and Relationship-based Identity Management

RRIRM Role- and Relationship-based Identity and Reputation Management

SAML Security Assertion Markup Language

SSL Secure Sockets Layer

TLS Transport Layer Security

TMS Trust Management Systems

XML eXtensible Markup Language

XACML eXtensible Access Control Markup Language

**CHAPTER 1**

# INTRODUCTION

The Internet of Things (IoT) denotes the interconnection of highly heterogeneous networked entities and networks following a number of communication patterns such as: human-to-human (H2H), human-to-thing (H2T), thing-to-thing (T2T), or thing-to-things (T2Ts). The term IoT was first coined by the Auto-ID center in 1999 [1]. Since then, the development of the underlying concepts has ever increased its pace. Nowadays, the IoT presents a strong focus of research with various initiatives working on the (re)design, application, and usage of standard Internet technology in the IoT. [2]

The project focuses on security and trust issues in IoT frameworks and cloud based platforms. Security needs of machine to machine (M2M) services will be analyzed and different architectures and protocols will be compared with focus on the security part. Based on fuzzy theory, security system will evaluate the risk of used technologies and policies.

## 1.1 Motivations

Over the next 15 years, the number of machines and sensors connected to the Internet will explode. According to IMS Research, there will be more than 22 billion web-connected devices by 2020.These new devices will generate more than 2.5 quintillion bytes of new data every day.

Cisco Visual Networking Index forecast predict that mobile data traffic increase is parallel to the increase in number of devices. The new devices like tables, smartphones, small embedded devices and sensor nodes will begin to account for a more significant traffic by 2017.

Traffic growth every day with significant rates because of increased mobile devices that are manufactured. This new devices became smart and easily connected to the Internet. By forecast till 2017, there will be 8.6 billion handheld or personal mobile-ready devices and 1.7 billion machine-to-machine connections (e.g., GPS systems in cars, asset tracking systems in shipping and manufacturing sectors, or medical applications making patient records and health status more readily available, et al.).

The overall share of non-smartphones will decline from 75 percent of all mobile connections in 2012 to 50 percent in 2017. The biggest gain in share will be M2M (5 percent of all mobile connections in 2012 to 17 percent in 2017) and smartphones (16 percent of all mobile connections in 2012 to 27 percent in 2017). The highest growth will be in tablets (CAGR of 46 percent) and M2M (CAGR of 36 percent). Average traffic per device is expected to increase rapidly during the forecast period, as shown in Table 1. [3]

Table 1. Summary of Per Device Usage Growth, MB per Month [3]

|  |  |  |
| --- | --- | --- |
| **Device Type** | **2012** | **2017** |
| Nonsmartphone | 6.8 | 31 |
| M2M module | 64 | 330 |
| Smartphone | 342 | 2660 |
| 4G Smartphone | 1302 | 5114 |
| Tablet | 820 | 5387 |
| Laptop | 2503 | 5731 |

M2M technology is designed to support wired or wireless communication between machines and is used in telemetry, robotics, remote monitoring, status tracking, data collection, remote control, road traffic control, offsite diagnostics, and even in telemedicine applications.

The rapid growth of the 'Internet of Things' in industries such as home networking, medical devices, energy grid management, industrial automation, M2M, and wireless devices is increasing demand for the delivery and deployment of standard-based applications which are capable of collecting and managing data and data traffic from numerous embedded devices.

Today there are more and more intelligent devices in all business and personal domains that help us to improve productivity and to take smart decisions. The main problem is that every device has single purpose and work in isolation from the other things. Good example is the camera that is made to take pictures, but today is combined with smartphones and use their internet connection to save and share the pictures. Each of technologies takes advantage from another and extends its own functionality while reduce the cost and improve the user experience.

Today every business sector has some M2M applications that transfer the data to remote application centers and data storages for further processing. This means that their work with centralized approach and generate more and more traffic. There is also other approach like decentralized networks or local clouds [4]. The idea is to store the information closer to the devices which generate the information and aggregate the data before transfer it over the Internet. This will be good way to overcome the problem with growing traffic in mobile network.

## 1.2 Problems statements

Security and trust is important point in the future communications. Like humans trust each other when they know same language and understand themselves we must understand in details our system – what topology have, what protocols speak and what applications can be run above this system. For that purpose we will describe the whole network and focus on security aspects of the technologies. The major part of the security is access control. The idea is describe like [5]. One major problem is how to translate access rights and roles through different clouds (domains).

The authors of [6] and [7] describe existing security solutions for the Internet and give reasons why these solutions do not suit the needs of constrained networks. The required security mechanisms for the IoT can be grouped into five categories.

Strong security services can be provided within the local cloud and used by all the applications. This provides an efficient mechanism, in terms of re-use and maintainability, to enforce data integrity and privacy. Access rights can be checked only at the boundaries of the local cloud, hence limiting the overhead and keeping the system simpler, i.e. robust. Therefore, the solution will be more acceptable compared to other solutions relying on centralised data centers.

The main problem addressed in this thesis is how to develop security architecture supporting the practical security needs in m2m environment while allowing the system to stay open for new protocols, services and applications.

## 1.3 Objectives

Within the thesis project the following set of objectives was define to specify the needs and issues in communication in M2M/IoT networks.

* Study the M2M/IoT networks and the security mechanisms involved in it from the literature. This includes different type of authentication and authorization (Chapter 2).
* Analysis of a set of M2M/IoT network protocols and their security aspects with regard to their discovery and integration requirements (Chapter 2).
* Analysis of different access control mechanisms (Chapter 2).
* Design and develop an access control mechanism specific for the M2M/IoT – cloud (Chapter 3).
* Define/design an architecture of M2M/IoT scenario with wellness approach for fitness centers with local cloud platform (Chapter 4).
* Verify and validate the effectiveness of the proposed model, in terms of security and other performance metrics, e.g. delay and scalability, by means of simulation, by using security verification tools, or mathematical analysis.
* The realization of the proposed concept by demonstrating how devices can be integrated and accessed by end users.

## 1.4 Scope and limits

In the current thesis work the proposed architecture and protocols will used to define their specifications and security aspects. The limits of physical layer for example hardware security like managing digital keys, crypto processes for digital signings and for providing strong authentication to access network and applications will not be discussed. The device and infrastructure layer with all their aspects like security, routing and connectivity will use as it is. The thesis will focus on the platform and software layer. This includes the integration, middleware, APIs and applications with all their standards and protocols.

## 1.5 Organization of the Thesis

Chapter 2 surveys existing work in M2M/IoT frameworks and protocols in related areas like security, access, trust and identity that effect privacy in the fitness environment. Chapter 2 also includes M2M business sectors and focus on Healthcare and Life Science. Different types of sensors used in monitoring systems for wellness are described. In Chapter 3 include proposed architecture system model. Chapter 4 presents the access control mechanism and policies. Chapter 5 reports implementation of the proposed access mechanisms and policies in fitness scenario. Chapter 6 concludes the results and summarizes the thesis and future work.

**CHAPTER 2**

# MACHINE TO MACHINE (M2M) COMMUNICATION

## 2.1 Background

The current IoT environment is in a state of near chaotic change, with new hardware, interfaces, network access technologies, application protocols and technologies, and other individual components added or deleted quite regularly.

To be effective for delivery of services for M2M networks, traditional cloud architecture must be extended to include the local device network stack. The IoT will largely be enabled through this M2M architecture. Through the virtualization of many of the layers within the stack the cost of network will be reduced to the levels not achievable with enterprise application architecture. Device cloud extends the traditional approach to include the end device network.

As the devices rapidly growing with each year the need of stable platform that easy can manage all the devices and connections also become essential part. To address this need the middleware platform must support machine to machine functional requirements. Platform provides basic functionality required to deploy M2M service such as authentication and control of various pieces of equipment (devices), data collection and storage, security functionality.

The traditional cloud stack includes IaaS, PaaS and SaaS. It’s work almost effective for devices or M2M networks. The reason is because this architecture was not design with idea to provide M2M services. Device cloud or local cloud extends the traditional cloud stack to include this M2M services.

The local cloud includes all the gadgets, sensors and other end devices that are use the local gateway as Internet connection. This gateway can be used to collect, transform and aggregate the data before send it to the Internet. The gateway may provide other functionality to help M2M communication. The basic things in the stack are M2M devices, their connection through LAN/Mesh and last thing is the Gateway/Router with all built in functionalities.

## 2.2 Standards Developing Organizations involved in Internet of Things/M2M standards and protocols

Connecting machine to machine (M2M) and also the services that they provide to people require wide range of technologies and standards. This brings many research fields for the future Internet of Things (IoT). For that reason many organizations like ITU, ETSI, TIA start to cooperate and develop collaborative networks that will communicate even if they talk different protocols. OneM2M initiative try to provide unified definition of M2M architecture that include specifications in order to implement standard API.

**Protocols and networking standards**

IEEE (Institute of Electrical and Electronics Engineers)

* IEEE 802.11 and 802.15

IETF (Internet Engineering Task Force)

* 6LoWPAN
* CoAP (Constrained Application Protocol)

ITU (International Telecommunication Union)

* Focus group on M2M service layer

ETSI (European Telecommunications Standards Institute) - EU

* OneM2M

TIA (Telecommunications Industry Association) – USA

* OneM2M

BBF (Broadband Forum)

* TR-069 protocol specification

OMA (Open Mobile Alliance)

* OMA-DM

OASIS (Advancing Open Standards for the Information Society)

* XACML

NIST (National Institute of Standards and Technology)

* Access Control

CSA (Cloud Security Alliance)

## 2.3 Protocols

To communicate with other devices or to connect to the Internet, devices use different protocols on the different lays of the OSI model. On the physical layer they can use Bluetooth or Wifi and on the transport layer different binary or text based protocols can be used to messages transfer.Some of the psysical interfaces that small end devices have are low-energy Bluetooth, ANT, ANT+, ZigBee, ZigBee RF4CE, WiFi, Nike+, IrDA, NFC and RFID. In this chapter the focus will be on the transport layer and M2M protocols.

### 2.3.1 MQ Telemetry Transport

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

User name and password can be passed with an MQTT packet in V3.1 of the protocol. Encryption across the network can be handled with SSL, independently of the MQTT protocol itself (it is worth noting that SSL is not the lightest of protocols, and does add significant network overhead). Additional security can be added by an application encrypting data that it sends and receives, but this is not something built-in to the protocol, in order to keep it simple and lightweight.

Even Facebook engineers start using MQTT as stable and fast lightweight asynchronous messaging protocol. They explain how they will use it:

“To accomplish this we built a system of modules. Modules provide view controllers that are presented when you tap a bookmark in the left navigation menu. News Feed, Messages, Friends—they’re all modules. Modules also specify their dependencies. For example, we use MQTT to update notifications, messages, and bookmarks. At application startup, we walk the dependency graph and ensure that our MQTT service has started before we start listening for new notifications. Even as we add new features, our modular system ensures that our application setup happens in the right place, at the right time.” [8]

### 2.3.2 Advanced Message Queuing Protocol

AMQP, which stands for Advanced Message Queuing Protocol, was designed as an open replacement for existing proprietary messaging middleware. Two of the most important reasons to use AMQP are reliability and interoperability. As the name implies, it provides a wide range of features related to messaging, including reliable queuing, topic-based publish-and-subscribe messaging, flexible routing, transactions, and security.

AMQP is a binary wire protocol which was designed for interoperability between different vendors. Where other protocols have failed, AMQP adoption has been strong. Companies like JP Morgan use it to process 1 billion messages a day. NASA uses it for Nebula Cloud Computing. Google uses it for complex event processing. Here are a couple of additional AMQP examples:

* It is used in one of the world’s largest biometric databases India’s Aadhar project—home to 1.2 billion identities.
* It is used in the Ocean Observatories Initiative—an architecture that collects 8 terabytes of data per day.

### 2.3.3 Micro M2M Data Access

M3DA is a protocol optimized for the transport of binary M2M data. It is made available in the Mihini project both for means of Device Management, by easing the manipulation and synchronization of a device's data model, and for means of Asset Management, by allowing user applications to exchange typed data/commands back and forth with an M2M server, in a way that optimizes the use of bandwidth with Bysant serializer specification. [9]

### 2.3.4 Supervisory Control And Data Acquisition

SCADA systems consist of a central host or master (usually called a master station, master terminal unit or MTU), one or more field data gathering and control units or remotes (usually called remote stations, remote terminal units, or RTU’s) and a collection of standard and/or custom software used to monitor and control remotely located field data elements.

