Public

A state of the art related to scope of M2MGrids project

M2MGrids
ITEA 2 13011

••

Edited by Juhani Latvakoski

Including contributions from: all M2MGrids project partners

TABLE OF CONTENTS

TABLE OF FIGURES

Abstract

This deliverable provides a view to the state of the art related to M2MGrids project scope. First a view to the architectures of Internet of things (IoT) is discussed. Then the M2M system levels related to information, services, communication and security are discussed. Finally some concluding remarks are provided.

List of Abbreviations

Abbreviation	Explanation
AoA	Angle-of-Arrival
AoD	Angle-of-Departure
AMQP	Advanced Message Queuing Protocol
DDS	Data Distribution Service for Real-Time Systems
DODAG	Destination Oriented Acyclic Directed Graph
ECM	Energy Consumption Monitoring and Optimization
FDM	frequency-division multiplexing
FSK	Frequency-shift keying
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HSPA	High-Speed Packet Access
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IoT	Internet-of-Things
ISP	Internet service provider
LTE	Long Term Evolution

M2M	Machine-to-Machine
MR-FSK	Multiple frequency-shift keying
MR-OFDM	Multi-Rate OFDM
MR-O-QPSK	Multi-Rate Offset Quadrature phase-shift keying
OFDM	Orthogonal frequency-division multiplexing
oneM2M	oneM2M is the global standardisation body for M2M and IoT
POP	Post Office Protocol
QPSK	Quadrature phase-shift keying
SIP	Session Initiation Protocol
SMS	short message service
SMTP	Simple Mail Transfer Protocol
SUN	Smart Metering Utility Network
VPN	Virtual Private Network
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
WWW	World Wide Web
XMPP	Extensible Messaging and Presence Protocol
Zero MQ	ØMQ (also spelled ZeroMQ, 0MQ or ZMQ) is a high-performance asynchronous messaging library

1 Introduction

This deliverable provides a view to the state of the art related to M2MGrids project scope. First a view to the architectures of Internet of things (IoT) is discussed. Then the M2M system levels related to information, services, communication and security are discussed. Finally some concluding remarks are provided.

2 Review of the Internet of Things architectures

A view to the related standards & industrial forums working with the Internet of Things (IoT) architectures are shown in the Figure 1. [AIOTI]. There are huge amount of different specifications arising from different vertical domains. The aim of this chapter is to discuss about a couple of essential IoT architectural descriptions and patterns and compare them with our approach.

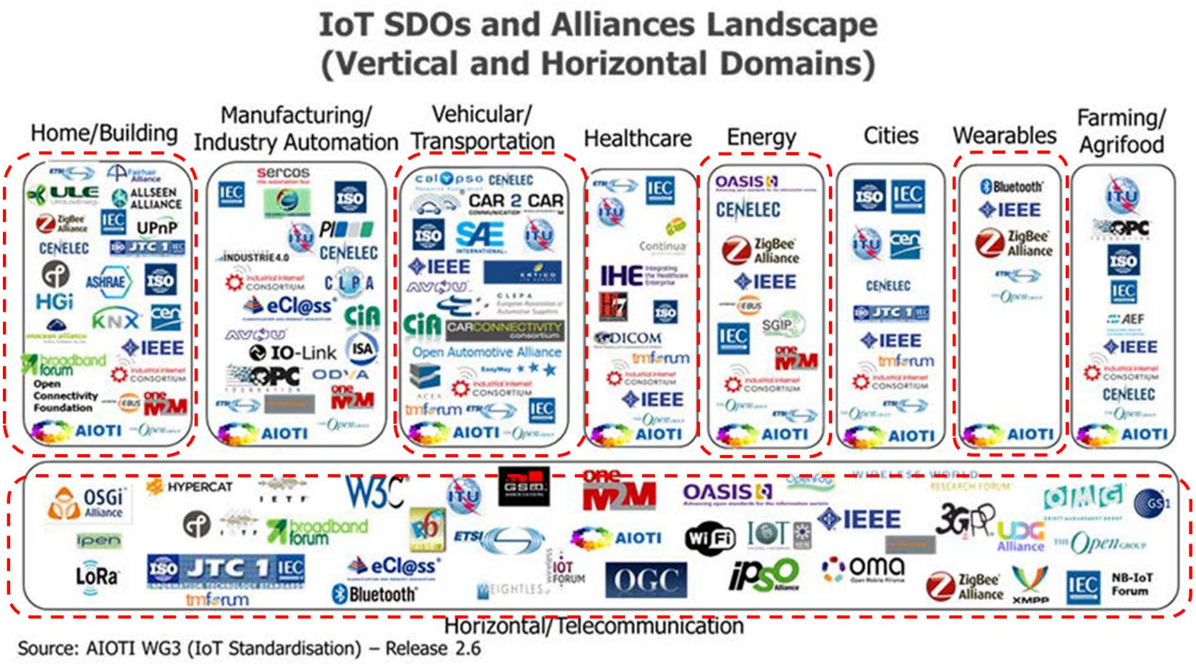


Figure 1. The related Standards & Industrial Forums for the Internet of Things standardization [AIOTI].

An architectural pattern of IoT system is related to the edge system, Figure 2 [IEC]. The edge system has typically identifiable physical entities, which can be connected to IoT infrastructures and platforms either directly or via some short of a gateway. The physical entities may be either online or offline at any given time and they may have information processing capabilities. The second typical element in the IoT architectures is the abstraction called as IoT platform, which is an integrated physical/virtual entity system capable for controlling, operation, information processing and application execution. Our approach applies both the edge and IoT platform patterns in the architectural approach.

There are several views to the IoT platform architectures, for example ITU-T IoT model is divided into four layers: application, service support and application, support, network, and device as shown in the Figure 3 [ITU12]. We basically apply here quite much the same structure than ITU-T model.

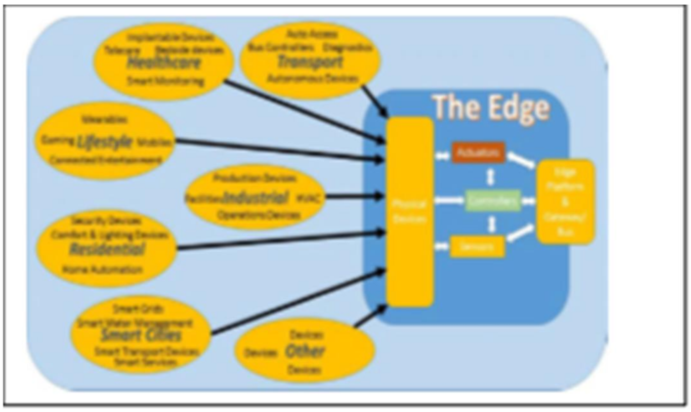


Figure 2. A view to the edge system [IEC].

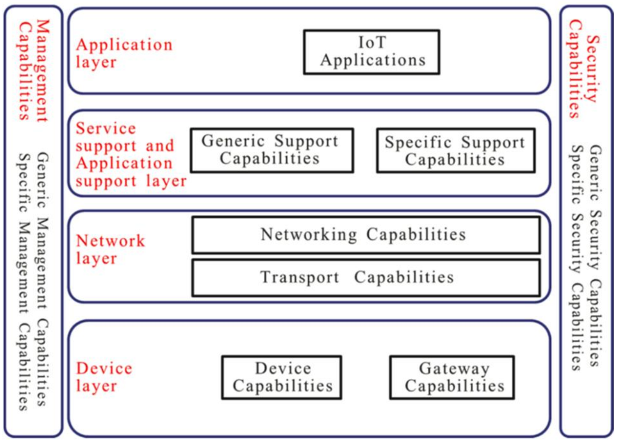


Figure 3. ITU-T Y.2060 view to the IoT platform [IEC].

The IoT reference architecture model defined for industrial manufacturing systems applies in the three dimensions: life cycle and value stream (IEC 62890), hierarchy levels (IEC 62264/IEC 61512) and system layers, Figure 4 [ZVEI]. Industrial Internet consortium has divided the architecture to three tiers: edge, platform and architecture tiers [IIC15, IEC]. The edge tier collect data from edge nodes, devices or things. This data is forwarded over the access network to the platform tier, which processes data for forwarding it to the enterprise tier (and vice versa). The platform tier uses the service network to communicate with the enterprise tier, which provides end user interfaces, control commands and domain-specific applications. The division to multiple tiers have been deployed in our approach, however, the our business cases related to multiple stakeholders and even human consumers/prosumers requires architectural capabilities to go beyond private networks of each stakeholder and represent more dynamic structures for the relationship management.

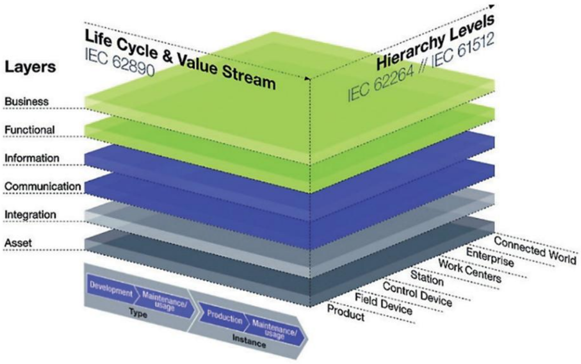


Figure 4. IoT Reference architecture model of Industry 4.0.

FP7 IoT-A project defined an architecture reference consisting of domain, information, functional, communication and security models, Figure 5. The domain model is responsible for outlining core concepts in the IoT such as "devices", "IoT services", and "virtual entities" (which model physical entities). The information model defines the generic structural properties of information in an IoT system. The functional model identifies groups of functionalities based on the relations defined in the domain model. The communications model addresses the complexity of communications in IoT environments. The trust, security and privacy (TSP) model is specifically identified by its importance to IoT use-case scenarios addressed separately [IoT12].

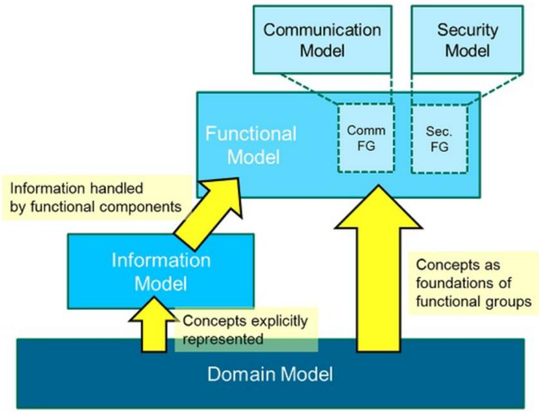


Figure 5. IoT-A project architecture reference model.