### 2.3.5 Universal Plug and Play

Set of networking protocols, mainly designed for residential networks, that enables networked devices, such as personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other’s presence on the network and to establish network services for entertainment, data sharing, and communications. The concept of UPnP is an extension of plug-and-play, a technology for dynamically attaching devices directly to a computer, although UPnP is not directly related to the earlier plug-and-play technology. UPnP devices are "plug-and-play" because when connected to a network they automatically (zero configuration) "collaborate" with other devices. On security point of view UPnP didn’t provide any mechanism for authentication and authorization. For that reason is proposed an extension of the UPnP specification called UPnP-UP [10], which allows user authentication and authorization mechanisms for UPnP devices and applications. These mechanisms provide the basis to develop customized and secure UPnP pervasive services, maintaining backward compatibility with previous versions of UPnP.

UPnP is relevant to M2M, telling apart the presentation step. UPnP put a lot of focus on video streaming, which is not so relevant to IoT, but all the mechanics involved are valid. [10]

## 2.4 Platforms and EU projects

Communication between machines, applications and users is made by M2M middleware platform. Analyze of available platforms is required to understand what are the problems and issues for the Internet of things. In “An analysis of M2M platforms: challenges and opportunities for the Internet of Things” [11] paper authors make short review of platforms and how they are connected to the devices and interact with users.

**Sen.se** is a simple IoT/M2M platform that bases its behavior in a three-step configuration process (channels, applications and visualization). [12]

**EVRYTHNG** is a social platform with the aim of creating a unique Active Digital Identity (ADI) profile for any physical thing, giving it global access using a unique URI and APIs for that individual object, making it visible, accessible and controllable trough the global network. An ADI is simply a Web resource with information about a thing in the form of dynamic or static attributes. [13]

**AMEE** focus its services in offering a platform as a service solution focused on innovation for environmental data. AMEE’s Platform handles the infrastructure to reduce costs and accelerate time-to-market processes. It enables an easily accessible and manageable platform with an Appkit to quickly build apps by the customer, offering also services for developing applications by AMEE. It is scalable and secure, providing a complete enterprise set of services based in those precepts. One of the main points of AMEE is that the platform is open source, built on a RESTful API in order to harness collaboration. [14]

**RunMyProcess** platform allows its customers to design and run business 'processes'. These processes can interact with users and/or other 'web services'. The platform conformed by an on-demand infrastructure which relies on a centric application platform enabling the development and deployment of applications simply dragging and dropping function boxes and assigning one of the predefined functions or defining a new one by the developer. This platform runs over an Amazon Web Service infrastructure, which means several replicated centers around the world, secured in order to prevent unauthorized access. Additionally, several authentication methods are supported such as Microsoft Azure, or Google 2-legged Open Authorization. [15]

**The Axeda Platform** is a complete M2M data integration and application development platform with infrastructure delivered as a cloud-based service. It is aware of the scalability and security needs, at the same time that offers a powerful development environment with flexible APIs, easing to build and deliver custom M2M applications for the most demanding requirements and integrate M2M data into enterprise applications and systems. [16]

**The ThingWorx platform** bases its operation model in treating all things (considering people, physical world and systems) at the same level. This enables to create processes connecting things in any possible combination. The platform stores information about this people, environment and systems, creating applications that evolve and grow together. On this way, applying the network effect to these applications produces a multiplier effect over data that enhances its value. ThingWorx enables a new type of transformational applications as they continuously evolve and increase in value over time, and allow users to answer questions, solve problems, and capture opportunities that have not been anticipated. [17]

To improve this analysis we include Eclipse Mihini platform and other European projects with focus on IoT, M2M communications and Clouds.

**Eclipse Mihini platform** is open source project just released in February 2013. Its begin August 2012 with the idea to provides low-level connectivity management to ensure that a reliable network connection is available to business applications. It’s also acts as an abstraction layer for underlying hardware and enables smart business data transmission between devices and servers, including the ability to consolidate data locally and use bandwidth-efficient communication protocols. Major focuses on the projects are MQTT protocol broker and just start to implement M3DA broker. [18]

**EU Projects**

**Internet of Things Architecture**, the European Lighthouse Integrated Project addressing the Internet-of-Things Architecture, proposes the creation of an architectural reference model together with the definition of an initial set of key building blocks. Together they are envisioned as crucial foundations for fostering a future Internet of Things. Using an experimental paradigm, IoT-A will combine top-down reasoning about architectural principles and design guidelines with simulation and prototyping to explore the technical consequences of architectural design choices. [19]

**OpenMTC platform** is to provide a standard compliant middleware platform for M2M oriented applications and services. While supporting application domain driven scenarios such as eHealth and Smart City services, OpenMTC will rely on advanced networking capabilities provided by our highly successful 3GPP Evolved Packet Core (EPC) implementation. [20]

**BETaaS Platform** (Building the Environment for the Things as a Service) propose a platform for the execution of M2M applications, which is built on top of services deployed in a “local cloud” of gateways, the latter being the devices which provide the smart things with connectivity to the Internet (e.g., smart phones, home routers, road-side units). Adaptation layers will be defined to interconnect BETaaS with the main architectures proposed at a European level for M2M communication, including ETSI M2M and IoT-A. [4]

**PrimeLife** (Privacy and Identity Management Europe) will resolve the core privacy and trust issues pertaining to these challenges. Its long-term vision is to counter the trend to life-long personal data trails without compromising on functionality. We will build upon and expand the sound foundation of the FP6 project PRIME that has shown privacy technologies can enable citizens to execute their legal rights to control personal information in on-line transactions. [21]

**OpenIoT** (Open source solution for the Internet of things into the cloud) is perceived as a natural extension to cloud computing implementations, which will allow access to additional and increasingly important IoT based resources and capabilities. In particular, OpenIoT will research and provide the means for formulating and managing environments comprising IoT resources, which can deliver on-demand utility IoT services such as sensing as a service as an example. [22]

## 2.5 Basic modules of M2M Service platform

Based on all platforms and M2M middleware described previously in the thesis, four basic functionalities can be notice in all platforms:

* **Data and device management** – process and store incoming M2M data (data gathering and storage function), data analysis and statistics functions.
* **M2M application services** – Allow developers to extend and customize the core platform functionality via powerful embedded scripting engine and a rich set of WebServices for both SOAP and REST consumption.
* **Service integration framework** – Accelerates integration with the Platform and enterprise systems including ERP (Enterprise Resource Planning), CRM, and almost every billing and data warehouse with standards-based message queue technology.
* **Security function** – built-in security for managing users, roles, user groups and device groups, device authentication and control function.

### 2.5.1 Data and device management

The problem with managing gateways, routers, devices and sensors become essential when the number of devices increase and also the geographical distance between them become time and money consuming. Management system must provide maintenance of network assets and devices over the network.

To manage devices and things can be really complex and difficult job. The typical approach for management is remote access and control of devices. However, even with that type of management is not suitable for growing IoT. The best way is to integrate the management capability into the architecture when is design from scratch.

### 2.5.2 M2M Application services

To provide application functionality, M2M service capable router or middleware must be able to make service discovery and service location. There are many standards and protocols developed for computer networks and they have some advantages and disadvantages when we talk about M2M networks.

#### 2.5.2.1 Service discovery

Service discovery functions allow computers and other devices easily to find in one IoT environment what is around them and how they can use it without any configuration. There are a few protocols developed for that purpose. Zeroconf networking allows servers and clients on an IP network to exchange their location and access details around the LAN without requiring any central configuration.

**Avahi** is an Implementation of the DNS Service Discovery and Multicast DNS specifications for Zeroconf Networking. It uses D-Bus for communication between user applications and a system daemon. The daemon is used to coordinate application efforts in caching replies, necessary to minimize the traffic imposed on networks. [23]

**Bonjour** is Apple's implementation of Zero configuration networking (Zeroconf), a group of technologies that includes service discovery, address assignment, and hostname resolution. Bonjour locates devices such as printers, other computers, and the services that those devices offer on a local network using multicast Domain Name System (mDNS) service records. [24]

**Universal Plug and Play (UPnP)** is a set of networking protocols that permits networked devices, such as personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing, communications, and entertainment. UPnP is intended primarily for residential networks without enterprise class devices.

The UPnP protocol, as default, does not implement any authentication, so UPnP device implementations must implement their own authentication mechanisms, or implement the Device Security Service. There also exists a non-standard solution called UPnP-UP (Universal Plug and Play - User Profile) which proposes an extension to allow user authentication and authorization mechanisms for UPnP devices and applications.

Unfortunately, many UPnP device implementations lack authentication mechanisms, and by default assume local systems and their users are completely trustworthy. DLNA-compatible devices use UPnP to communicate, and there are three classes of DLNA devices: Home Network Devices, Mobile Handheld Devices and Home Infrastructure Devices. The first category encompasses media servers, AV receivers, TVs, consoles and tablets; the second category includes smartphones and media tablets; and the third category covers routers and hubs. [25]

#### 2.5.2.2 Service location

The Service Location Protocol (SLP) allows computers and other devices to find services in a local area network without prior configuration. SLP has been designed to scale from small, unmanaged networks to large enterprise networks. [26] [27]

SLP has three different roles for devices. A device can also have two or all three roles at the same time.

* User Agents (UA) are devices that search for services;
* Service Agents (SA) are devices that announce one or more services;
* Directory Agents (DA) are devices that cache services. They are used in larger networks to reduce the amount of traffic and allow SLP to scale. The existence of DAs in a network is optional, but if a DA is present, UAs and SAs are required to use it instead of communicating directly.

Today most implementations are daemons that can act both as UA and SA. Usually they can be configured to become a DA as well.

SLP contains a public-key cryptography based security mechanism that allows signing of service announcements. In practice it is rarely used:

* The public keys of every service provider must be installed on every UA. This requirement defeats the original purpose of SLP, being able to locate services without prior configuration.
* Protecting only the services is not enough. Service URLs contain host names or IP addresses, and in a local network it is almost impossible to prevent IP or DNS spoofing. Thus only guaranteeing the authenticity of the URL is not enough if any device can respond to the address.
* As addresses can be spoofed, the authenticity of the device must be proven at a different level anyway, e.g. in the application protocol (e.g. with SSL) or in the packet layer (IPsec). Doing it additionally in SLP does not provide much additional security.

### 2.5.3 Security

Security is an important part of Internet and M2M systems. Trust in the system from different stakeholders is one of the key concepts. They must be sure that their own assets are protected. For example in some service sectors like health there are different legal and regulation requirements for data protection depending on country or medical area. This different requirement makes the security a tough task.

**Security for the hardware**

A hardware security module (HSM) is targeted at managing digital keys, accelerating crypto processes in terms of digital signings/second and for providing strong authentication to access critical keys for server applications. These modules are physical devices that traditionally come in the form of a plug-in card or an external TCP/IP security device that can be attached directly to the server or general purpose computer.

The goals of an HSM are:

* Onboard secure generation;
* Onboard secure storage;
* Use of cryptographic and sensitive data material;
* Offloading application servers for complete asymmetric and symmetric cryptography.

**Security for the session layer**

At the session layer of the OSI (Open Systems Interconnection) stack both SSL (Secure Socket Layer) or TLS (Transport Layer Security) can be used. The SSL was originally developed by Netscape Communications Corporation to provide privacy and reliability between two communicating applications at the Internet session layer. SSL uses public-key encryption to exchange a session key between the client and the server. This session key is used to encrypt the HTTP transaction. Each transaction uses a different session key. Even if someone manages to decrypt a transaction the session itself is still secure ( just the one transaction is violated). In the past encryption made use of a 40-bit (rest of the world) or 128-bit (USA) secret key, but the situation changes as export restrictions are relaxed.

**Security for application layer**

Higher layer security systems have different technology to protect the privacy of the data and applications. Good example for this type of security technics is PGP. Pretty Good Privacy (PGP) use IDEA encryption, RSA key management and digital signatures. Data integrity is protected by the MD5 algorithm. Application security is really important and we can consider it like entry point of the system. For that reason a lot of threats and attacks are focused on the application layer.

#### 2.5.3.1 Privacy

Privacy is one of key concepts now days. Everyone is afraid for his personal data that is on the internet or enterprise companies try to protect their entire infrastructure. That’s why they have own mail servers, data storages and etc. Privacy can be divided on few categories that have technical aspects:

* Communication privacy
* Position privacy (Location privacy)
* Path privacy
* Identity privacy (Personal privacy)
* Local information (use crypto for data protection)

Sticky policies are a way to cryptographically associate policies to encrypted (personal) data. These policies function as a gate keeper to the data. The data is only accessible when the stated policy is honored. System keeps track of personal data relating to the user, as well as applied policies and service customizations. [28]

#### 2.5.3.2 Authentication

Most common method for authentication is to provide username and password. Another method for authentication is SSO (Single Sign-on), which help to reduced sign-on and avoid continually re-authenticating for each application. (Example HomeCloud/Enterprise)

In computer security, access control includes, among other features, the authentication and the authorization mechanisms. Identification and authentication are the processes of checking something (or someone) as authentic. In short, authentication is the basic building block of security.

User identification and authentication in pervasive environments are also important due to the range of devices and services to which users have access.

#### 2.5.3.3 Trust

Trust is the main concern of consumers and service providers in a cloud computing environment. The different local systems and users of diverse environments brings special challenges to the security of cloud computing. In trust we can consider QoS, key management systems, lightweight PKI certification concept and decentralized system for establishing the trust, which must be alternative to PKI. For M2M/IoT systems we need novel method to establish trust in people, devices and data beyond the today’s reputation systems.

**Cross-certification trust model**

n this model, each organization must individually certify that every other par- ticipating organization is worthy of its trust. The organizations review each other’s processes and standards and their due diligence efforts determine whether the other organizations meet or exceed their own standards. Once this verification and certi- fication process is complete the organizations can then begin to trust other organi- zations’ users. The example of cross-certification model is shown on **Figure 1**.

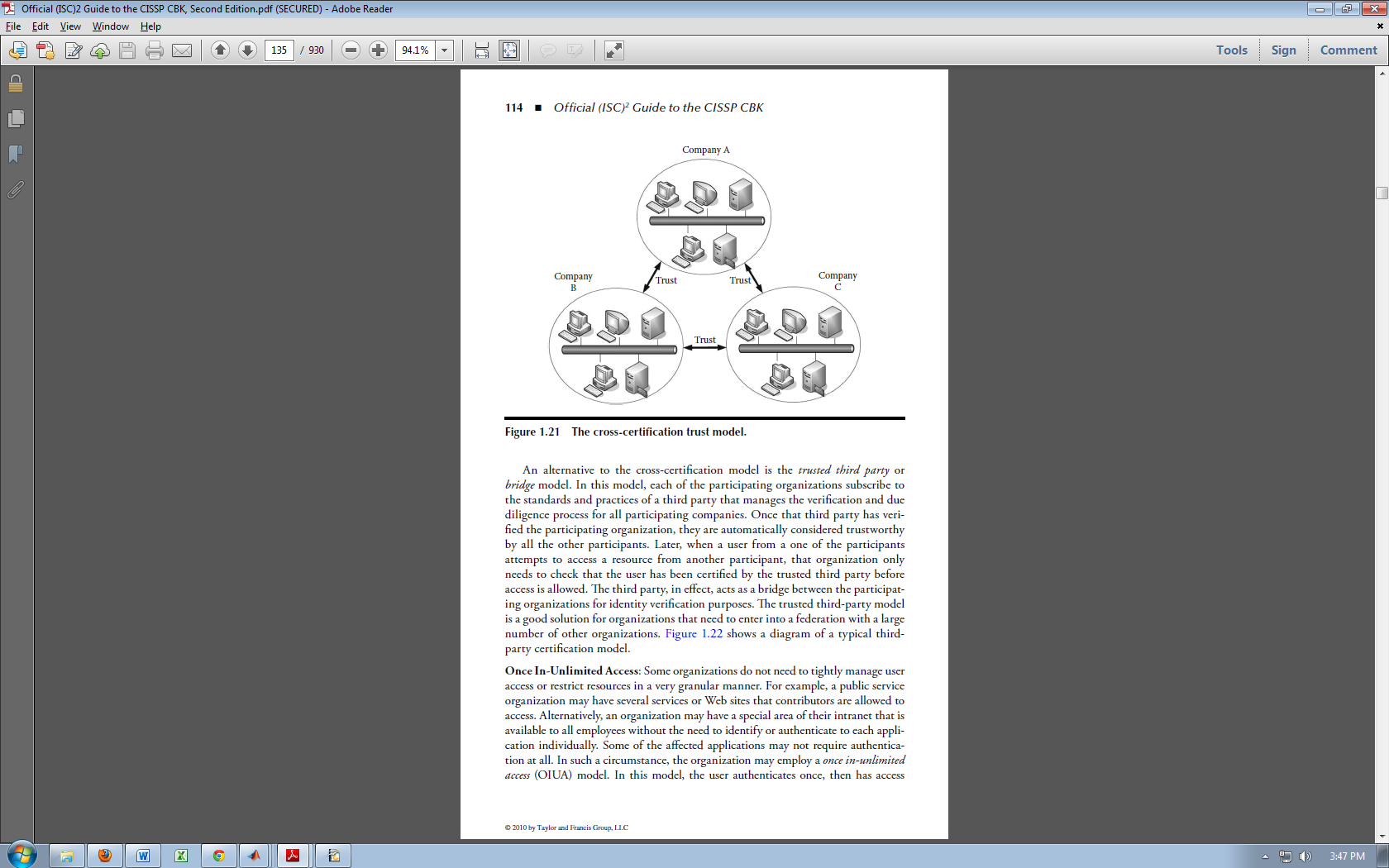


Figure 1. The cross-certification trust model [29]

The issue with cross-certification trust model is that when the number of participating cloud grows, the numbers of trust relationships grows also.

**Third-party bridge trust model**

The way to overcome that problem is to use trusred third party or bridge model shown on figure x. In this model, each of the participating organizations subscribe to the standards and practices of a third party that manages the verification and due diligence process for all participating companies. Once that third party has verified the participating organization, they are automatically considered trustworthy by all the other participants. Later, when a user from a one of the participants attempts to access a resource from another participant, that organization only needs to check that the user has been certified by the trusted third party before access is allowed. [29] Figure 1.21 shows a graphical representation of a cross-certification trust model.

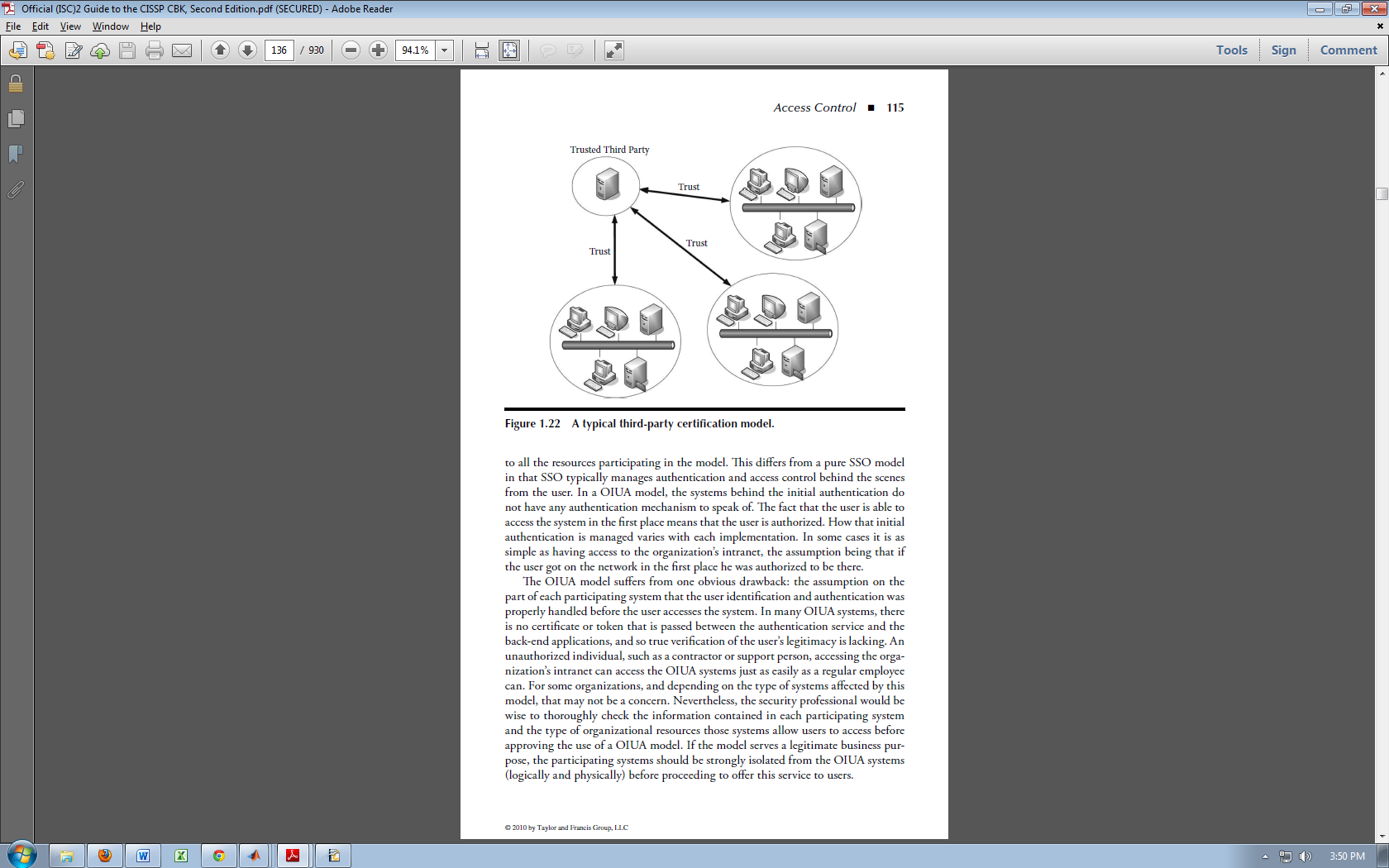


Figure 2. Third-party certification model [29]

**Trust by means of Reputation**

Different models have been proposed to fix the trust issues in cloud computing and exchanging private data between users. The most common used is Reputation model (e.g. Amazon, E-bay, Mac App Store, Google play Android Apps). In this examples the reputation and trust in application is based on ranking of the other users. The problem with this kind of system is that the reputation score is based on past behavior of the customers and service providers. When one service provider with good reputation start to receive negative rates from his customers there is some jitter to his current rate. It will take some time to gather more negative feedback so other users can obtain correct information. Other problem is when new company start to sell some service and didn't have any past reputation feedback How we can know is this company providing secure services or not? All this examples use centralized architecture of service discovery and the reputation information has a single point of failure.

Peer-to-peer web service discovery that uses QoS and users’ feedback to rank and select services was proposed in "Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings" [30]. QoS data about services and reputation rates from consumers are stored in multi-peers in peer-to-peer systems. Monitoring agents are used to prevent cheating by users and providers. Trusted agents monitor and provide reports of services to a UDDI peer and, based on this information, services are evaluated and ranked. However, the monitoring of reports differs from peer to peer, because each peer uses different criteria to provide feedback about services.

Trust management in distributed systems like P2P and mobile ad hoc networks is still big issue. Centralised approach for trust system will be not effective and scalable. The broker framework [31] or third parties trust model are more proper choice for peer to peer networks.

### 2.4.3 Access Control

Computer security architects and administrators deploy access control mechanisms (ACM) in logic aligned to protect their objects by mediating requests from subjects. These ACMs can use a variety of methods to enforce the access control policy that applies to those objects.

An access control policy simply states, “Who can do what to what”. [5] The assumption that access control is always (human) user-based does not hold any longer in many environments like Machine to Machine and Internet of Things. Access control may need to be machine-to-machine or application-to-application-based, and may only be easily enforceable if it is expressed with the protected resource in mind (“what is allowed on this system”) rather than user-centric (“what user xyz is allowed to do”).

These access control models provide a framework and set of boundary conditions upon which the objects, subjects, operations, and rules may be combined to generate and enforce an access control decision. Each model has its own advantages and limitations. The major types of data access control are:

* MAC – Mandatory access control
* DAC – Discretionary access control
* RBAC – Role-Based access control
* ABAC – Attribute-Based access control
* CBAC – Context-Based access control
* PBAC – Policy-based access control
* CCAAC – Capability-based Context Aware Access Control model

**Use and availability**

The use of RBAC to manage user privileges (computer permissions) within a single system or application is widely accepted as a best practice. Systems including Microsoft Active Directory, Microsoft SQL Server, SELinux, grsecurity, FreeBSD, Solaris, Oracle DBMS, PostgreSQL 8.1, SAP R/3, ISIS Papyrus, FusionForge and many others effectively implement some form of RBAC. A 2010 report prepared for NIST by the Research Triangle Institute analyzed the economic value of RBAC for enterprises, and estimated benefits per employee from reduced employee downtime, more efficient provisioning, and more efficient access control policy administration. [32]

In an organization with a heterogeneous IT infrastructure and requirements that span dozens or hundreds of systems and applications, using RBAC to manage sufficient roles and assign adequate role memberships becomes extremely complex without hierarchical creation of roles and privilege assignments. Newer systems extend the older NIST RBAC model to address the limitations of RBAC for enterprise-wide deployments. The NIST model was adopted as a standard by INCITS as ANSI/INCITS 359-2004. A discussion of some of the design choices for the NIST model has also been published. [33]

### 2.4.4 XACML

The eXtensible Access Control Markup Language (XACML) is an access control policy speciﬁcation language created by the OASIS committee [34].