Architecture of Internet of Things Innovation (AIOTI) has basically continued the IoT-A initiated works, and defined the IoT reference architecture in the form of Figure 6. In the AIOTI domain model, a user interacts with a physical entity via digital representation "virtual" IoT service. The functional model is composed of three layers - application, IoT and network layers. The same type of approach seems to be used also in the Fiware open specifications [FIW16] and ETSI Smart M2M WG [ETS17], which is studying virtualized IoT architectures with cloud back-ends which tries to combine traditional IoT with cloud computing. This approach try to apply the best practices derived from IT back-office services, and try to deploy them also in the edges of the networks with IoT devices. We use the same type of approach and try to apply cloud/virtualization techniques in the information service layer.

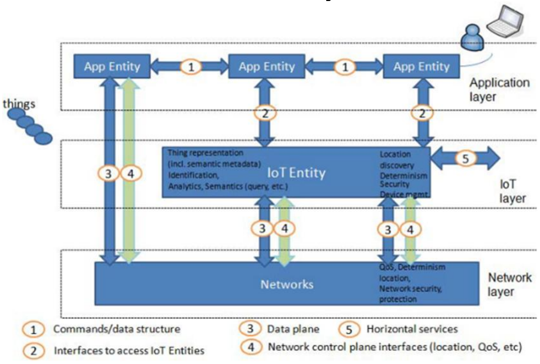


Figure 6. AIOTI, Alliance for Internet of Things Innovation, view to the IoT architecture.

IEEE, OMG, W3C, OpenFog, AllSeen alliance, NGMN are defining/have defined their own views towards the IoT architectures. In addition to these approaches, there are several publications, like e.g. comparison of IoT Platform Architectures [Guth16], Wireless communication and security issues for cyber-physical systems and internet of things [Bur18], including state of the practise Amazon Web Services [Ama18].

The WG3 of the ISO/IEC JTC1 WG41 has been working towards specifying the reference architectures for the Internet of Things [ISOI17]. The WG3 has divided the IoT reference architecture into functional, system deployment, communications and usage views. In addition, the IoT system is composed of user, operation & management, application & services, access & communications, sensing & controlling, physical entity domains and a set of cross-domain capabilities such as connectivity, privacy protection, security, resilience, interoperability and dynamic composition. The specification work has been ongoing, snapshot visualized in the

Figure 7 [ISOI17]. In addition, there are interesting IoT architectures related actions ongoing such as e.g. securing IoT products with blockchain [TRU-IoT]. However, it seems that our architecture is a bit more detailed level and we could have quite much to contribute into the architecture works e.g. under the referred forums.

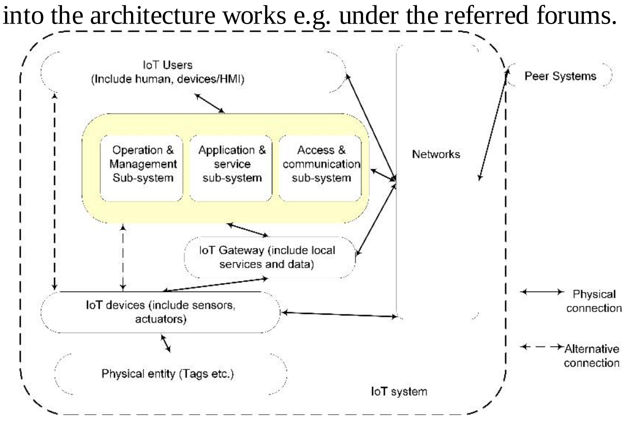


Figure 7. IoT reference architecture [IEC17].

There are a number of challenges in the existing IoT architectures. An essential challenge arises from our use cases collaborating with both consumer and industrial endpoints, which are not supported properly by any of the existing architectures. There are also fundamental requirements towards crosscutting solutions in the areas of security, safety, integrability, interoperability, composability, data management, analytics, resiliency, composability, virtualization, and regulation. Our architecture relies on the principles for creating horizontal solutions, which can be deployed in multiple domain scenarios and minimizing the need for application domain specific solutions [LAG14].

The dynamic changes in the IoT systems, such as continual adding of physical entities involving OEMs and related SPs, heterogeneous sensors and actuators and other devices, have proved to be challenging especially from security point of view. The industrial IoT devices are often planned to be used in physically protected and isolated environments, however, today there is need to enhance the operation also with many other devices (e.g. in energy flexibility cases). The state of the art IoT architectures lacks proper solutions for solving this challenge. Therefore, the proposed architecture contribute novel means for enabling required collaborations in the selected use cases.

Proper identification, authentication and authorization capabilities seem to be missing for dynamic IoT environments, which prevents establishment of trust relationship. Therefore, uncertainty is caused for the data/information ownership and validity, and to remote management of the physical assets including reconfiguration and reprogramming on the fly. Therefore, our architecture contributes communication and security related solutions towards solving these challenge at least partially.

References

[1] [GLW01] Gillet, S. H., Lehr, H., Wroclawski, J.T., Clark, D.D. Do Appliances threaten internet innovation? IEEE Communication Magazine. Oct 2001. Pp 46-51.

[2] [LJ16] Latvakoski, J. Small world for dynamic wireless cyber-physical systems. Academic Dissertation publication VTT Science 142. December 2016. 102p. + app. 163p. ISBN 978-951-38-8477-2. ISBN 978-951-38-8476-5.

[3] [CSH06] N. Choi, I.-Y. Song, and H. Han. A survey on ontology mapping. SIGMOD Rec., 35(3):34–41, 2006.

[4] [CRM02] R. Culmone, G. Rossi, and E. Merelli. An Ontology Similarity Algorithm for Bioagent. In Proc. of NETTAB'02, 2002.

[5] [DZ13] Damen, S., Zannone, N.: Privacy implications of privacy settings and tagging in facebook.

[6] In: Secure Data Management. pp. 121–138. LNCS 8425, Springer (2013).

[7] [NHG06] L. D. Ngan, T. M. Hang, and A. E. S. Goh. Semantic Similarity between Concepts from Different OWL Ontologies. In Proc. of IEEE International Conference on Industrial Informatics, pages 618–623, 2006.

[8] [NA06] H. Nguyen and H. Al-Mubaid. A Combination-based Semantic Similarity Measure using Multiple Information Sources. In Proc. of IRI'06, pages 617–621. IEEE Press, 2006.

[9] [LAG14] Latvakoski, J.; Alaya, M.B.; Ganem, H.; Jubeh, B.; Iivari, A.; Leguay, J.; Bosch, J.M.; Granqvist, N. Towards Horizontal Architecture for Autonomic M2M Service Networks. *Future Internet* **2014**, *6*, 261-301.

[10] [LIV14] Latvakoski, J.; Iivari, A.; Vitic, P.; Jubeh, B.; Alaya, M.B.; Monteil, T.; Lopez, Y.; Talavera, G.; Gonzalez, J.; Granqvist, N.; Kellil, M.; Ganem, H.; Väisänen, T. A Survey on M2M Service Networks. *Computers* 2014, *3*, 130-173.

[11] [OGC13] OGC Geospatial eXensible Access Control Markup Language (GeoXACML) 3.0 Core. OGC Discussion Paper, Open Geospatial Consortium, 2013.

[12] [Dat14] Study on future information exchange solutions in the electricity retail market, 2014. http://www.fingrid.fi/fi/asiakkaat/asiakasliitteet/Datahub/Datahub_final_en.pdf

[13] [eBI16] ebIX - Home, 2016. http://www.ebix.org/content.aspx?ContentId=984&SelectedMenu=1

[14] [eBI15] ebIX - Introduction to Business Requirements and Information Models, 2015. https://dl.dropboxusercontent.com/u/18527515/Ebix/Introduction_to_ebIX_Models_1r0B_20151101.pdf

[15] [ENT16] ENTSO-E - Common Information Model (CIM), 2016. https://www.entsoe.eu/major-projects/common-information-model-cim/Pages/default.aspx

[16] [One15] OneM2M organization Technical Specification Functional Architecture TS-0001-V1.6.1 Jan. 2015

[17] [ETS13] ETSIM2M organization Technical Specification Machine-to-Machine communications (M2M); Functional architecture Dec. 2013.

[18] [ISOI17] ISO/IEC. Information technology – Internet of Things Reference Architecture (IoT RA). ISO/IEC DIS/CDV (committee draft for voting) 30141:20170628(E). ISO/IEC JTC 1/SC41/WG3

[19] [IEC] IoT 2020: Smart and secure IoT platform. 181p. Available online (accessed 10th Jan 2018) http://www.iec.ch/whitepaper/pdf/iecWP-IoT2020-LR.pdf

[20] [IoT12] Internet of Things – Architecture Consortium, *The IoT Architectural Reference Model (ARM) - D1.3*,

[21] European Commission, Luxembourg, 2012

[22] [ZVEI] ZVEI – German Electrical and Electronic Manufacturers' Association, *Industrie 4.0: The Reference*

[23] *Architectural Model Industrie 4.0 (RAMI 4.0)*, Frankfurt am Main, 2015.

[24] [IIC15] Industrial Internet Consortium, *Industrial Internet Reference Architecture (Version 1.7)*, Object

[25] Management Group, Needham, MA, US, 2015.

[26] [ETS17] SmartM2M Virtualized IoT architectures with Cloud Back-ends. ETSI TR 103.527 V0.2.1 (2018-01)

[27] [ITU12] ITU-T Study Group 13, *Next Generation Networks – Frameworks and Functional Models: Overview*

[28] *of the Internet of Things*, International Telecommunication Union, Geneva, 2012.