**Data-ﬂow model of XACML**

An access control system using XACML as its policy speciﬁcation language is meant to be used on the Internet, where a diﬀerent component of the system locates throughout the network. The data-ﬂow model which describes how information is exchanged between the components is shown in **Figure 3**.

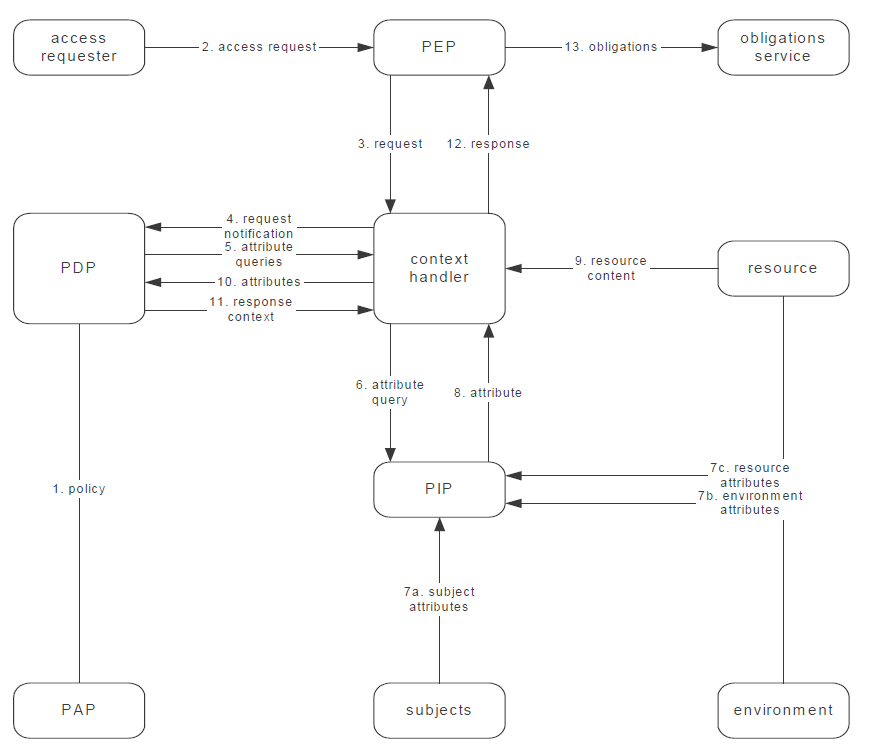


Figure 3. Data-flow diagram

Access control policies written in XACML are stored in the policy administration point (PAP). This PAP is known to the policy decision point (PDP), which is the entity that makes access decisions. The policy enforcement point (PEP) is the entity which implements and enforces mechanisms of access control. When it receives a request, it passes the request to the context handler. The context handler then assembles the request into a format speciﬁed by XACML and passes it to the PDP. On receiving the request, the PDP searches through the policies provided by the PAP and picks up an applicable policy, if there is one, and makes a decision based on the policy and the content of the request. To make the decision, the PDP may need to consult the context handler to ﬁnd out values of certain attributes which are necessary to make the decision. The context handler will gather all that information from diﬀerent sources, such as from the policy information point (PIP), from the environment, from the subjects and resource. Once a decision is made, the PDP will send it back to the context handler, who will transform the response into a format understandable to the PEP and forward it to the PEP.

**Rule, policy and policy-set**

The most basic functional unit in XACML is a rule. A number of rules form a policy. A number of policies form a policy set. A complete rule consists of a head, a description, a target and a condition. The head contains a XML name space declaration, a name for the rule, and the eﬀect of the rule, either Deny or Permit. The description describes the rule in human languages, and thus makes the rule more understandable. The target deﬁnes applicable situations for the rule. If the target is evaluated to false, the rule will be simply rendered as not applicable and the condition will not be considered. The condition represents a boolean expression, just as the target, which reﬁnes the applicability of the rule. Only if the target and the condition are both evaluated to true, is the eﬀect of the rule returned. Otherwise this rule is reckoned as not applicable. The structure of a policy is very much like that of a rule. It contains a head, a description about the policy, a target deﬁning the applicability of the policy, and a number of rules. However, in the policy, a rule-combining algorithm must be speciﬁed to resolve conﬂicting results returned by diﬀerent applicable rules. For example, if the deny-overrides algorithm is used, the eﬀect is that if any rule is evaluated to Deny, the policy must return Deny. The rule-combining algorithm is speciﬁed in the head of the policy. Likewise, the structure of a policy set is like that of a policy, except that a policy set uses a policy-combining algorithm

## 2.6 Tools and theory

To develop system with heterogeneous devices, protocols, middleware, services and applications that support adaptable security is really hard task. To overcome this problem of wide security aspects fuzzy theory and multi-attribute utility theory are selected to be core engine of proposed algorithm.

### 2.6.1 FUZZY

Fuzzy logic is widely used in many security and network systems where is need some type of decisions. Fuzzy Logic inference system is used in proposal of a power-efficient secure routing protocol for wireless sensor networks [35], in the designing an Expert System for Cyber Security [36], Expert Systems in Network Design [37] and many other.

The most important two types of fuzzy inference method are Mamdani’s fuzzy inference method, which is the most commonly seen inference method. This method was introduced by Mamdani and Assilian (1975). Another well-known inference method is the so-called Sugeno or Takagi–Sugeno–Kang method of fuzzy inference process. This method was introduced by Sugeno (1985). [37]

Advantages of the Sugeno Method

• It is computationally efficient.

• It works well with linear techniques (e.g., PID control).

• It works well with optimization and adaptive techniques.

• It has guaranteed continuity of the output surface.

• It is well suited to mathematical analysis.

Advantages of the Mamdani Method

• It is intuitive.

• It has widespread acceptance.

• It is well suited to human input.

Fuzzy inference system is the most important modeling tool based on fuzzy set theory. The FISs are built by domain experts and are used in automatic control, decision analysis, and various other expert systems.

### 2.6.2 MCDA/MAUT

Multi-Criteria Decision Analysis (MCDA) methods utilize a decision matrix to provide a systematic analytical approach for integrating risk levels, uncertainty, and valuation, which enables evaluation and ranking of many alternatives. MCDA overcomes the limitations of less structured methods such as comparative risk assessment (CRA), which suffers from the unclear way in which it combines performance on criteria. [38]

Multi-Attribute Utility Theory is a systematic method that identifies and analyzes multiple variables in order to provide a common basis for arriving at a decision. As a decision making tool to predict security levels depending on the security context (network state, the resource's and user's environments, etc.), MAUT suggests how a decision maker should think systematically about identifying and structuring objectives, about vexing value tradeoffs, and about balancing various risks. The decision maker assigns utility values to consequences associated with the paths through the decision tree. This measurement not only reflects the decision maker's ordinal rankings for different consequences, but also indicates her relative preferences for lotteries over these consequences [39].

According to MAUT, the overall evaluation v(x) of an object x is defined as a weighted addition of its evaluation with respect to its relevant value dimensions [40]. The common denominator of all these dimensions is the utility for the evaluator [41].

The utility quantifies the personal degree of satisfaction of an outcome. The MAUT algorithm allows us to maximize the expected utility in order to become the appropriate criterion for the decision maker's optimal action.

Security management system that dynamically adapts the security level according to a set of contextual information such as terminal types, service types, network types, user's preferences, information sensitivity, user's role, location, time, using MAUT (Multi-Attribute Utility Theory) in order to support secure transactions in the heterogeneous network.

The security research community is hardly working on these problems, and most efforts are directed towards developing strongest cryptographic protocols and more effective authentication methods.

## 2.7 Wellness approach

All new technologies and services are made to provide the people better life. One of the everyday people problems is health and wellness. With the help of science and modern technologies we can bring better and healthy life. Wellness approach is key concept of monitoring everyday activities and vital sign like weight, exercise, sleep and cardiac health.

The growing end devices and open source projects will help a lot for future Internet of Things. On the market there a lot of small embedded devices that can collect and transmit different data. Open source communities can provide the implementations and real data to the organization that write standards and specification for this wide range of technologies.

Example of this open source projects is e-Health Sensor Platform and Waspmote Wireless Sensor Platform [43]. This small developments kit that include sensor shield and API with most of the communication protocol libraries are really good choice for education and research propose. They can provide entry point of the data for M2M commutations.

On the market there are also ready for use solutions like Withings WiFi Body Scale, Withings Smart Blood Pressure Monitor, BodyMedia FIT armbands, Zeo Personal Sleep Coach, and Runkeeper [44].

**mHealth sensors**

On the market are different types of sensors that help users to perform biometric and medical measurements. The major one for body monitoring are pulse, oxygen in blood (SPO2), airflow (breathing), body temperature, electrocardiogram (ECG), glucometer, galvanic skin response (GSR - sweating), blood pressure (sphygmomanometer) and patient position (accelerometer).

This sensors can be connected to different types of micro-controllers and can send biometric information wirelessly by Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 and ZigBee depending on the application.

This information can be used for medical diagnosis of users and also to monitor their state in real time.

**SPO2 sensor**

Pulse oximetry a noninvasive method of indicating the arterial oxygen saturation of functional hemoglobin. Oxygen saturation is defined as the measurement of the amount of oxygen dissolved in blood.

**ECG sensor**

The electrocardiogram (ECG or EKG) is a diagnostic tool that is routinely used to assess the electrical and muscular functions of the heard.

**Airflow sensor**

Abnormal respiratory rates and changes in respiratory rate are a broad indicator of major physiological instability, and in many cases, respiratory rate is one of the earliest indicators of this instability. AirFlow sensor can provide an early warning of hypoxemia and apnea.

**Temperature sensor**

Body temperature depends upon the place in the body at which the measurement is made, and the time of day and level of activity of the person. Different parts of the body have different temperatures.

**Blood pressure sensor**

Blood pressure is the pressure of the blood in the arteries as it is pumped around the body by the heart. When your heart beats, it contracts and pushes blood through the arteries to the rest of your body. This force creates pressure on the arteries. Blood pressure is recorded as two numbers—the systolic pressure (as the heart beats) over the diastolic pressure (as the heart relaxes between beats).

**Position sensor**

The Patient Position Sensor (Accelerometer) monitors five different patient positions (standing/sitting, supine, prone, left and right.) In many cases, it is necessary to monitor the body positions and movements made because of their relationships to particular diseases (i.e., sleep apnea and restless legs syndrome).

**GSR sensor**

Skin conductance, also known as galvanic skin response (GSR) is a method of measuring the electrical conductance of the skin, which varies with its moisture level. This is of interest because the sweat glands are controlled by the sympathetic nervous system, so moments of strong emotion, change the electrical resistance of the skin.

**Glucometer sensor**

Glucometer is a medical device for determining the approximate concentration of glucose in the blood. A small drop of blood, obtained by pricking the skin with a lancet, is placed on a disposable test strip that the meter reads and uses to calculate the blood glucose level.

**Summary**

This chapter provides a general overview of all technologies that are important for creating one M2M communication. In the commercial market some of the products like gateways start to support M2M communication technologies. Example for that are Cisco ISR routers [45], deviceWISE routers [46]. New companies and open-source groups start to develop their M2M platforms that will fill the gap between the devices and the human interaction. With all vast of middleware still there are not so well develop peer-to-peer communications. The BETaaS approach for local clouds and also Qualcomm AllJoyn framework for mobile devices only focus on the ad-hoc and peer-to-peer communications. All this gateways and frameworks still didn’t combine the device network requirements on one hand and on another the end users and business sector vision. All the activities and with help of collaboration work of the standards developing organizations in few years we will have generalized framework for m2m communications that will support many heterogeneous devices.

**CHAPTER 3**

# SYSTEM MODEL

In this chapter system model will be described in details. Each part of the system has different security aspects. Requirements for one M2M/IoT platform and communication of one local cloud with other systems and clouds also have specific issues.

## 3.1 Requirements

The conditions for one proper working M2M system are countless. For that reason is good to separate them on different categories.

Basic requirements:

• Cross-vertical communication between M2M applications;

• Lightweight solutions for low-power end devices;

• Scalable to billions of devices.

Interoperability requirements:

• Heterogeneity of devices and platforms;

• Simple integration with other systems and interfaces.