[29] [FIW16] Fiware open specifications. Online documents (Accessed 13th Jan 2018). https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Summary_of_FIWARE_Open_Specifications

[30] [Ama18] Amazon Web Services. Online documents (Accessed 13th Jan 2018). https://aws.amazon.com/documentation/iot/

[31] [Guth16] Guth J., Breitenbucher U., Falkenthal M., Leumann F., Reinfurt L. Comparison of IoT Platform Architectures: A field study based on a reference architecture. CIoT'16.
[32] [Bur18] Burg A., Chattopadhyay A., Lam K.-Y. Wireless communication and security issues for cyber-physical systems and internet of things. Proceedings of the IEEE Vol 106. No.1 Jan 2018.
[33] [OMTC18] Open MTC documentation. Online documents (Accessed 13th Jan 2018). https://www.openmtc.org/about.html
[34] [OM218] OM2M connecting things. Online documents (Accessed 13th Jan 2018) http://www.eclipse.org/om2m/
[35] [CIM2016] Common Information Model (CIM) standards, http://wiki.cimtool.org/, consulted on January 2018
[36] [DROOLS2017] Drools Language documentation, https://docs.jboss.org/drools/release/6.5.0.Final/drools-docs/html_single/index.html#DroolsLanguageReferenceChapter, consulted on January 2018
[37] [EFI2017] Energy flexibility interface, http://flexiblepower.github.io/technology/efpi/, consulted on January 2018
[38] [SEAS2016] SEAS Ontology. https://w3id.org/seas, consulted on January 2018
[39] [SAREF2015] Daniele L., den Hartog F., Roes J. (2015) Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology. In: Cuel R., Young R. (eds) Formal Ontologies Meet Industry. FOMI 2015. Lecture Notes in Business Information Processing, vol 225. Springer, Cham
[40] [TZE14] D. Trivellato, N. Zannone, and S. Etalle. GEM: A distributed goal evaluation algorithm for trust management. TPLP, 14(3):293–337, 2014.
[41] [TZGSE13] D. Trivellato, N. Zannone, M. Glaundrup, J. Skowronek, and S. Etalle. A semantic security framework for systems of systems. Int. J. Cooperative Inf. Syst., 22(1), 2013.
[42] [MAXIME2017] Lefrançois, Maxime. "Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Paaerns." Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT. 2017
[43] [SGAM12] CEN-CENELEC-ETSI Smart Grid Coordination Group (SG-CG), Smart Grid architecture model, 2012. https://www.cencenelec.eu/standards/Sectors/SustainableEnergy/SmartGrids/Pages/default.aspx
[44] [OSGP] OPSG Alliance & ETSI, Open Smart Grid Protocol. http://www.osgp.org/en/technical
[45] [IEC61850] IEC, IEC 61850. http://www.iec.ch/smartgrid/standards/
[46] [EFPI] Flexiblepower Alliance Network (FAN), Energy Flexibility Platform & Interface (EF-Pi). http://flexible-energy.eu/ef-pi/
[47] [EFI] Flexiblepower Alliance Network (FAN), Energy flexibility interface. https://github.com/flexiblepower/efi
[48] [PM] Flexiblepower Alliance network (FAN), PowerMatcher. https://fan-ci.sensorlab.tno.nl/builds/powermatcher-documentation/master/html/
[49] [USEF] USEF Foundation, Universal Smart Energy Framework. https://www.usef.energy/
[50] [BLOV] https://www.bluetooth.com/what-is-bluetooth-technology
[51] [BLSP] https://www.bluetooth.com/specifications
[52] [BLRE] https://www.bluetooth.com/develop-with-bluetooth
[53] [OGC] http://www.opengeospatial.org/
[54] [TRU-IoT] https://www.trusted-iot.org/

3 State of the Art Analysis on M2M Information systems

3.1 Technologies

3.1.1 Knowledge Representation

A basic possibility existing today for domain modelling as well as for service logic representation in a World Model are the W3C standard ontological knowledge base representations (RDF, RDF Schema, OWL, OWL-S [2-5]. A basic Service Request on such environment and corresponding simple service template or building block declaration can be realised by SPARQL SELECT and CONSTRUCT query statements, respectively [6].

Such ontological approach is already quite powerful in expressing semantic relations and dependencies. One could support long standing SPARQL SELECT queries as Service Requests that express e.g. a goal of *conditional energy demand activation when prices and various generators are under particular load conditions*, and derive from the knowledge base the implied sensors and actuators to be used. Provided that all concepts, sensors and actuators are referencing executable code for each needed operation on the implied data streams, this can be translated to a Service Execution Request that, when consecutively executed in the M2M Service Platform (at the lower layer, WP3), binds the right sensor and actuator instances to a flow graph of stream processing operator instances, thus fulfilling the stated high level goal of the initial Service Request.

3.1.1.1 Probabilistic Relational Models

Beyond standard RDF, the World Model may need to express **stochastic dependencies** between the observable information, so that service execution can take also this into account, e.g. for **optimising resource use** both in the application domain as well as in the M2M service execution and network infrastructure.

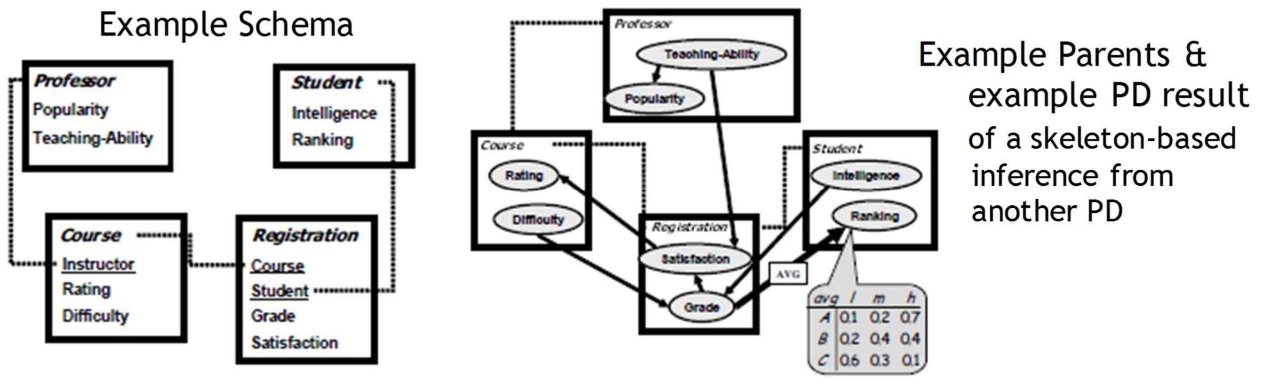


Figure 3.1Example PRM: (left) example schema, (right) inference from parent variables in a skeleton

An interesting richer representation of the World Model could therefore be found in Probabilistic Relational Models (PRM), which are a kind of mix between a semantic knowledge graph and Bayesian Networks [7]. Figure 2.1 shows a graphical representation of a simple example consisting of a university world with professors, students, courses and registrations. Given that there are also operators relating the statistics of one random variable in such a scheme to another (e.g. students may be ranked according to an aggregate value of their grades for courses they registered to, or teaching-ability of a professor may have a direct relation implying the professor's popularity and student satisfaction, in turn influencing the average rating that students give to a course), a probability distribution of one semantic stochastic variable in the model implies probability distributions (PD) for other variables in the schema.

In general as in the example, given a relational schema consisting of classes **X** as ground types and attributes **X.A** as properties or reference relations between attributes across classes, there is in addition a set of parent relations **Pa(X.A) = g(X.K.B)** where **K** represents a *slot chain* of attribute relations, and **g** is an *aggregate operator*. For each such **Pa (X.A)**, a conditional probability distribution (CPD) **P(X.A|Pa(X.A))** can be derived. Partial instantiation of PRM, called a relational *skeleton*, provides marginal probabilities over all remaining stochastic variables wherever there is a need to calculate them. In M2M services, this is particularly useful when the binding to actual sensors/actuators is determined at service execution time: the probabilistic properties of which sensors/actuators will most likely be selected may be a criterion that is e.g. used for operator placement within the to-be executed service graph.

An especially interesting fact is also that research can be found in literature concerning the *learning/population* of such PRMs based on probabilities determined from observed data [8].

3.1.2 Multi-agent Systems

A multi-agent system (MAS) is a computerized system composed of multiple interacting intelligent agents within an environment.

Multi-agent systems have been standing out in the last decades as a particularly interesting paradigm in what concerns the modelling and implementation of applications in dynamic and unpredictable environments [9].

Agent's usage is owed to a holistic vision, where the total is more than the simple sum of the parts. Therefore, multi-agent systems can be used in problems that present an intractable complexity to humans or single-agent systems. Using problem-decomposing techniques allows the computation's distribution, so that the resulting simpler problems can be mapped by a set of distinct agents, which interact in a way they can obtain satisfactory global solutions.

The fields of application of MAS approaches are vast [10-12], and the achieved results have shown to be valuable, principally when the modelled problems have a significant and visible impact in the reality. The facilitation of model enlargements, including new different types of agents in the society at any time; the analysis of agents' individual and internal performances; and the study of the interactions between the involved entities, are just some of the main advantages of using a MAS approach to model a real environment.

The power systems' sector is one of the main fields of application of MAS approaches [12-14]. The huge impact it presents on the global population, its high level of complexity, and the constant changes it is always subject to, are the main reasons.

3.1.2.1 MAS in Microgrids

MAS is a distributed computational intelligent approach suitable for modelling the operation of microgrids in short-term markets [15, 16]. It represents loosely connected network of interacting distributed intelligent hardware and software agents [16, 17]. The smart grid concept is also easily implemented in this type of platform, due to its ability in modelling the autonomous decision making entities. The MAS is used in this paper to model the short-term operation of integrated microgrids in a multi-market environment.

The model in [18] introduces agent-based approaches as alternatives to centralized control and management of microgrids. The practical implementation of integrated microgrids has been tested with a laboratory-based case study in [19], assessing the efficiency of the control and protection strategies in relation to the DG and ESS technologies. The short-term operation of a microgrid containing DGs and ESSs is modelled in [20], and it has been noted that the ESSs are more profitable when the MGCC plans to increase the share of intermittent units. Microgrids attain more benefits from resource sharing among electrically interconnected microgrids [15]. A hierarchical control scheme is used in [15, 16] for optimal scheduling of DGs and the power exchange between integrated microgrids and the power grid. An agent-based market-clearing model for DR exchange is proposed in [21].

3.1.2.2 MAS in Market Simulators

Simulator for Electric Power Industry Agents (SEPIA) [22] is a Microsoft Windows platform oriented simulator. It is based on a Plug and Play architecture, allowing users to easily create simulations involving several machines in a network, or in a single machine, using various processing units. SEPIA allows specifying the number of participating agents, as well as their behaviours, interactions, and changes during the simulation. The simulation developments can be followed and oriented by mechanisms for that purpose.

Power Web [23] is a Web-based market simulator, allowing the various participants to interact from very distinct zones of the globe. It is a rather flexible system that allows the definition of simulations with a large set of scenarios and rules. This simulator includes a centralized agent, acting as an independent system operator, to guarantee the reliability of the system, according to a defined group of entities, acting in various markets. It also allows users to enter an open market, competing against producers controlled by other users or computational algorithms.

The Short – Medium run Electricity Market Simulator (SREMS) [24] is based on game theory and is able to support scenario analysis in the short-medium term and to evaluate market power, in some situations. Some of its main features are: short-medium run (a time horizon of one month or multiples of it) simulation of electricity markets based on game theory, calculating price-makers optimal hourly bids; inelastic load, defined hour by hour and zone by zone; tree-like network with interzonal transit limits; monthly scheduling of reservoir hydro pumping storage plants; highly realistic representation of thermal plants; possible quota appointed to physical bilateral contracts, depending on producers share and risk attitude. It is particularly suitable to study the Italian electricity market.

The Electricity Market Complex Adaptive System (EMCAS) [25] uses an agent based approach with agents' strategies based on learning and adaptation. Different agents are used to capture the restructured markets heterogeneity, including generation, demand, transmission, and distribution companies, independent system operators, consumers and regulators. It allows undertaking Electricity Markets simulations in a time continuum ranging from hours to decades, including several Pool and Bilateral Contracts markets.

Agent-based Modelling of Electricity Systems (AMES) [26] is an open-source computational laboratory for the experimental study of wholesale power markets restructured in accordance with U.S. Federal Energy Regulatory Commission (FERC)'s market design. It uses an agent-base test bed with strategically learning electric power traders to experimentally test the extent to which commonly used seller market power and market efficiency measures are informative for restructured wholesale power markets. The wholesale power market includes independent system operator, load-serving entities and generation companies, distributed across the busses of the transmission grid. Each generation company agent uses stochastic reinforcement learning to update the action choice probabilities currently assigned to the supply offers in its action domain.

The Genoa Artificial Power Exchange (GAPEX) [27] is agent-based framework for modelling and simulating power exchanges. GAPEX is implemented in MATLAB and allows the creation of artificial power exchanges reproducing exact market clearing procedures of the most important European power-exchanges.

These are important contributions but, in general, lack flexibility as they adopt a limited number of market models and of players' methodologies. The evolution of some of these simulators is difficult to track but some of them, as is the case of AMES, are evolving in a very dynamic way. At the present state, it is important to go a step forward in Electricity Markets simulators as this is crucial for facing the changes in Power Systems. Increasing number and diversity of active players (due to high penetration of distributed energy resources and demand side participation) are a huge challenge [28].

3.1.2.3 MASCEM - MultiAgent System that Simulates Competitive Electricity Markets

MASCEM [29, 30] aims to facilitate the study of complex electricity markets. It considers the most important entities and their decision support features, allowing the definition of bids and strategies, granting them a competitive advantage in the market. Players are provided with biding strategic behavior so they are able to achieve the best possible results depending on the market context. MASCEM players include: market operator agents, independent system operator agents (ISO), market facilitator agents, buyer agents, seller agents, Virtual Power Player (VPP) [30] agents, and VPP facilitators.

MASCEM allows the simulation of the main market models: day-ahead pool (asymmetric or symmetric, with or without complex conditions), bilateral contracts, balancing market, forward markets and ancillary services. Hybrid simulations are also possible by combining the market models mentioned above. Also, the possibility of defining different specifications for the market mechanisms, such as multiple offers per period per agent, block offers, flexible offers, or complex conditions, as part of some countries' market models, is also available. Some of the most relevant market models that are fully supported by MASCEM are those of the Iberian electricity market – MIBEL, central European market – EPEX, and northern European market – Nord Pool. Some other market types can be provided by different external systems, by using an upper-ontology, which defines the main concepts that must be understood by agents that participate in power systems and electricity markets' related simulations.

Simulation scenarios in MASCEM are automatically defined, using the Realistic Scenario Generator (RealScen) [31]. RealScen uses real data that is available online, usually in market operators' websites. The gathered data concerns market proposals, including quantities and prices; accepted proposals and established market prices; proposals details; execution of physical bilateral contracts; statement outages, accumulated by unit type and technology; among others. By combining real extracted data with the data resulting from simulations, RealScen offers the possibility of generating scenarios for different types of electricity markets. Taking advantage on MASCEM's ability to simulate a broad range of different market mechanisms, this framework enables users to consider scenarios that are the representation of real markets of a specific region; or even consider different configurations, to test the operation of the same players under changed, thoroughly defined scenarios [31]. When summarized, yet still realistic scenarios are desired (in order to decrease simulations' execution time or facilitate the interpretation of results), data mining techniques are applied to define the players that act in each market. Real players are grouped according to their characteristics' similarity, resulting in a diversity of agent types that represent real market participants.

In order to allow players to automatically adapt their strategic behaviour according to the operation context and with their own goals, a decision support system has been integrated with MASCEM. This platform is ALBidS (Adaptive Learning Strategic Bidding System) [32], and provides agents with the capability of analyzing contexts of negotiation, allowing players to automatically adapt their strategic behaviour according to their current situation. In order to choose the most adequate strategy for each context, ALBidS uses reinforcement learning algorithms (RLA), and the Bayesian theorem of probability. The contextualization is provided by means of a context definition methodology, which analyses similar contexts of negotiation (e.g. similar situations in the past concerning wind speed values, solar intensity, consumption profiles, energy market prices, and types of days and periods, i.e. business days vs. weekends, peak or off-peak hours of consumption, etc.). This contextualization allows RLAs to provide the most adequate strategic support to market players depending on each current context. ALBidS strategies include: artificial neural networks, data mining approaches, statistical approaches, machine learning algorithms, game theory, and competitor players' actions prediction, among others.

ALBidS is implemented as a multi-agent system itself, in which each agent is responsible for an algorithm, allowing the execution of various algorithms simultaneously, increasing the performance of the platform. It was also necessary to build a suitable mechanism to manage the algorithms efficiency in order to guarantee the minimum degradation of MASCEM execution time. For this purpose, a methodology to manage the efficiency/effectiveness (2E) balance of ALBidS has been developed [32].

3.1.3 Optimization techniques

The main purpose of optimization techniques is to solve complex problems in several areas.

The optimization problems typically have three fundamental elements.

1) objective function to be maximized or minimized;

2) a collection of variables, whose values can be manipulated in order to optimize the objective;

3) a set of constraints, which are restrictions on the values that the variables can take.

There are numerous optimization techniques proposed in the literature. Their choice must consider the type of problem to solve. There are several proposals for optimization techniques classification. The most common is grouping them into deterministic / heuristics or static / dynamic.

Deterministic optimization refers to techniques and methods in order to find the optimal solution. The deterministic optimization is not the best option to solve complex problems due to the hard computational process. On the other hand, heuristics optimization should be used when the problem is complex and its solution is based on probabilities. In this case the goal is to produce a solution, the more approximate the exact, within a reasonable time and that is good enough to solve the considered problem.

Static Optimization is the process of minimizing or maximizing the costs/benefits of some objective function for only one instant in time. On the other hand, Dynamic Optimization refers to the process of minimizing or maximizing the costs/benefits of some objective functions over a period of time.

Another common classification in the literature to characterize the optimization techniques are the "Meta-heuristics". This kind of optimization technique is defined as a subset of heuristics techniques. Meta-heuristics unlike the heuristics does not focus on a specific problem. It is common to say that the heuristic exploits the problem-dependent information in order to find a 'good enough' solution to a specific problem, while meta-heuristics are like design patterns. The general algorithm idea is to be applied to a broad range of problems.

James Momoh [33] proposed a structure for organizing optimization techniques as shown in figure 2.2.

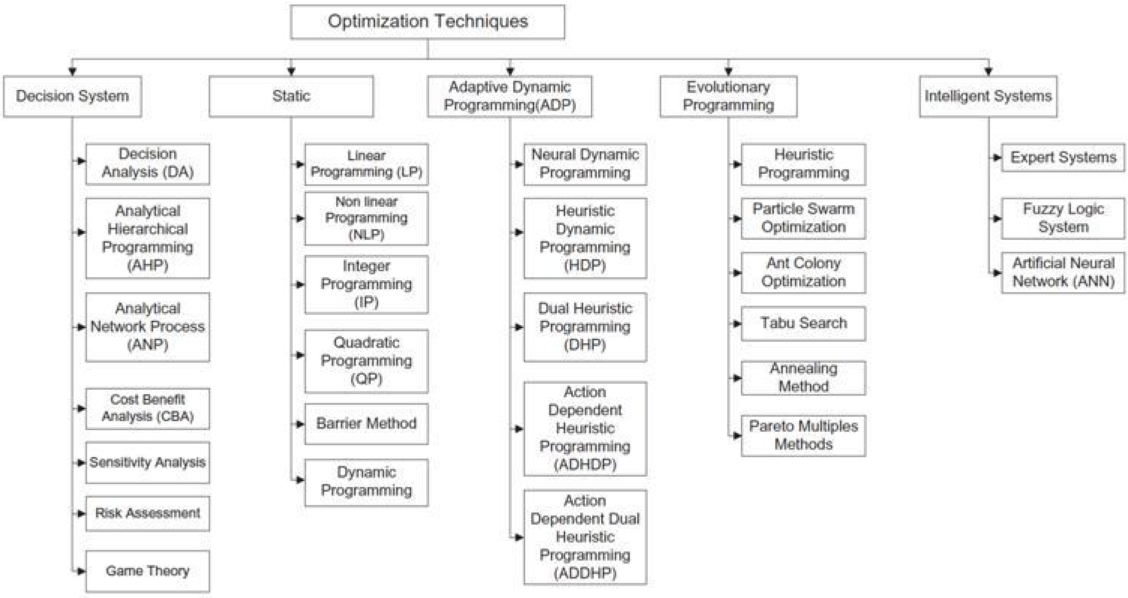


Figure 3.2 Optimization techniques [33]

Following are described in detail some of the main optimization techniques according to the classification of deterministic methods, heuristic and meta-heuristic. This description it is focused in the most generic techniques.

3.1.3.1 Deterministic Techniques

Linear Programming (LP)

Linear programming uses a mathematical model to describe the problem with linear objectives and linear constraints. The process to achieve the global optimum uses variants of the interior point method, or integer programming.

Nonlinear Programing (NLP)

NLP typically employs Lagrangian or Newtonian techniques for constrained and unconstrained optimization problems. The approach assumes that all objective functions are modeled as smooth and continuous functions. The procedure used in solving the NLP class of problems stems from identifying the feasibility of satisfaction. The sensitivity method, barrier method, and quadratic programming all feature in this class.

Mixed Integer Linear Programming (MIP)

MIP is a special case of LP where all or some of the decision variables are restricted to discrete integer values, for instance.The discrete values are limited to zero and one only (yes or no decisions, or binary decision variables). A branch-and-bound technique is usually used for this method.

Dynamic Programming (DP)

This approach is used to solve sequential or multistage decision problems. Mainly, it solves a multivariable problem by solving a series of single variable problems. This is achieved by *tandem* projection onto the space of each of the variables. That is, it projects first onto a subset of the variables, then onto a subset of these, and so on. This optimization technique is the good choice for handling time variability in the objective and constrained optimization problems.

Two common techniques derived from *Bellman* principles are

- Backward and forward recursion or table look up;
- Calculus based on composition.

Both techniques have the drawback of the curse of dimensionality and thus are not suitable for large and complex problems.

Stochastic Programming and Chance Constrained Programming (CCP)

Stochastic programming solves LP problems where the uncertainty assumption is so badly violated that the same parameters must be treated explicitly as random variables.

The two ways to handle LP with variability are:

- Stochastic programming (SP);
- Chance-constrained programming (CCP).

SP requires all constraints to hold with probability whereas CCP permits a small probability of validating any functional constraint.