Security requirements:

• For distributed clouds all the policy must be built at the local cloud;

• Trust between clouds.

## 3.2 Clouds model

In the distributed cloud system we have small networks that operated over gateways and coordinators. The following scenario that is show on **Error! Reference source not found.** we have three different local clouds.

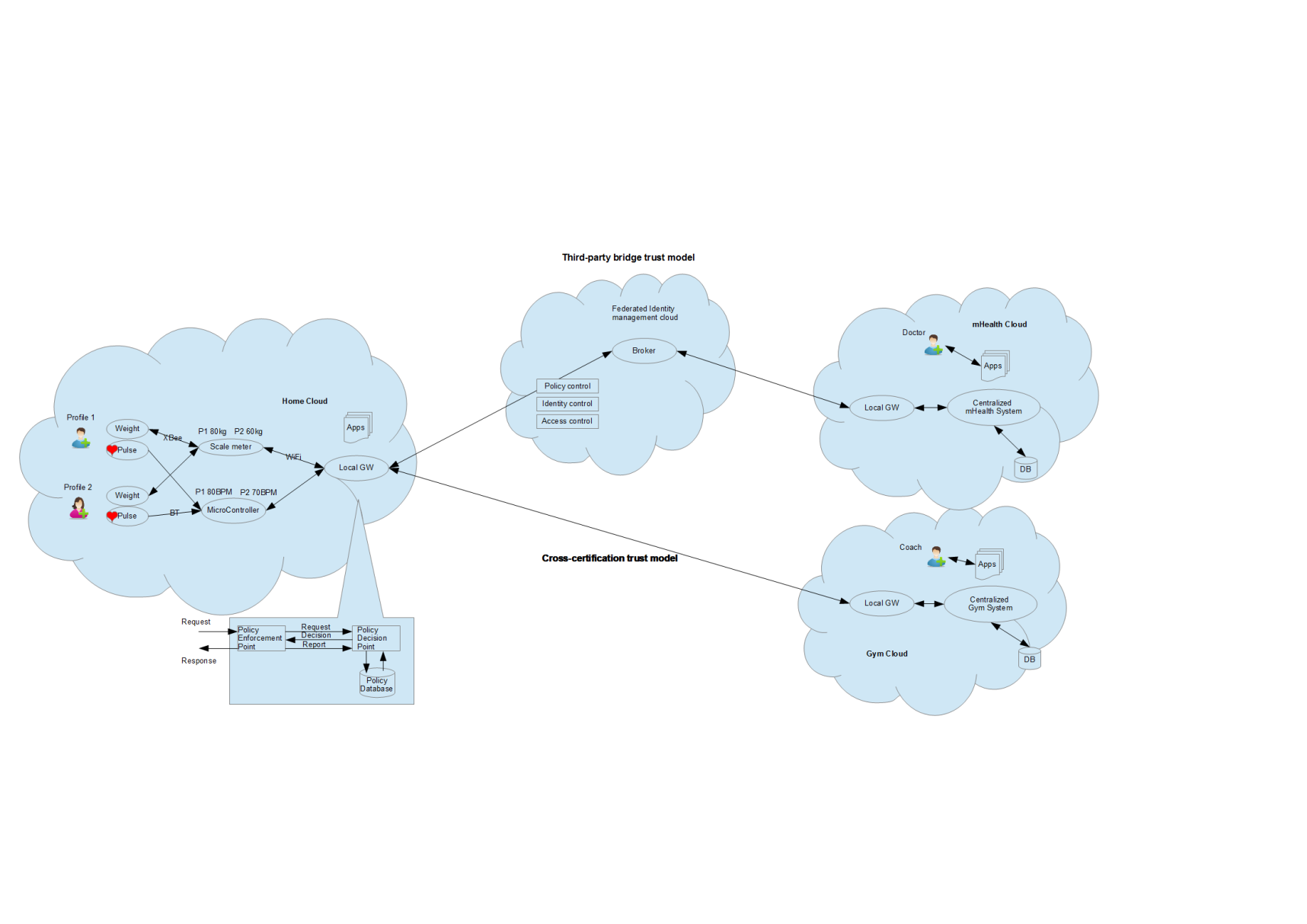


Figure 4. Cloud scenario

The idea of the local cloud is to have all things that are required for M2M/IoT environment. This can include many nodes, micro-controllers, embedded devices, smart meters, sensors and actors. Everything needs to communicate with the coordinators if there is a mesh network. After coordinators has the local gateway, which can connect to the Internet or to another distributed cloud. On the top of the local gateway is running middleware software that is capable to collect the data from sensors and execute M2M applications. It includes also policy module that combine the policy decision and enforcement points in the cloud. The policies are required when the user want to share the data within the local cloud with other users or to send it to the Internet.

The communication between different clouds must be secured and the users must have trust in the destination cloud which will process their data. This can happen with the help of certificates directly between parties or with help of third party that will provide the trust of each cloud. The example scenario will show some of the problems and issues with the security and trust in clouds.

## 3.2 Example scenario

This section presents an example scenario of use the distributed cloud system. Proposed system have three different clouds – Home, mHealth(critical) and Gym(non-critical). In Home cloud we have two actors Alice and Bob. In mHealth cloud we have doctor Charlie. Charlie wants to deploy new medical application to monitoring continuous heart rate of Bob. The main idea is that data from pulse sensor is stored locally and Bob is owner of it. The medical application that Charlie deploys will take the data and will transform it into right syntax (correct format and values) that will fit the medical records and will transfer important information to the hospital.

The problem here is the security threats from both sides. How can Bob be sure that Charlie Application will access only pulse rate data from his cloud and on the other hand how Charlie will be sure that Bob didn't modify the data that will be sent to hospital and it will be with acquired format and associated with Bob profile.

Use case 2: Gym Coach Dave want to access the pulse sensor information in same time with doctor Charlie and have low priority.

Use case 3: Alice and Bob go to gym and use the scale meter. They want to collect the data and transfer it to their home cloud. The gym want to have information from the scale meter but in term of privacy they must have only value and no profile information. How we can achieve this goal?

Trust in this use case scenario has two main problems – trust in the human and in the machines. Some questions can help to determine this trust.

To determine trust in human part of the system few questions can be focused for the doctor:

* From where is he trying to access out network;
* What time is it, when he want to access the network;
* What app he want to deploy;
* Is he access the network for first time.

On the other hand we have trust in the machines and for example how the doctor can trust in our system. Here are the questions that help to determine the trust into the system:

* What is the latency;
* What is the performance;
* What is the availability;
* Is it scalable, can he extend the system.

## 3.3 Detailed scenario

Fitness goals will be more easily achievable with the help of IoT systems. Thanks to compatible devices and information systems, individuals can maximize the effectiveness of their fitness programs. They can track their progress and share workouts results with a trainer, who can provide feedback, or with a friend who can help them stay motivated [47].

Gym use case:

1. Alice and Bob go to gym

2. They bring their mobile phones with lightweight BETaaS with profile information

3. Alice and Bob use the gym treadmill (scale meter), which is also BETaaS enabled

4. Phones make device discovery (service discovery also)

5. Equipment finds their phones

6. How they can trust each other?

7. Trust evolution (User define parameters)

7.1. Trust in devices:

• Wireless connection encryption;

7.2. Trust in the service:

• Put the service in the sandbox;

7.3. Trust in the application:

• Sure that application will not harm other services;

Scenario for each phase:

The users go to local gym and try to use new BETaaS middleware. They must be authenticated to access the network and to get authorised for M2M services that gym provides via his or her smart device (phone or tablet). Then the security module in BETaaS middleware will load the zero configurations. The details information about the scenario is displayed in Table 2.

Table 2. The context information about each participant in the scenario

|  |  |
| --- | --- |
| User | Context information |
| Alice | UserID: Alice  Connection type: Bluetooth v2.1 SSP  Time: 10:00:00  SmartPhone UserAgent:   * Operating System: iOS 4.0 * BETaaS version: 1.0 |
| Bob | UserID: Bob  Connection type: Bluetooth v4 SecurityManager service with AES encryption.  Time: 12:00:00  Tablet UserAgent:   * Operating System: Android 4.0 * BETaaS version: 1.0 |
| Treadmill(machine) | UserID: Treadmill#11  Connection type: WiFi (WPA2)  Message protocol: MQTT  Time: 08:00:00  Machine UserAgent:   * Service discovery; * BETaaS version: 1.0 |
| Treadmill(machine) | UserID: Treadmill#12  Connection type: WiFi (WPA2)  Message protocol: AMQP  Time: 08:00:00  Machine UserAgent:   * Service discovery; * BETaaS version: 1.0 |

**CHAPTER 4**

# PROPOSED METHODS

In this Section security level is determined by proposed algorithm and correct access control mechanism is select for the scenario. The main idea is how to determine security level of the local cloud without help of third-party management, because that is one of the main problems of Internet of Things.

First part of understanding the topology of the networks is to determine actors and what type of connections are used. Basically systems can be dividing to two types – Machine to Machine (M2M) communication and Human to Machine (H2M) communication. It’s important to detect what is the information flow, because different access control methods can be used. Figure 5 shows the algorithms in which we have user that launch application on the user device. This user device can be smart phone, tablet, laptop or any device that has computing capabilities. Next step is to connect the user device within the local network through the gateway. On the other flow is machine to machine communication. The main component here is machine device that is capable to provide some service. This can be any smart router or setup box that is connected to the local cloud through the gateway. This machine device is also connected to the small low-power sensors, smart meters or actors that provide the data required to create service. And the final flow is between the user application and machine device that provide the specific service. For example this data can be generated from any temperature sensor, pulse meter, light switch and etc.

****

Figure 5. Simple algorithm

The final flow is between the user application and machine device that provide the specific service. Between the different parts of the heterogeneous network are used many protocols, brokers, services and applications. This variety of security requirements need to extend the simple algorithm and to make it adaptable with the help of fuzzy theory. Detailed algorithm is shown on Figure 6.

This algorithm displays the authentication and authorization stages from the application and end devices to the destination services. First stage is when the application, respectively the user devices wants to connect the network. Before the authentication procedure start fuzzy logic is applied to determine security level of the connection and equipment. The inputs for the FIS can be from physical layer like the used protocol (for example Bluetooth, WiFi, ZigBee), through the transport layer (type of message protocol MQTT, AMQP and M3DA) and finally to the application layer where can be used different digital signatures, message authentication codes, security certifications and etc.

Second stage is before the start authentication procedure. Now after the devices are connected to the network some metrics can be taken. Based on this networks metrics and application profile, input parameters for FIS are created and security level is determined. For example if some health application is used, network parameters like time-delay, latency and bandwidth are considered. The system can be used for different applications: real-time, best-effort or emergency. Here different admission control and traffic engineering can be applied to achieve the required QoS of the applications.

On the next stage combination of environment parameters, accurate access control, policies and user credentials are used with different weights. The inputs from environment can be time, space and so on (for example open hours and days in Gym). It’s important and to divide the H2M from M2M communication, because this will give better access control and simple RBAC (for H2M) or ABAC (for M2M) can be applied. With this separation the speed and redundancy of the system will be improved.

The last stage is the authorization to the service. Here for the user credentials correct policies (based on type, status, time, location and etc.) are applied. After this the system shows that applications have achieved the required access control security level and can use the service.

****

Figure 6. Adaptable security algorithm / Access control design concept

The proposed adaptable security algorithm can be easy extend and adapt with different weights on the inputs and stages. This access concept gives the interoperability that is needed to connect all different M2M/IoT networks. It’s easy to adapt also different policies for the system and usage of M2M service. This means when the service is used for business logic and it is part of CRM, ERP, billing or telemedicine system.

In the different stages of the system model fuzzy logic is used to determine security level and to decide the access control. Fuzzy logic is a multi-value logic which permits intermediate values to be defined between conventional ones like true/false, low/high, good/bad etc. In a classical set theory, an element may either belong to set or not. In fuzzy set theory, an element has a degree of membership. A degree of membership function can be described as an interval [0, 1]. Each fuzzy system has its own criteria (inputs) and applies different functions.

Fuzzy inference systems (FISs) are also known as fuzzy rule-based systems, fuzzy model, fuzzy expert system and fuzzy associative memory. This is a major unit of a fuzzy logic system. The decision-making is an important part in the entire system. The FIS formulates suitable rules and based upon the rules the decision is made. This is mainly based on the concepts of the fuzzy set theory, fuzzy IF–THEN rules, and fuzzy reasoning. FIS uses “IF. . . THEN. . . ”statements, and the connectors present in the rule statement are “OR” or “AND” to make the necessary decision rules. The basic FIS can take either fuzzy inputs or crisp inputs, but the outputs it produces are almost always fuzzy sets. When the FIS is used as a controller, it is necessary to have a crisp output. Therefore in this case defuzziﬁcation method is adopted to best extract a crisp value that best represents a fuzzy set.

**CHAPTER 5**

# IMPLEMENTATION AND RESULTS

For implementation of the proposed method in pervious chapter is used Matlab and all build in tools that help to develop fuzzy system. The detailed scenario from the system model helps to design the inputs variables and to explain more easily the algorithm.

First step of algorithm is to determine the security of used protocol for device connection when they try to access the service. This will give the security level of the used devices and can reject the further communication even before authentication because of incompatibility.

## 5.1 Fuzzy system for device connection

The design of system is divided into three steps: design of the linguistic variables, design the fuzzy membership functions and finally define the rules for fuzzy system.

### 5.1.1 Linguistic variables

The inputs variables of different FIS are concrete like type of encryption or abstract like level of bandwidth. The outputs of each system are the level of security and the final output of all systems is again the level of security with linguistic variables low, medium and high.

### 5.1.2 Membership functions

The security level of device connections is defined based on different security criteria. For the Bluetooth connection is based on the version from 1.2, 2.1, 3.0 and 4. The level of security in WiFi is based on encryption of the keys – WEP, WPA and WPA2. The ranges for the inputs are shown in **Table 3** and **Table 4**.

Table 3. Range of inputs for Bluetooth

|  |  |
| --- | --- |
| Level of security | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

Table 4. Range of inputs for Wireless

|  |  |
| --- | --- |
| Level of security | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

The fuzzy sets above are represented by membership functions. The corresponding membership functions for Bluetooth and Wireless connections are presented in **Figure 7** and **Figure 8**.

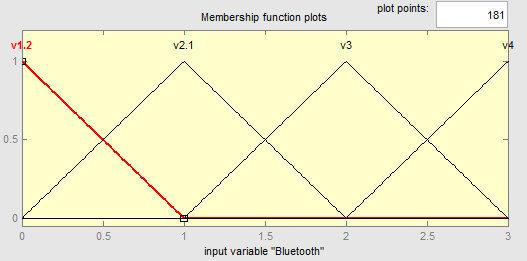


Figure 7. Membership function of bluetooth

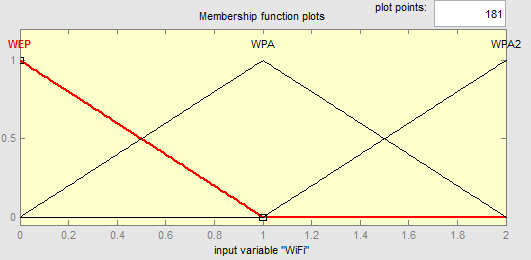


Figure 8. Membership function of wireless

Similarly, the output level of security is also represented by fuzzy sets and then a membership function. The level of security is defined based on the scales: low, medium and high secure within the range of [0 - 1]. The range definition is shown in **Table 5**.

Table 5. Level of Security

|  |  |
| --- | --- |
| Level of security | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

The membership function for the output fuzzy set is presented in **Figure 9**.

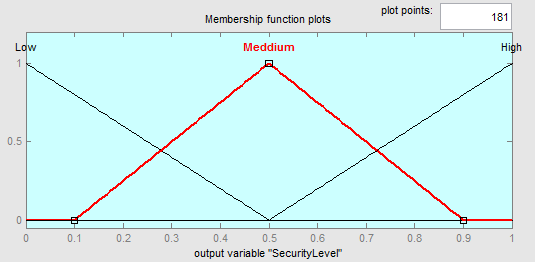


Figure 9. Membership function of security level

### 5.1.3 Rules of the fuzzy system

Once the input and output fuzzy sets and membership functions are constructed, the rules are then formulated. The rules are formulated based on the input parameters (Bluetooth, Wireless, ZigBee, RFID, ANT+) and the output i.e. level of security. The levels of bluetooth and wireless are used in the antecedent of rules and the level of security risk as the consequent of rules.

A fuzzy rule is conditional statement in the form:

IF x is A THEN y is B, where x and y are the variables while A and B are the values determine by fuzzy system. Some of the rules in this fuzzy system are as follow:

If (Bluetooth is v1.2) then (SecurityLevel is Low)(1)

If (Bluetooth is v2.1) then (SecurityLevel is Medium)(1)

If (Bluetooth is v3) then (SecurityLevel is Medium)(1)

If (Bluetooth is v4) then (SecurityLevel is High)(1)

If (WiFi is WEP) then (SecurityLevel is Low)(1)

If (WiFi is WPA) then (SecurityLevel is Medium)(1)

If (WiFi is WPA2) then (SecurityLevel is High)(1)

The antecedent and consequent of the rules can have multiple parts. All parts of the antecedent are calculated simultaneously and resolved in a single number. The antecedent affects all parts of the consequent equally. To view the rules can be used built in rule viewer.

### 5.1.4 FIS Evaluation

Final step is to evaluate the output of fuzzy system for a given input. For example, to evaluate device connection for Alice at the input is entered number 1. This number is equal to use of bluetooth version 2.1 with secure simple pairing. After evaluation the FIS will give score 0.500 which is equal to medium security level because of the rules that are defined for the system. Next user is Bob with device that has bluetooth version 4 with security manager and AES encryption. Here the input number is 3 and after evaluation the system give score 0.8367. This is in range between 0.7 and 1 and that score is for high level security. The gym scenario is not critical and desire level of security is medium. If it was critical scenario like hospital environment, then the minimum security level will be high and Alice will be deny from connection to the network. In this case where they are in the gym and both are trying to connect with secured smart phones and algorithm can continue to next stage and display them the authentication application where they can type their user names and passwords. After they have successfully connected to the network next stage of algorithm is to check the used protocols before give them authorised access to the services.

## 5.2 Fuzzy system for the protocols evaluation

This FIS will evaluate the different message protocols to give an accurate assessment of system security. Protocols that will be compared are designed for M2M communications. The major problem here is that some of them are text based when the other are binary. FIS system will help to compare them while ignore this and other difference between protocols. The main idea of evaluation the protocol is to change used one if the network connection is unstable and not reliable. This will give the users trust in the system and they will be sure that their connection is secured and no data will be lost.

### 5.2.1 Design the inputs

To compare the message protocols four different variables are select. The first input is latency which can deviate between 3s for some emergency medical applications, 30sec for home automation and 3min for ships monitoring systems. The second input is bandwidth which will determine how lightweight actually the protocol is it. For example MQTT protocol is good for low-bandwidth communications. Third variable is scalability.

### 5.2.2 Membership functions

The first membership function is the latency. The security of the system may depend on the latency of the system and that may cause some errors. The ranges for the inputs are shown in **Table 5Table 6**.

Table 6. Range of inputs for Latency

|  |  |
| --- | --- |
| Level of latency | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

The membership function for the input fuzzy set is presented in Figure 10.

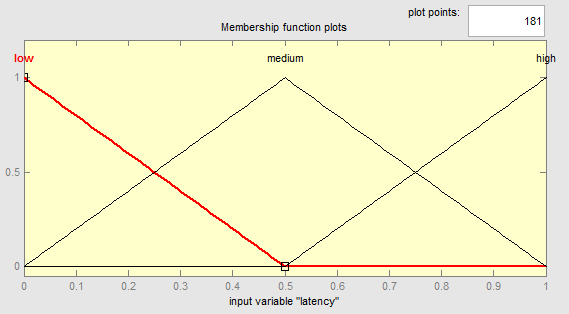


Figure 10. Membership function of latency level

The bandwidth is also important parameter. If the system must operate over low bandwidth network that will cause use of optimize for that protocol with fix sized headers and payloads. The ranges for the inputs are shown in Table 7.

Table 7. Range of inputs for bandwidth

|  |  |
| --- | --- |
| Level of bandwidth | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

The membership function for the input fuzzy set is presented in **Figure 11**.

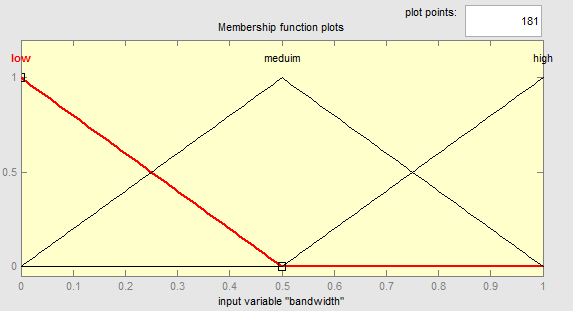


Figure 11. Membership function of bandwidth level

The scalability is defined in the same range, so the final decision will be easy comparable. Membership function is shown in **Figure 12Error! Reference source not found.**.

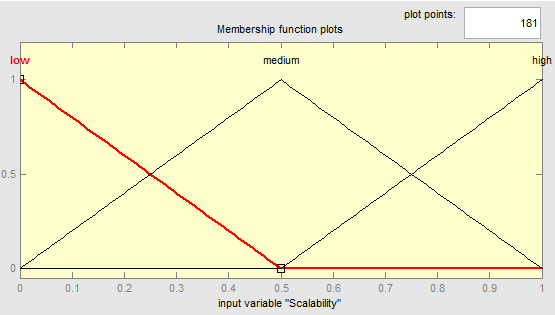


Figure 12. Membership function of scalability level

The example for scalability input of the protocol can be the decentralized approach or support unicast and multicast messages. If there is some discovery mechanism in the protocols this will also considered like scalability of growing networks. The fuzzy sets that represent the output security level are: low (L), medium (M) and high (H).

### 5.2.3 Rules of the fuzzy system

Once the input and output fuzzy sets are defined, next step is to write the rules. The rules are made of combination from the inputs and the output. The levels latency, bandwidth, QoS and scalability are used in the antecedent of rules and the level of security risk as the consequent of rules. Some of the rules in this fuzzy system are present in **Figure 13**.

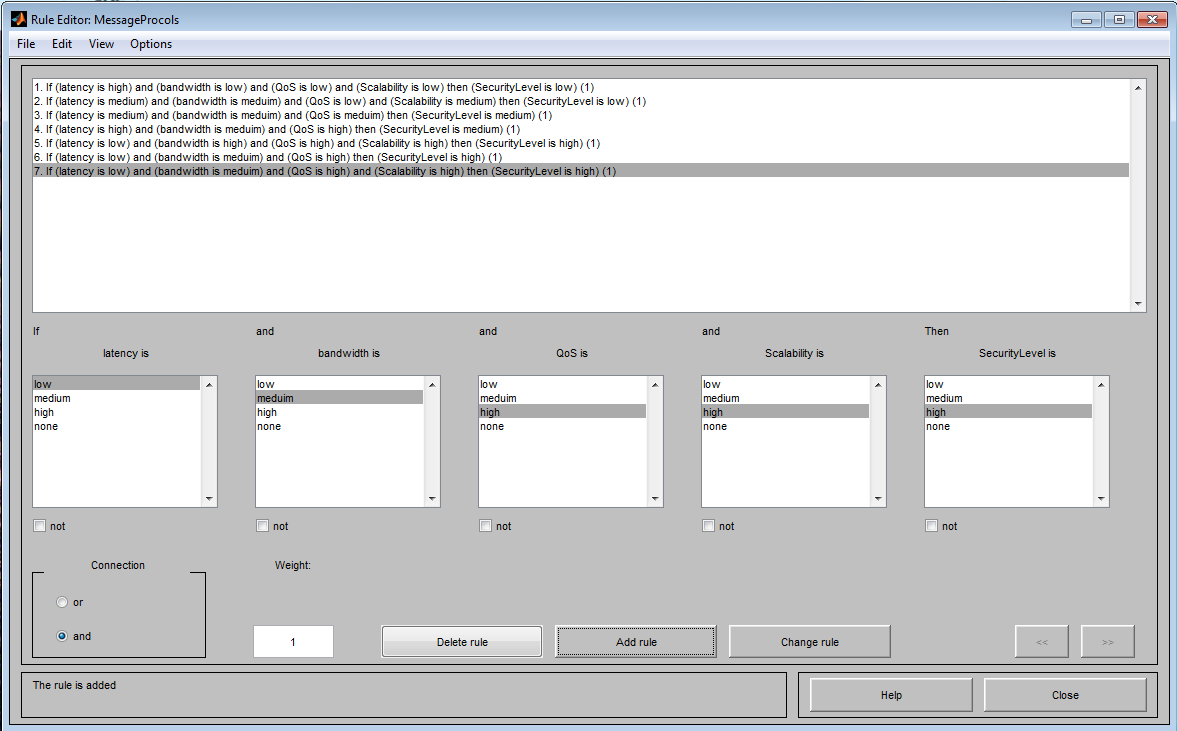


Figure 13. Fuzzy rules for message protocols security

### 5.2.4 FIS Evaluation

The evaluation of message protocols is more interesting in use case with the machine to machine communication. That’s why here the treadmills will be selected for evaluation. The latency of treadmill#11 is low based on the measurements of the network and this is because it used MQTT message protocol. For the scenario the other treadmill#12 has medium latency because it used AMQP protocol. The bandwidth for both treadmills has high score because they use fast wireless connection. The scalability is also scored with high values because the both message protocols can handle multiple connections from the users and also from the coach without problem. The final evaluation of protocols has score 0.6843 for treadmill#11 and 0.4729 for treadmill#12. Both scores are between 0.3 and 0.7, where is the area of medium security level. The usage of this protocols require broker to handle the messages. This is the reason that it should be FIS system and for the brokers. Evaluation of the protocols will be used like input for the FIS system of the brokers.