Considering $Max\ Z = \sum c_j \cdot x_j$ between the limits 1 and n. It is possible to replace Z by some deterministic function, so, $E(Z) = \sum c_j \cdot x_j$ between the limits 1 and n. Similarly, the functional constraints $\sum a_{ij} \cdot x_j \leq b_j\ \ for\ i = 1,2,3, \dots, m$. The solution requires feasibility requirements of the constraints for all possible combinations of parameters in a_{ij}, b_j which must be assigned to x_j.

Mixed-Integer Nonlinear Programming (MINLP)

MINLP refers to mathematical programming with continuous and discrete variables and non-linearities in the objective function and/or constraints. The use of MINLP is a natural approach of formulating problems where it is necessary to simultaneously optimize the system structure (discrete) and parameters (continuous).

Methods for solving MINLPs include innovative approaches and related techniques taken and extended from MIP. Outer Approximation (OA) methods [34, 35], Branch-and-Bound (B&B) [36, 37], Extended Cutting Plane methods [38], and Generalized Bender's Decomposition (GBD) [39] for solving MINLPs have been discussed in the literature since the early 1980's. These approaches generally rely on the successive solutions of closely related NLP problems.

Solvers to handle optimization problems by type

The NEOS Server (http://www.neos-server.org/neos/solvers/) is a free internet-based service for solving numerical optimization problems, with tools for more than 60 state-of-the-art solvers in more than a dozen optimization categories. Some of the solvers are commercial products, available for free usage through the NEOS Server, other are open source code, which may or may not be available for download.

3.1.3.2 Heuristic Optimization

Artificial Neural Networks (ANN)

ANN are based on the natural genetics of the brain. Common techniques include back and forward propagation techniques. ANN have the ability to classify and recognize patterns in a large quantity of data through training and tuning of the algorithm. The key element of this paradigm is the novel structure of the information processing system, which comprises a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANN learn by example. They are configured for a specific application, such as pattern recognition or data classification, through a learning process. Back-propagation nets are probably the most well-known and widely applied of the neural networks in use today. ANN include, Adaptive learning, Self-organization, Real-Time Operation and Fault Tolerance via Redundant Information Coding.

Expert Systems (ES)

ES are used as a method of optimization that relies on heuristic or rule-driven decision-making. They are sometimes used for fault diagnosis with prescription for corrective actions. While the expert system/computer application performs a task that would otherwise be performed by a human, the method is only as reliable as the designed engineering rule-base. Figure 2.3 depicts the components.

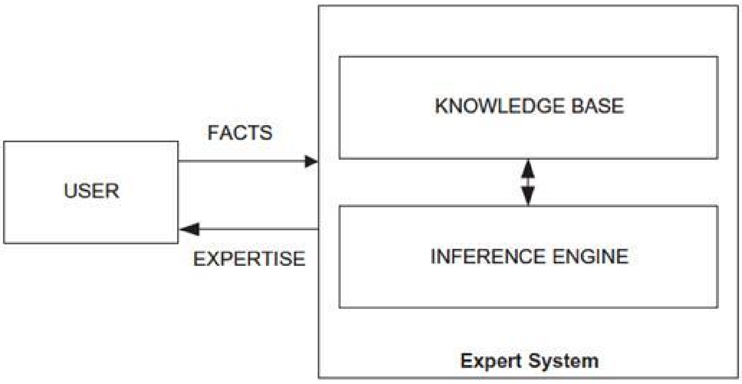


Figure 3.3 Fundamental components of an expert system [33]

Fuzzy logic

Fuzzy logic is a superset of conventional logic that has been extended to handle the concept of partial truth, which are the truth values between “completely true” and “completely false”. Special membership functions transform uncertainty in data to a crisp form for analysis. Fuzzy - logic control is often used to determine whether the process variables are within acceptable tolerances. A fuzzy set is a generalization of ordinary sets that allows assigning a degree of membership for each element to range from (0, 1) interval. Figure 2.4 depicts the simplified block diagram of the fuzzy logic approach.

Figure 3.4 Fuzzy logic approach diagram

3.1.3.3 Meta-Heuristic Optimization

Based on natural genetics, meta-heuristics solve combinatorial optimization problems. The techniques in this category, including particle-swarm, ant-colony, genetic algorithms (GA), and artificial intelligence, learn or adapt to new situations, generalize, abstract, discover, and associate. The evolutionary algorithms use a population of individuals. An individual is referred to as a chromosome which defines the characteristics of individuals in the population. The characteristic of each individual is termed a gene. Individuals with the best survival capabilities have the best chance to reproduce [40]. Offspring are generated by combining parts of the parents, a process referred to as crossover. Each individual in the population can also undergo mutation which alters some of the bits of the chromosome.

Genetic Algorithm (GA)

GA mimics biological evolution such that the elements in the algorithm are synonymous with genetic system terminology. Figure 2.5 shows a typical GA in which offspring are produced from selected parents, modified through crossover or mutation, and evaluated to find the fittest offspring. They are placed in the population to become parents while the unfit offspring are discarded. The process can be repeated until a suitable offspring or solution is created.

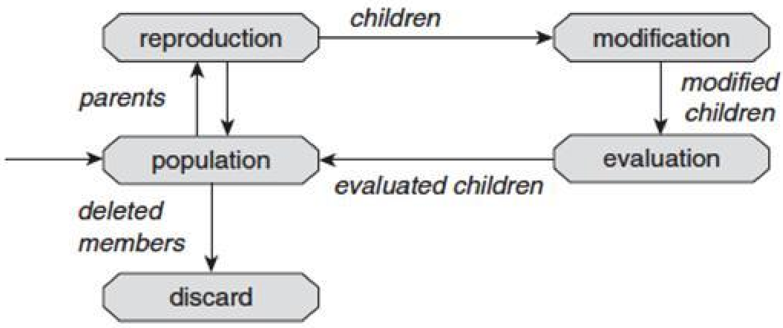


Figure 3.5 GA cycle [40, 41]

Particle Swarm Optimization (PSO)

(PSO) is a population - based stochastic optimization technique developed by Eberhart and Kennedy in 1995. It uses a simple mechanism that mimics the social behavior of bird flocking and fish schooling to guide the particles' search for globally optimal solutions. Individuals in a particle swarm follow a very simple behavior: to emulate the success of neighboring individuals and their own successes. The collective behavior that emerges is that of discovering optimal regions of a high dimensional search space [42]. The population of the study is referred to as the population and each individual is termed a particle

Two fundamental variants of PSO are Local Best (lbest) and Global Best (gbest). The lbest provides a local best, utilizing a ring network topology and a subset of the swarm as the neighborhood of the particle. The gbest presents the global best where the neighborhood for the particle taken to be the swarm.

Ant Colony Optimization (ACO)

ACO is a class which is applied to combinatorial optimization problems. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions [40, 43]. ACO uses computational concurrent and asynchronous agents termed a colony of ants that move through states of the problem corresponding to the partial solutions.

An ACO algorithm includes trail evaporation and daemon actions. The trail evaporation action decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Conversely, daemon actions implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information used to decide whether to bias the search process from a nonlocal perspective [40].

Non-Dominated Sorting Genetic Algorithm (NSGA-II)

In a multi-objective problem, the concept of optimality relates to a set of solutions rather than to a single solution [44]. There are several mathematical and evolutionary algorithms to find the non-dominated solution of a multi-objective optimization problem [45]. The NSGA-II algorithm is based on the concept of non-dominance. It is widely used in multi-objective optimization problems to create non-dominated solutions. Detailed description of NSGA-II can be found in [46].

The multi-objective evolutionary algorithm is accepted by the industry to solve real-world engineering problems. It can cope adequately with the difficulties of solving nonlinear, non-convex and mixed integer optimization problems [47, 48]. Multiple Pareto-optimal solutions can be achieved in one simulation run of NSGA-II algorithm. Its elitist selection mechanism prevents the loss of good solutions once they have been found [44]. In NSGA-II, the initial randomly generated parent population is sorted into different Pareto fronts based on the non-dominated concept [49]. Front 1 is a set of non-dominated solutions in the whole population, front 2 is a set of non-dominated solutions ignoring front 1, and so on, until classifying the whole population to different fronts [45]. Then the binary tournament selection algorithm is used to select the parent population [49]. The offspring population is achieved using the conventional genetic operators (crossover and mutation). In the next step, parents and offsprings are merged to form a collection and next generation is selected among this collection [49]. This process ends when the maximum number of iterations is reached.

In genetic algorithm, the parent individuals are selected from the population to create offspring individuals for the crossover and mutation operation. The crossover process improves the search capability of the genetic algorithm, and the mutation operator changes the offspring individual at a small probability after crossover process [50]. Mutation ensures the diversity of the population and prevents the immature convergence [50].

3.1.3.4 Adaptive Dynamic Programming (ADP)

ADP incorporates time dependency of the deterministic or stochastic data required for the future. ADP is also termed reinforcement learning, adaptive critics, neural – dynamic programming, and approximate dynamic programming [40, 42] (Figure 2.6). ADP consider the optimization over time by using learning approximation to handle problems that severally challenge conventional methods due to their very large scale and lack of sufficient prior knowledge. ADP overcomes the curse of dimensionality in DP. ADP is designed to maximize the expected value of the sum of future utility over all future time periods:

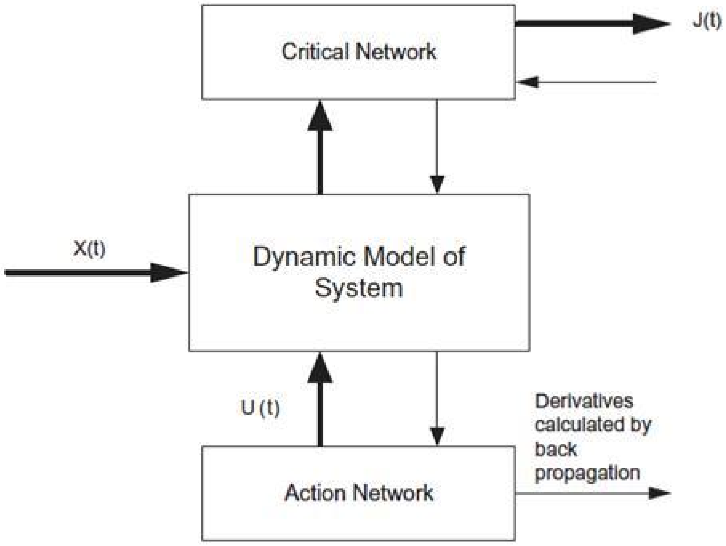


Figure 3.6 Adaptive dynamic programming structure [33]

3.1.3.5 Pareto Methods

Pareto analysis is used for solving the multi-objective optimization problem. The goal is to seek Pareto - optimal solutions through various approaches. Due to the conflicting nature of the objective functions based on the concepts of convergence and diversity, solutions to the above problem are both multiple and Pareto optimal. Two methods for solving such problems are:

- One-at-a-time strategy: A multi-objective optimizer may be applied repeatedly with the goal each time of finding one single Pareto - optimal solution.
- Simultaneous strategy: This approach utilizes Evolutionary Algorithms due to the population or archive-based approach to facilitate a parallel search.