## 5.3 Fuzzy system for the brokers evaluation

The different type of M2M protocols need message broker. To support the new developed protocols many projects appears last few years. Some of the projects build the message broker from the scratch and another just extend the previous versions and try to include the new protocol. With all this diversity it is really hard to select the right broker and also to trust it like secure solution.

### 5.3.1 Design the inputs

To determine the security level and trust in the broker must be defined the inputs of FIS. The performance of one broker depends of the latency, bandwidth and QoS. Another major characteristic is the supported protocols and also language of the APIs (C, C++, Objective C, .NET, Java, Erlang, Perl, Python, PHP, Ruby and etc.). Each of this inputs have its own security aspects and after evaluate them with FIS system the common grade of security will be reached.

### 5.3.2 Membership functions

The performance of the broker is the critical membership function when we talk about M2M communications and millions of requests. Defined are three level of performance: low, medium and high. The ranges for the inputs are shown in **Table 8**.

Table 8. Range of inputs for performance

|  |  |
| --- | --- |
| Level of Performance | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

Another membership function is security level of protocols shown in **Table 9**.

Table 9. Range of inputs for security

|  |  |
| --- | --- |
| Level of Security (Protocols) | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

The last membership function is the interoperability of the broker and his APIs. Again the inputs are defined in three levels low, medium and high (**Table 10**).

Table 10. Range of inputs for interoperability

|  |  |
| --- | --- |
| Level of Interoperability (APIs) | Range |
| Low | 0 – 0.3 |
| Medium | 0.3 – 0.7 |
| High | 0.7 – 1 |

### 5.3.3 Rules of the fuzzy system

Different output levels of the FIS are based on the rules that are applied on the membership functions. Some of the rules in this fuzzy system are as follow:

If (Performance is low) and (ProtocolsSecurity is low) then (SecurityLevel is Low)(1)

If (Performance is high) and (ProtocolsSecurity is low) then (SecurityLevel is Low)(1)

If (Performance is low) and (Interoperability is high) then (SecurityLevel is Medium)(1)

If (Performance is medium) and (Interoperability is high) and (ProtocolsSecurity is medium) then (SecurityLevel is Medium)(1)

If (Performance is high) and (Interoperability is high) and (ProtocolsSecurity is high) then (SecurityLevel is High)(1)

### 5.3.4 FIS Evaluation

The used brokers in the scenario are Mosquitto and ActiveMQ. The performance on both brokers is high, but the protocol security on MQTT is low while the AMQP has high security. The interoperability on ActiveMQ is also high because support both protocols from the scenario and has more APIs. After the evaluation Mosquitto has score 0.284. This means that is has low security level because even it has high performance it communicate with low security protocol MQTT and didn’t support AMQP and other APIs. The ActiveMQ broker has score 0.837 and that is in range between 0.7 and 1 where is the area of high level security. From this evaluation we can conclude that is better two improve the system and use ActiveMQ broker for both protocols.

The final step is access control and policies that are applied for authorization. The algorithm, described in chapter 4 is used to separate the H2M and M2M flow. The idea of is to have fast and scalable access control that can handle billions of connections. It will give also an opportunity to have finer grade policies.

## 5.4 Policy model

Developed policy model includes different roles (e.g., staff, coach, customer). A user can be a member of many roles. The policy model has four permission types: human, machine, services and application management. Human management permission represents the ability to add, remove or modify users role. Machine management permissions represent the ability to add, remove and modify devices. The services management has ability to start, stop or modify some service like start or stop the bikes or treadmills in the gym. The application management permissions represents the ability to deploy, execute, modify or to monitor the applications like to getting heart monitoring feed or to get the current speed and program time of treadmill.

The policy model represents access policies similar to the ones we enforce in real life. We identified universal attributes capable of expressing a wide range of rules when combined. In that type of definition will be easy to define access to a device, service or application. The thing type attribute limits the access to a given thing type (e.g., treadmill, customer, application). The location attribute determines access to the device or machine in a specific location. It also limits access to a machine by restricting user location at access time. Thing status attribute limits the access to a machine or application by restricting the status of the thing at that time (e.g., treadmill is broken). The time attribute limits the access to a machine for a period of time (e.g., gym working hours, program time period). The policy maps a user role with permissions on things with a collection of attributes.

The things type attribute is important to determine with what we want to interact and what type of user roles will be applied when the system will interact with third party or other federated system with limited access. Inside the local cloud with have also different type of users with different permissions depends on the policies. The location attribute is good for more detailed control over the local system. For example in the fitness instructor can limit the access of some equipment for special training. The male users will have access to all programs, while female users will be able to use only lightweight equipment or programs to prevent some injuries. Thing status is also important, because almost for every action it will require some feedback. Figure 14 shows that instructor wants to limit the access to some treadmills or bike but there are customers that already use them. He will be not able to stop them will they have training program running already. The time attribute will make easy to control the access for different time of the week or within the day. For example when there is some discount of price for the fitness in the morning hours until 2PM.

|  |
| --- |
| <Rule>  <Target>  <Subjects><Subject>  <SubjectMatch MatchId=”string-equal”>  <SubjectAttribute DataType=”string” AttributeId=”group”/>  <AttributeValue DataType=”string”>Training</AttributeValue>  </SubjectMatch>  </Subject></Subjects>  <Resources> <Resource>  <ResourceMatch MatchId=”string-equal”>  <ResourceAttribute dataType=”string” AttributeId=”domain”/>  <AttributeValue DataType=”string”>Treadmill</AttributeValue>  </ResourceMatch>  <ResourceMatch MatchId=”string-equal”>  <ResourceAttribute DataType=”string” AttributeId=”status”/>  <AttributeValue DataType=”string”>Reserved</AttributeValue>  </ResourceMatch>  </Resource></Resources>  <Actions><Action>  <ActionMatch MatchId=”string-equal”>  <ActionAttribute DataType=”string”AttributeId=”action-id”/>  <AttributeValue DataType=”string”>control</AttributeValue>  </ActionMatch>  </Action></Actions>  </Target>  <Condition FunctionId=”time-equal”>  <Apply FunctionId=”time-one-and-only”>  <TimeAttribute DataType=”time” AttributeId=”current-time”/>  </Apply>  <AttributeValue DataType=”time”>17:00:00</AttributeValue>  </Condition>  </Rule> |

Figure 14. Example of XACML Policy

The final step of proposed algorithm is authorization. After successful applied policies both users and machines will have access to the service from the network. They can star using the system that will provide the desire security level. This will increase the trust in the system and also give the users confidence for their data.

**CHAPTER 6**

# CONCLUSIONS

## 6.1 Findings

In the process of research and learning became clear that a many issues and problems surround the dynamic developing of new technologies. The computing capabilities of small devices increase significant over the last few years and this open whole new sector for smart mobile devices. With this growing market of smart devices, services and applications customers have problems to choose the correct technical solution. That’s why standards developing organization start to cooperate and combine the effort to make unified definition for the M2M/IoT specifications. This will help the customers to choose easy any kind of solution and to feel safe that it will work properly and the information will be secured.

The security and access control systems are really important to whole Internet of Things and M2M world. There are a lot of discussions and working groups for this topic and how to bring it to the end users. Using the Internet is an essential part and doing it in a secure way for both the devices and users is really important. The security requirements for the users and for the devices are different and this leads to various technics for data protection.

## 6.2 Future work

It will be interesting to use the existing library to export the fuzzy logic into programmable language like C, java, .net or python. This will help to implement proposed access control design into future M2M middleware frameworks. It might be also need to include ANFIS (Adaptive neuro fuzzy inference system) for creating more accurate rules and self-learning system. ANFIS also can fit into the intrusion prevention part of the security system.

It might be necessary to redesign this system in a way that it will be deployable and will be without the use of MATLAB. It might also be necessary to use an adaptive fuzzy logic technique for security risk analysis.

Designed system can be used to evaluate the security and trust in used unknown environment and to extend the secure module in already design software systems. This can be the right way to achieve different requirement from the stakeholders and specific business domain. With adaptable system security the information will be transferred between M2M or H2M in whole internet of things.

Interoperability it’s really important aspect and can be achieved through a mediation layer. The proposed access control logic can act like an adapter that allows applications and services to communicate regarding their differences. When the logic is implement in software with open interfaces the system can be easily integrated. With RESTfull APIs and support of many protocols integration will be easy task without rewriting some of the interfaces.

The Security of M2M/IoT is really important aspect of future architectures and frameworks. When every small device that generates some sort of data has been connected to the Internet, risks of unauthorized access of this data increase significant. With all the variety of standards and protocols there will be new range of attacks design for this type of communications.

# REFERENCES

|  |  |
| --- | --- |
| [1] | A.-I. LABS, " http://www.autoidlabs.org/," [Online]. |
| [2] | "IETF". |
| [3] | "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update," 2012-2017. |
| [4] | "BETaaS," [Online]. Available: http://www.betaas.eu/. |
| [5] | M. .J, "Web Services Security : Access," 2004. |
| [6] | S. K. S. K. R. R. O. Garcia-Morchon, "Security Considerations for teh IoT," IETF CoRE, 2012. |
| [7] | T. H. K. W. R. Hummen, "A security protocol adaptationlayer for the ip-based internet of things," Interconnecting Smart Objects with the Internet Workshop, 2011. |
| [8] | "Facebook engineers," [Online]. Available: https://www.facebook.com/Engineering. |
| [9] | "M3DA," [Online]. Available: http://wiki.eclipse.org/images/d/d4/M3DAProtocolSpecification.pdf. |
| [10] | L. S. H. A. Thiago Sales, "A UPnP extension for enabling user authentication," 2010. |
| [11] | A. J. J. a. A. F. S. Miguel Castro, "An analysis of M2M platforms: challenges and opportunities for the Internet of Things," Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012. |
| [12] | "Sen.se," [Online]. Available: http://www.sen.se. |
| [13] | " EVRYTHNG," [Online]. Available: http://www.evrythng.com. |
| [14] | "AMEE," [Online]. Available: http://www.amee.com. |
| [15] | "RunMyProcess," [Online]. Available: www.runmyprocess.com. |
| [16] | "Axeda Machine Cloud," [Online]. Available: http://www.axeda.com/node/811. |
| [17] | "ThingWorx - M2M Application Platform," [Online]. Available: www.thingworx.com. |
| [18] | "Eclipse Mihini project," [Online]. Available: http://www.eclipse.org/mihini/. |
| [19] | "Internet-of-Things Architecture," [Online]. Available: http://www.iot-a.eu/. |
| [20] | "OpenMTC," [Online]. Available: http://www.open-mtc.org/. |
| [21] | "PrimeLife," [Online]. Available: http://primelife.ercim.eu/. |
| [22] | "OpenIoT," [Online]. Available: http://openiot.eu/. |
| [23] | "Avahi," [Online]. Available: www.avahi.org. |
| [24] | "Bonjour," [Online]. Available: www.apple.com/support/bonjour/‎. |
| [25] | "UPnP Forum," [Online]. Available: www.upnp.org/. |
| [26] | "Service Location Protocol RFC 2608," http://tools.ietf.org/html/rfc2608. |
| [27] | "Vendor extensions for Service Location Protocol RFC 3224," http://tools.ietf.org/html/rfc3224. |
| [28] | S. R. G. U. B. J. K. Tyrone Grandison, "Protecting Privacy while Sharing Medical Data between Regional Healthcare Entities," IBM Almaden Research Center, 2007. |
| [29] | Official ISC2 Guide to the CISSP CBK. |
| [30] | D. Hilley, "Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings," 2009. |
| [31] | K. Lin, "A Reputation and Trust Management Broker Framework for Web Applications," 2005. |
| [32] | R. J. Alan O'Connor, "2010 Economic Analysis of Role-Based Access Control," NIST, 2010. |
| [33] | D. F. R. K. Ravi Sandhu, "The NIST Model for Role-Based Access Control," NIST, 2000. |
| [34] | T. M. S. Godik, "eXtensible Access Control Markup Language," OASIS committee speciﬁcation, 2003. |
| [35] | E. A. IMAN ALMOMANI, "A POWER-EFFICIENT SECURE ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS," 2010. |
| [36] | K. Göztepe, "Designing a Fuzzy Rule Based Expert System for Cyber Security," International Journal of Information Security Science, March 2012. |
| [37] | S. S. a. S. N. D. S. N. Sivanandam, Introduction to Fuzzy Logic using MATLAB, Springer, 2007. |
| [38] | S. T. Belton V., "Multiple Criteria Decision Analysis," 2002. |
| [39] | R. a. R. H. Keeney, "Decisions with Multiple Objectives: Preferences and Value," John Wiley & Sons, 1976. |
| [40] | D. a. E. W. Winterfeld, "Decision Analysis and Behavioral Research," Cambridge University Press, 1986. |
| [41] | R. Schaefer, "Rules for Using Multi-Attribute Utility Theory for Estimating a User’s Interests," ABIS-Adaptivität und Benutzermodellierung, 2001. |
| [42] | "IoT Toolkit," [Online]. Available: http://iot-toolkit.com/. |
| [43] | "Libelium," [Online]. Available: http://www.libelium.com/. |
| [44] | "Withings," [Online]. Available: http://www.withings.com/. |
| [45] | "Cisco ISR G2," [Online]. Available: http://www.cisco.com/en/US/products/ps10906/Products\_Sub\_Category\_Home.html. |
| [46] | "deviceWISE," [Online]. Available: http://www.devicewise.com/. |
| [47] | "Continua Alliance - Health and wellness vision," [Online]. Available: http://www.continuaalliance.org/connected-health-vision/health-wellness. |
| [48] | K. H. C. B. a. B. F. Z. Shelby, "Constrained Application Protocol (CoAP)," draft-ietf-corecoap-07, 2011. |

# APPENDIX

Matlab software is used for the implementation of prosed algorithm. It has built it fuzzy logic toolbox graphical user interface (GUI). It can be start from command line by typing fuzzy. This will launch FIS editor.

## The FIS editor

The fuzzy inference system editor (Figure 15) shows a summary of the fuzzy inference system. It shows the mapping of the inputs to the system type and to the output. The names of the input variables and the processing methods for the FIS can be changed through the FIS editor.

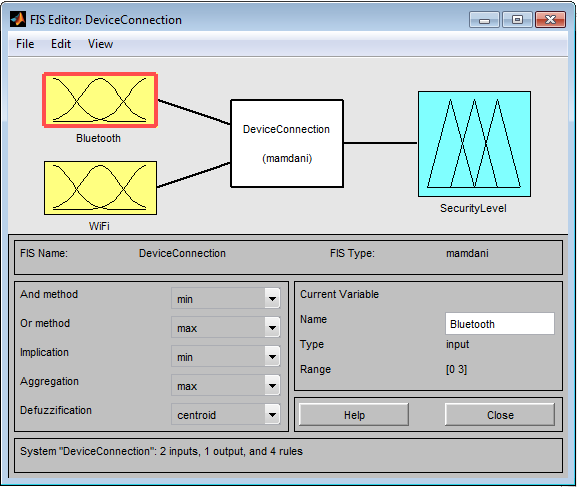
****

Figure 15. The FIS editor

## The membership function editor

This can be opened from the command window by using the “plotmf” function but more easily through the GUI. The membership function editor (Figure 16) shows a plot of highlighted input or output variable along their possible ranges and against the probability of occurrence. The name and the range of a membership value can be changed, so also the range of the particular variable itself through the membership function editor.

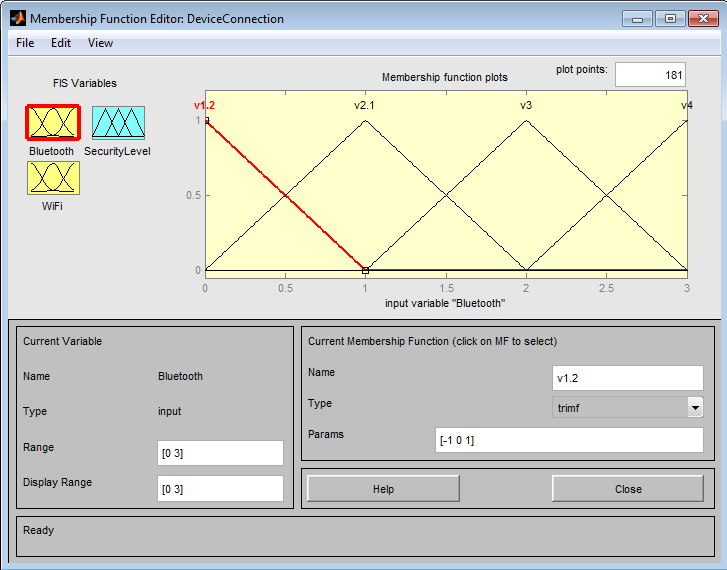


Figure 16. The Membership Function editor

## The rule editor

The rule editor can be used to add, delete or change a rule. It is also used to change the connection type and the weight of a rule. The rule editor for this application is shown in Figure 17.

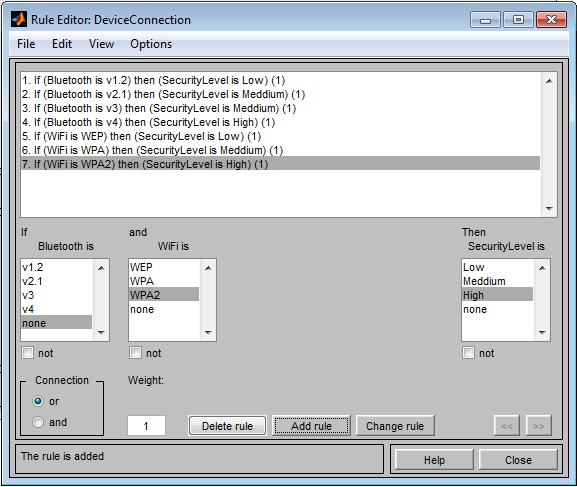


Figure 17. The Rule editor

## The rule viewer

The text box captioned input is used to supply the two input variables needed in the system. The appropriate input corresponds to the number of YES answer in the questionnaire for each of the input variables. The input for each of the input variables is specified at the top of the section corresponding to them, so also the output variable. The rule viewer for this work is presented in Figure 18.

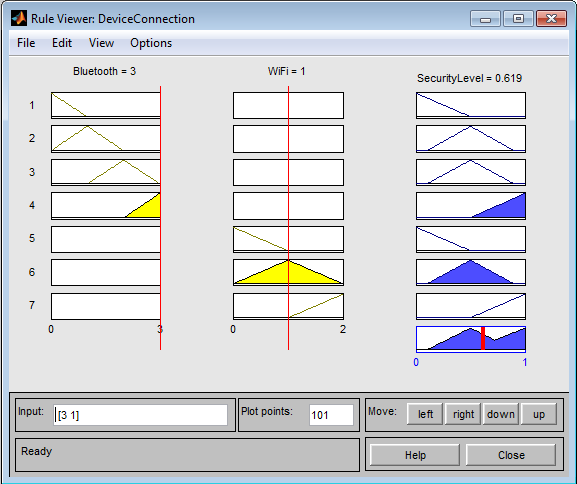


Figure 18. The Rule viewer

## The surface viewer

The surface viewer shown in Figure 19**Error! Reference source not found.** is a 3-D graph that shows the relationship between the inputs and the output. The output (Security level) is represented on the Z-axis while 2 of the inputs (Bluetooth and WiFi) are on the x and y axes. The surface viewer shows a plot of the possible ranges of the input variables against the possible ranges of the output. The same view can be evoke by command line with “gensurf” function. Example: a = readﬁs(‘tippersg’); gensurf(a);

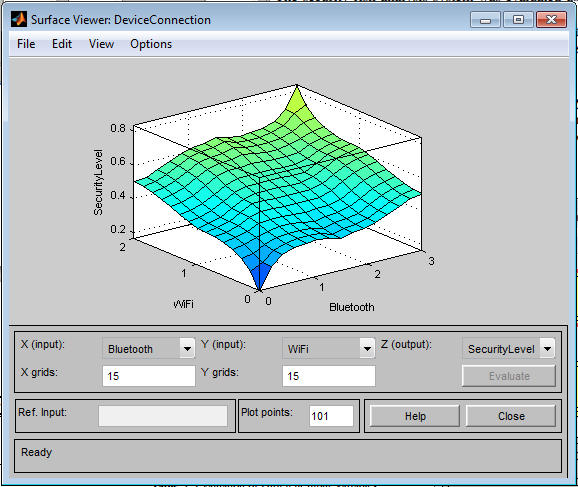


Figure 19. The Surface viewer

The main idea to use FIS for protocols is to choose best protocols based on requirements of the use case. For example if we want bandwidth efficiency using a binary protocol is the right choice for that scenario.

## FIS Evaluation

To evaluate the output of a fuzzy system for a given input, is used the function evalfis. For example, the following script evaluates device connection at the input, [0.5 0.2].

>> evalfis([0.5 0.2], a)

ans =

0.3632

## The FIS Structure

The FIS structure is the MATLAB object that contains all the fuzzy inference system information. This structure is stored inside each GUI tool. Access functions such as getfis and setfis make it easy to examine this structure.

All the information for a given fuzzy inference system is contained in the FIS structure, including variable names, membership function definitions, and so on. This structure can itself be thought of as a hierarchy of structures, as shown in the following Figure 20.

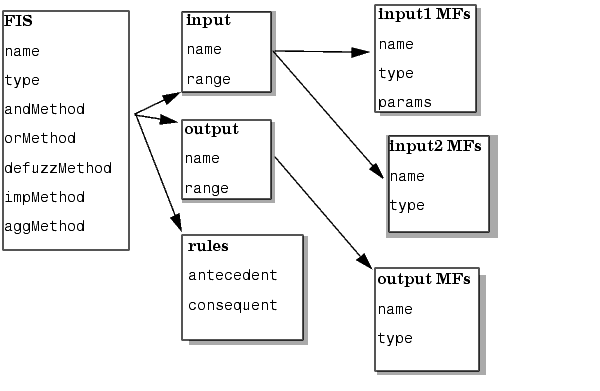


Figure 20. FIS structure

Listing of information on the FIS can be generated by using the “showfis” command:

showfis(b)

1. Name MessageProcols

2. Type mamdani

3. Inputs/Outputs [4 1]

4. NumInputMFs [3 3 3 3]

5. NumOutputMFs 3

6. NumRules 7

7. AndMethod min

8. OrMethod max

9. ImpMethod min

10. AggMethod max

11. DefuzzMethod centroid

12. InLabels latency

13. bandwidth

14. QoS

15. Scalability

16. OutLabels SecurityLevel

17. InRange [0 1]

18. [0 1]

19. [0 1]

20. [0 1]

21. OutRange [0 1]

22. InMFLabels low

23. medium

24. high

25. low

26. meduim

27. high

28. low

29. meduim

30. high

31. low

32. medium

33. high

34. OutMFLabels low

35. medium

36. high

37. InMFTypes trimf

38. trimf

39. trimf

40. trimf

41. trimf

42. trimf

43. trimf

44. trimf

45. trimf

46. trimf

47. trimf

48. trimf

49. OutMFTypes trimf

50. trimf

51. trimf

52. InMFParams [-1 0 0.5 0]

53. [0 0.5 1 0]

54. [0.5 1 1.5 0]

55. [-1 0 0.5 0]

56. [0 0.5 1 0]

57. [0.5 1 1.5 0]

58. [-1 0 0.5 0]

59. [0 0.5 1 0]

60. [0.5 1 1.5 0]

61. [-1 0 0.5 0]

62. [0 0.5 1 0]

63. [0.5 1 1.5 0]

64. OutMFParams [-1 0 0.5 0]

65. [0 0.5 1 0]

66. [0.5 1 1.5 0]

67. Rule Antecedent [3 1 1 1]

68. [2 2 1 2]

69. [2 2 2 0]

70. [3 2 3 0]

71. [1 3 3 3]

72. [1 2 3 0]

73. [1 2 3 3]

67. Rule Consequent 1

68. 1

69. 2

70. 2

71. 3

72. 3

73. 3

67. Rule Weight 1

68. 1

69. 1

70. 1

71. 1

72. 1

73. 1

67. Rule Connection 1

68. 1

69. 1

70. 1

71. 1

72. 1

73. 1

>>