3.1.3.6 Hybridizing Optimization Techniques and Applications

To facilitate the computational methods, intelligent systems such as immunized-neuro systems, immunized-swarm systems, neuro-fuzzy systems, neuro-swarm systems, fuzzy-PSO systems, fuzzy-GA systems, and neuro–genetic systems are considered.

3.1.4 Knowledge Discovery

This section presents the state of the art in knowledge discovery with special focus in data mining (DM) techniques, defining their concept in the context of the Knowledge Discovery in Databases (KDD) process. The main tasks of data mining, as well as the several used algorithms are presented.

Nowadays we live in the information and telecommunications technology era, computers and communications systems are responsible for changing the essence of people's work and, in addition, contribute to a general overhaul in the business world.

The traditional concept of business, which involved large amounts of papers and more or less complicated reports and decision making, in most cases, based on slight concrete facts, is gradually being replaced by the modern concept supported by the use of the most varied information technology.

For many years, most companies accumulated and gathered vast amounts of information, creating large amounts of data. Most of these companies were unaware how such information could prove useful for their business. What, in principle, it could be an advantage has shown as a problem as there was no ability to study such a high volume of data.

With the development of Decision Support Systems (DSS), supported by the information accumulated and stored in data warehouses, it was possible to drive the query operations and Decision Support (AD), enabling users the extraction of knowledge contained in these databases.

Currently, there are software tools to support decision making that facilitate the data warehouses access, enabling the data analyse even in complex contexts. The operation and analysis of large amounts of data automatically or semi-automatically to the discovery of patterns using the theories, concepts and algorithms are so-called data mining techniques. Knowledge Discovery systems are usually complex systems, so that they require strictness during the development process.

However, the study of large amounts of data for knowledge discovery extraction remains a difficult problem to solve. DM can be considered as a part of the Knowledge Discovery in Databases process.

According [51], the term KDD is used to represent the process to transform low-level data in high-level knowledge, whereas DM can be defined as extraction patterns models. Figure 2.7 illustrates the knowledge discovery in databases process enhancing the difference between KDD and DM.

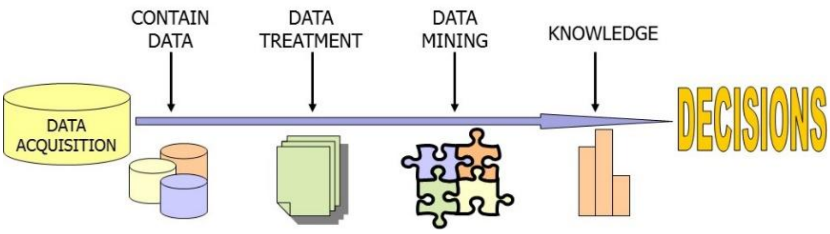


Figure 3.7 Knowledge Discovery in Databases (KDD) process

The KDD process is segmented into different stages, with different degrees of complexity:

- Selection data: This stage includes the definition of a data sample that will be analysed. It is at this stage that are defined which parameters to measure and collect, recorded time interval, etc.
- Pre-processing: Consist in the application of basic operations such as removal wrong data, detection and estimation missing records, reducing the data volume and normalization;
- Data Mining: In this stage is chosen and applied the DM algorithm to discover patterns and interesting relationships among data, such as regression, classification, artificial neural networks, clustering algorithms, etc. The model definition and parameters to be applied is sometimes difficult to choose as these should be consistent with the objectives of the KDD process;
- Interpretation: Once obtained the model and proceeding to its validation, using or not, the help of experts, it should be proved that the knowledge and consequent conclusions are valid and sufficiently satisfactory. It is also included the verification and resolution of any conflicts that might arise from this obtained new knowledge with previously known knowledge.

3.1.4.1 Data Mining Concept

The term data mining describes the discovery phase of the KDD process and cluster all different theories and technical research areas dealing with knowledge extraction from database.

The knowledge discovery from database focuses on the study of theories and algorithms for systems that extract and identify interesting patterns and relations among data. DM is a whole range of data analysis techniques applied to the problem of knowledge extraction contained and stored in large volumes of data that apparently have no meaning.

In [52] uses the term KDD to describe the process of knowledge operation from databases, while the application of DM algorithms is considered as just one-step in the whole process. Thus, data mining designates the task of finding out new knowledge generally unpredictable, relying on a database previously collected and properly adapted for this purpose.

Figure 2.8 depicts is the concept of DM expressing the technique of information extracting from databases aiming the support to decision making.

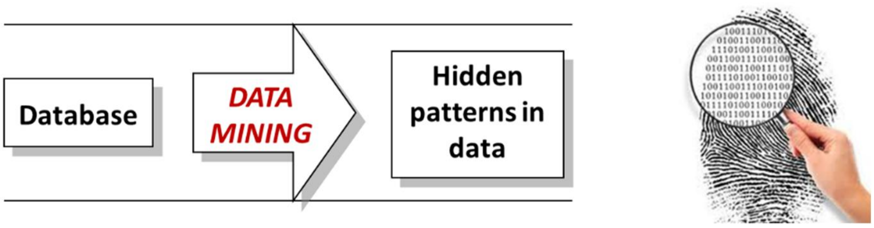


Figure 3.8 Data Mining purpose

The distance between DM and traditional statistic is extremely tenuous, given that both share a common goal, which is to discover relationships in the data, and many of the data analysis techniques used in DM tools are focused on statistical approaches. However, DM also makes use of ideas and other areas tools, such as the field of computing and artificial intelligence.

At times, the available data in the database may be non-linear and non-numeric. It is here that the DM operations differ from traditional statistical methods, since they are able to treat non-linearity and non-numeric data. Thus, most of DM techniques can be considered as an extension or mixture of basic statistical techniques.

In statistics field, preconditions must be checked, such as the distribution of the data must be known a priori, and the development of a particular hypothesis test, is done through the analysis process. Instead, in DM is assumed a distribution of source data and is generated a hypothesis based on historical data.

The previous step of DM phase is the data pre-processing. In this phase the data are “clean” and properly transformed to be used by DM algorithms. The goal of data pre-processing is to identify and estimate missing data, convert non-numeric data to numeric values, eliminate data in poor condition, convert units, etc.

Figure 2.9, based on [53] presents the several pre-processing techniques, not necessarily following the shown order. The data pre-processing is an important phase of the whole process of KDD, since the quality of decisions is based on data quality. This is also the step of the all KDD process that consumes more time to the analyst/researcher.

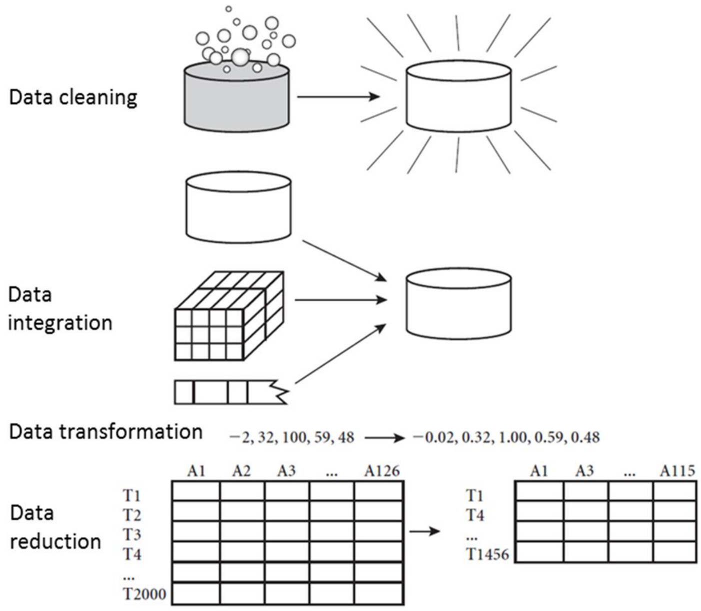


Figure 3.9 Data pre-processing forms [53]

3.1.4.2 Main tasks and data mining techniques

Data mining involves the use of methods (algorithms) to extract useful knowledge from a certain database. The DM algorithms usually perform unknown data prediction tasks and patterns description. The main data mining tasks are:

- Classification;
- Estimation or regression;
- Grouping by affinity or association;
- Segmentation and Clustering;
- Deviation analysis.

In this context, the task is a knowledge discovery problem to be solved. Pattern recognition operations can be distinguished by supervised and unsupervised operations.

In supervised operation is provided to the system the vector of features and its "label" (correct answer). Thus, considering a pair of variables {X, W}, X is a set of features and W the class to which it belongs. Thus, before performing the DM technique is made an identification of a particular specific objective.

Unsupervised operation: In this type of operation the DM algorithms “not know” the class that the records belong. How do not require a specified target in advance, these operations analyze a database and discover relationships among them.

There is not a single technique that solves all data mining problems. Different methods are used for different purposes, each method offers advantages and disadvantages in each application. It is, thus, required a certain expertise with DM different techniques to facilitate the selection of one of them, according to the problem that is presented.

The DM techniques come from different areas, such as statistics (descriptive statistics, decision trees and clustering), rules of associations and artificial intelligence (genetic algorithms, artificial neural networks and fuzzy sets). The DM techniques that are more known and used are:

- Decision trees;
- Regression;
- Artificial Neural networks;
- Genetic algorithms;
- Clustering algorithms;
- Nearest neighbourhood algorithms;
- Fuzzy Logic;
- Rough Sets.

The techniques used in DM contain a certain degree of automatic search, that is, algorithms are applied to, without users' interference, extract knowledge from data. Different algorithms may implement the same technique.

The choice of the DM technique to apply to a given problem is not considered an easy task, depending not only on the specific problem to be solved, but also on the data available for analysis. It is important to know the nature of the data in terms of content, type of attributes and relationships among records.

This choice can also be based on criteria for classification of techniques. Several classification schemes may be used to categorize DM techniques. A knowledge discovery system can be classified according to the type of data on which the DM techniques are applied.

Depending on the characteristics of data, such as attribute types, data quality, among others, and DM technique to be achieved, there are DM algorithms more suitable than other algorithms. The challenge is precisely to choose appropriately the algorithm to use, taking into account the type of existing data.

3.1.4.3 Data Mining: Application Areas

Data Mining came up with the main objective to support the decision-making in a large range of fields of study. Briefly, the main areas of interest in the use of DM techniques are:

- Marketing: discovery of consumer preferences and buying patterns. Knowing the consumer profile, the companies may be held promotional campaigns and offers specific products;
- Banking: detection of fraudulent credit card usage, identify "loyal" customers, predict good or bad customers, discover hidden correlations between different financial indicators, determine credit card expenses by customer group;
- Medicine: Characterization of patient behaviour to forecast visits, identification of medical therapies of success for different diseases, searching for patterns of new diseases;
- Transportation: definition of the best distribution schedules, analysis of load patterns;
- Government institutions: claims analysis, discovery of patterns in order to improve tax collection or discover fraud;
- Science and Technology: discovery of patterns in molecular structure, genetic data, global climate change and strategic planning production lines;
- Electric Power Systems: Identify typical load profile from electricity customers. Thus, with the use of DM techniques it is intended to discover knowledge concerning the patterns of electricity consumption by customers. The DM technique most appropriate for use depends on several factors, including the type of existing data and results to be achieved. It is therefore appropriate to conduct a comparative analysis of various algorithms to choose the one that demonstrates better performance. Others techniques includes the classification of new customers in one of the obtained clusters, electricity price forecasting, wind power prediction, the support of demand response programs and supporting the management of distributed energy resources.

The researcher of this field of study have to need a solid understanding of the database field, or a close collaboration with experts in order to be able to correctly select the data subsets, and the most interesting cluster patterns.

3.1.4.4 Machine learning

Machine learning is a field of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to adapt and expand when exposed to new data. Therefore, it can be said that "The goal of machine learning is to build computer systems that can adapt and learn from their experience" [54].

The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension - as is the case in data mining applications - machine learning uses that data to improve the program's own understanding. Machine learning programs detect patterns in data and adjust program actions accordingly. Thus "Learning denotes changes in a system that (…) enables a system to do the same task more efficiently the next time" [55], this way making computer systems more capable of solving problems, by adapting and being more self-sufficient, with the purpose of supporting humans in their tasks.

Learning is essential for unknown environments. When the system's designer is not acquainted with the complete aspects of the "world", by exposing a program agent to reality rather than trying to write it down (programming it), it becomes a useful method in system construction [56]. Also when the amount of information or knowledge is too large, it becomes impossible to be defined and encoded by humans, hence it becomes essential that the machine is capable of doing it itself. A similar situation is when the environments change over time, and when new knowledge about certain tasks is constantly being discovered by humans, it becomes impossible to re-design all systems by hand all the time.

Given that the environment is always evolving, adapting to that potentially dynamic environment is crucial. This adaptation can be performed by utilizing an implicit learning characteristic of evolution in the systems' algorithms. This way, by considering the algorithms' evolution, taking into account the own algorithm's perception of what it experiences, provided that "Learning is constructing or modifying representations of what is being experienced" [57], computer programs that learn and evolve in uncertain environments can be created.

There are several learning techniques that can be used to develop this kind of evolutionary algorithms, acquiring the necessary knowledge to the system, and choosing the significant information, making them more appropriate to each distinct situation. In [58] and [59] it is said that "Learning is the process that acquires knowledge to the system. It is hard to say that the machine is intelligent if it has no learning capability". However, these authors offer different opinions when categorizing the different learning techniques. In [59] learning techniques are classified into five distinct groups:

- rote learning;
- learning by taking advice;
- learning from examples;
- learning in problem solving;
- discovery learning.

One year later, in [58] two types of learning techniques are presented:

- coupling new information to previously acquired knowledge;
- digging useful regularity out of data.

Later on, in [60] learning techniques are categorized it into two major techniques: generalization and discovery learning. Being the first one divided into deductive and inductive learning.

Nowadays, the types of learning are usually grouped in:

- unsupervised learning:
  - clustering;
  - discovery;
  - genetic algorithms;
- supervised learning:
  - rote learning;
  - induction;
  - analogy;
- reinforcement learning.

These ideas are compared and confronted in Table 2.1, presenting the main points in which they converge and disagree.

Table 3.1 Machine learning techniques classification.

<table>
<tr><th>P. H. Winston
[58]</th><th>Rich and Knight
[59]</th><th>M. L. Ginsberg
[60]</th><th>Nowadays</th></tr>
<tr><td>coupling new information to previously acquired knowledge</td><td>rote learning</td><td>generalisation learning</td><td>rote learning</td></tr>
<tr><td>digging useful regularity out of data</td><td>learning by taking advice</td><td colspan="2">inductive learning</td></tr>
<tr><td></td><td>learning from examples</td><td>deductive learning</td><td>analogy</td></tr>
<tr><td></td><td>learning in problem solving</td><td></td><td>clustering</td></tr>
<tr><td></td><td colspan="3">discovery learning</td></tr>
<tr><td></td><td></td><td></td><td>genetic algorithms</td></tr>
<tr><td></td><td></td><td></td><td>reinforcement learning</td></tr>
</table>

As presented in Table 2.1, the discovery learning was the first technique to start being considered by the majority of authors. Discovery learning is a restricted form of learning that acquires knowledge without the help of a teacher. Unlike the other learning techniques, discovery learning can perform the tasks for which none, or little knowledge can be provided. In general, discovery learning can be broken down into three categories namely theory-driven discovery, data-driven discovery and clustering.

Afterwards, inductive learning started being consensual too. Inductive learning is the system that tries to induce a general rule based on observed instances. In other words, the system tries to infer an association between specific inputs and outputs. In general, the input of the program is a set of training instance where the output is a method of classifying subsequent instance. For example, the input of the program may be the colour of mushrooms where the output is classification that determines whether they are edible or not [61].

3.2 Domains of Application

3.2.1 Intelligent Power Systems

An intelligent power network is considered to be self-supporting, self-healing, adaptive and active, cost-effective, environmentally friendly, inherently secure and stable. It fulfils power quality requirements. It operates in a market environment and facilitates the exchanges of electrical energy between continuously varying, highly distributed producers and consumers.

An intelligent grid is also able to connect and to integrate all kinds of energy sources, including renewable and distributed energy sources which will play an increasingly important role in future energy systems. These resources can only be accommodated if the rather passive electricity supply network of today is transformed into an active, intelligent network with many (large and small) actors.

3.2.1.1 Smart Grids

"Smart grid" generally refers to a class of technology people are using to bring utility electricity delivery systems into the 21st century, using computer-based remote control and automation. These systems are made possible by two-way communication technology and computer processing that has been used for decades in other industries. They are beginning to be used on electricity networks, from the power plants and wind farms all the way to the consumers of electricity in homes and businesses. They offer many benefits to utilities and consumers, mostly seen in big improvements in energy efficiency on the electricity grid and in the energy users' homes and offices.

For a century, utility companies have had to send workers out to gather much of the data needed to provide electricity. The workers read meters, look for broken equipment and measure voltage, for example. Most of the devices utilities use to deliver electricity have yet to be automated and computerized. Now, many options and products are being made available to the electricity industry to modernize it.

The "grid" amounts to the networks that carry electricity from the plants where it is generated to consumers. The grid includes wires, substations, transformers, switches and much more.

Much in the way that a "smart" phone these days means a phone with a computer in it, smart grid means "computerizing" the electric utility grid. It includes adding two-way digital communication technology to devices associated with the grid. Each device on the network can be given sensors to gather data (power meters, voltage sensors, fault detectors, etc.), plus two-way digital communication between the device in the field and the utility's network operations center. A key feature of the smart grid is automation technology that lets the utility adjust and control each individual device or millions of devices from a central location.

The number of applications that can be used on the smart grid once the data communications technology is deployed is growing as fast as inventive companies can create and produce them. Benefits include enhanced cyber-security, handling sources of electricity like wind and solar power and even integrating electric vehicles onto the grid. The companies making smart grid technology or offering such services include technology giants, established communication firms and even brand new technology firms [62].

A comprehensive outlook of a smart grid infrastructure is illustrated in figure 2.10. Distributed generation (DG) units are represented such as offshore wind farm, solar plants, fossil fuel based power plants, and power-heat coupling units that may be usually found in any extended grid scenario. The SG infrastructure manages the generation parts of this scenario by remote monitoring and control subsystems that are considered as an "intelligent node" where the AMI or SCADA like operations are executed. A dedicated data center is usually operated to supervise the transmission and distribution networks. Another constituent of SG scheme involves smart homes and intelligent buildings through smart metering with their microgrids and several loads such as household electronics and electric vehicles (EVs) [63]. All these components require an accurately managed grid infrastructure from generation sites to consumption using the communication of measured values or using the transmitted control information more efficiently.

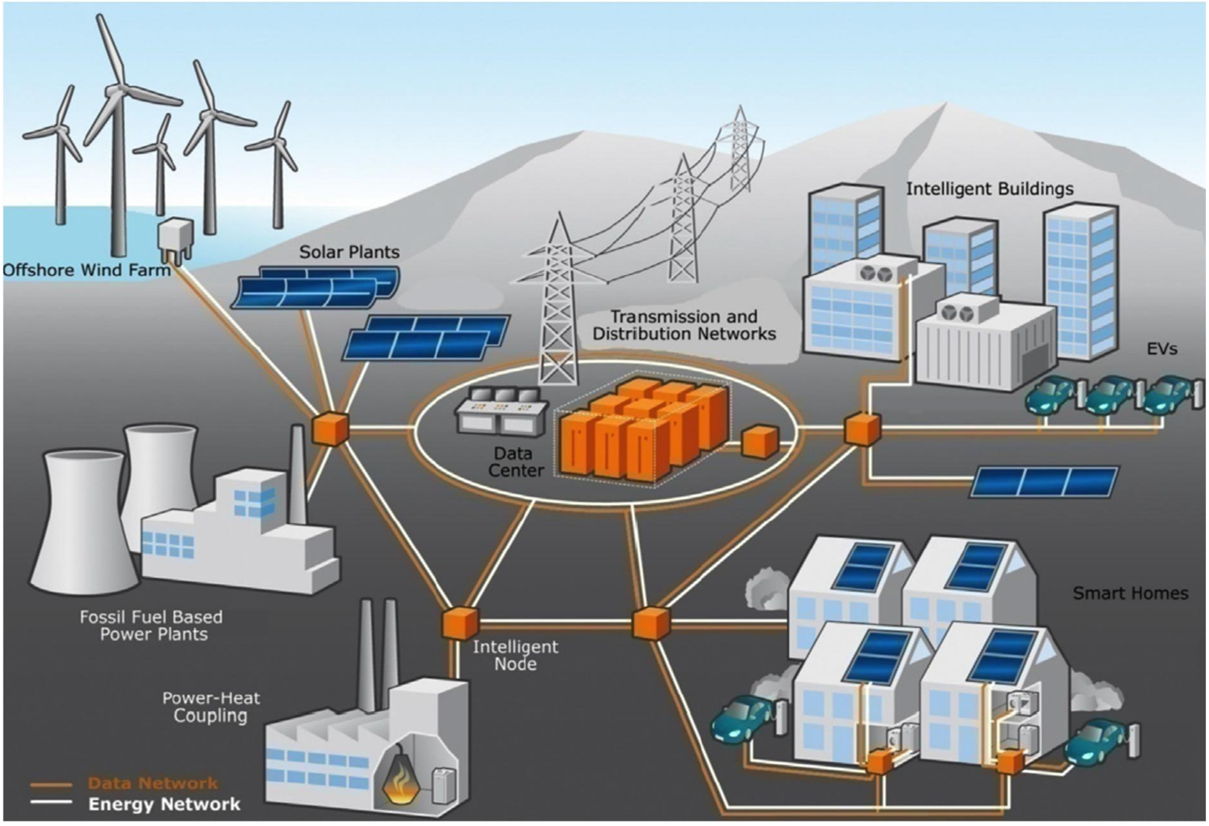


Figure 3.10 Complete schematic outlook of a smart grid infrastructure [64]

3.2.1.2 Main actors

According to NIST, in order to support planning and organization of the diverse, expanding collection of interconnected networks that will compose the Smart Grid, a correct approach is to divide the power systems in Smart Grid context into seven domains. Table 2.2 and figure 2.11 shows this idea.

Table 3.2 - Domains and actors in the Smart Grid conceptual model [65]

Domain	Actors in the domain
Customers	The end users of electricity. May also generate, store, and manage the use of energy. Traditionally, three customer types are discussed, each with its own domain: residential, commercial, and industrial.
Markets	The operators and participants in electricity markets.
Service Providers	The organizations providing services to electrical customers and utilities.
Operations	The managers of the movement of electricity.
Bulk Generation	The generators of electricity in bulk quantities. May also store energy for later distribution.
Transmission	The carriers of bulk electricity over long distances. May also store and generate electricity.
Distribution	The distributors of electricity to and from customers. May also store and generate electricity.

Each domain and its sub-domains include Smart Grid actors and applications. Actors include devices, systems, or programs that make decisions and exchange information necessary for performing applications (smart meters, solar generators, and control systems). Applications, on the other hand, are tasks performed by one or more actors within a domain. For example, corresponding applications may be home automation, solar energy generation and energy storage, and energy management. Figure 2.11 is a diagram that combines attributes of the seven domain diagrams.

In general, actors in the same domain have similar objectives. To enable Smart Grid functionality, the actors in a particular domain often interact with actors in other domains, as shown in figure 1. However, communications within the same domain may not necessarily have similar characteristics and requirements. Moreover, particular domains also may contain components of other domains. For instance, the ten Independent System Operators and Regional Transmission Organizations (ISOs/RTOs) in North America have actors in both the Markets and Operations domains. Similarly, a distribution utility is not entirely contained within the distribution domain - it is likely to contain actors in the operations domain, such as a distribution management system, and in the customer domain, such as meters. Underlying the conceptual model is a legal and regulatory framework that includes policies and requirements that apply to various actors and applications and to their interactions. Regulations, adopted by the Federal Energy Regulatory Commission at the federal level and by public utility commissions at the state and local levels, govern many aspects of the Smart Grid. Such regulations are intended to ensure that electric rates are fair and reasonable and that security, reliability, safety, privacy, and other public policy requirements are met. The transition to the Smart Grid introduces new regulatory considerations, which may transcend jurisdictional boundaries and require increased coordination among federal, state, and local lawmakers and regulators. The conceptual model must be consistent with the legal and regulatory framework and support its evolution over time. The standards and protocols identified in the framework also must align with existing and emerging regulatory objectives and responsibilities. The conceptual model is intended to be a useful tool for regulators at all levels to assess how best to achieve public policy goals that, along with business objectives, motivate investments in modernizing the nation's electric power infrastructure and building a clean energy economy.

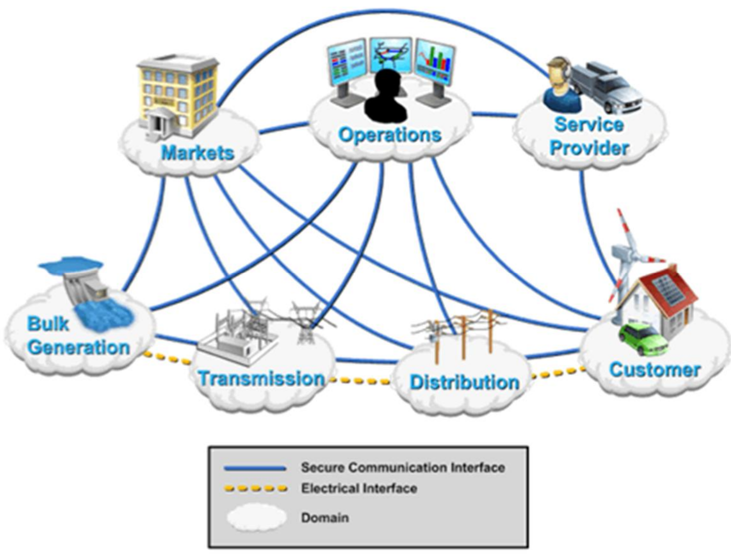


Figure 3.11 Interaction of actors in different Smart Grid Domains through Secure Communication Flows and Electrical Flows [66]

Bulk generation

Applications in the Bulk Generation domain (figure 2.12) are typically the first process in the delivery of electricity to customers. Electricity generation is the process of creating electricity from other forms of energy, which may vary from chemical combustion to nuclear fission, flowing water, wind, solar radiation and geothermal heat. The Bulk Generation domain is electrically connected to the Operations, Markets and Transmission domains. Some benefits to the Bulk Generation domain from the deployment of the smart grid are the ability to automatically reroute power flow from other parts of the grid when generators fail. Actors in the Bulk Generation domain may include generators of electricity in bulk quantities, as well as storage devices that can store energy for later distribution.

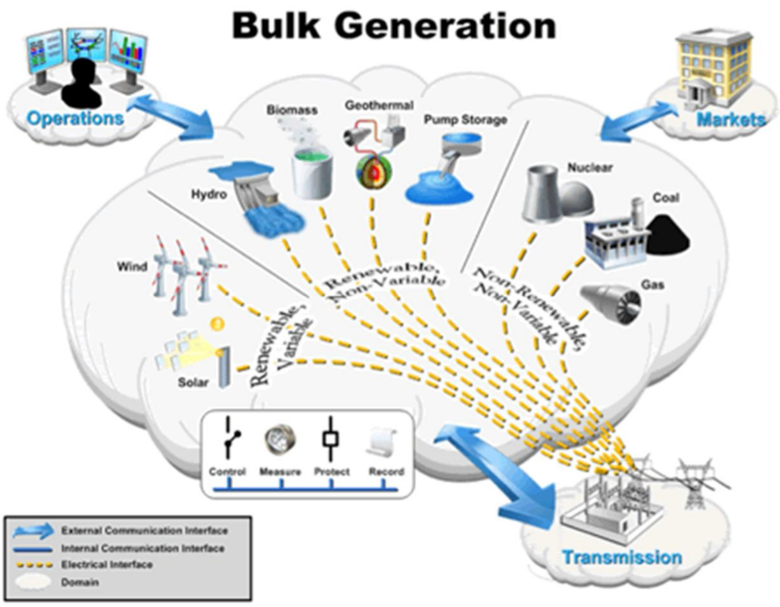


Figure 3.12 Bulk generation [66]

Transmission

Transmission (figure 2.13) is the bulk transfer of electrical power from generation sources to distribution through multiple substations. The Transmission domain is electrically connected to the Bulk Generation and Distribution domains, as well as communicating with the Operations, and Markets domains.

A transmission network is typically operated by a Regional Transmission Operator or Independent System Operator (RTO/ISO) whose primary responsibility is to maintain stability on the electric grid by balancing generation (supply) with load (demand) across the transmission network.

The Transmission domain may contain distributed energy resources such as electrical storage or peaking generation units. Energy and supporting ancillary services (capacity that can be dispatched when needed) are procured through the Markets domain and scheduled and operated from the Operations domain. They are then delivered through the Transmission domain to the utility-controlled distribution system and finally to customers.

Actors in the transmission domain may include remote terminal units, substation meters, protection relays, power quality monitors, phasor measurement units, sag monitors, fault recorders, and substation user interfaces.

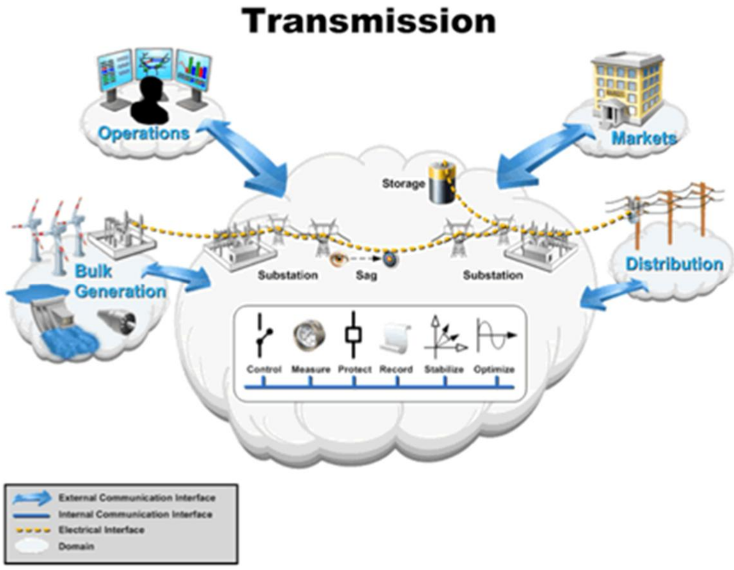


Figure 3.13 Transmission [66]

Distribution

The Distribution domain (figure 2.14) is electrically connected between the Transmission domain and the Customer domain at the metering points for consumption. The Distribution domain also communicates with the Operations and Markets domains

Historically distribution networks have little instrumentation installed, and there was very little communications within this domain that was not manually done by humans. Many communications interfaces within this domain were hierarchical and unidirectional.

With the advancement of distributed storage, distributed generation, demand response and load control, the ability of the Customer domain to improve the reliability of the Distribution domain exists. Distribution networks are now being built with much interconnection, extensive monitoring and control devices, and distributed energy resources capable of storing and generating power. Such distribution networks may be able to break into self-supporting "micro-grids" when a problem occurs and customers may not even be aware of it.

Actors in the Distribution domain may include capacitor banks, sectionalizes, recloses, protection relays, storage devices, and distributed generators.

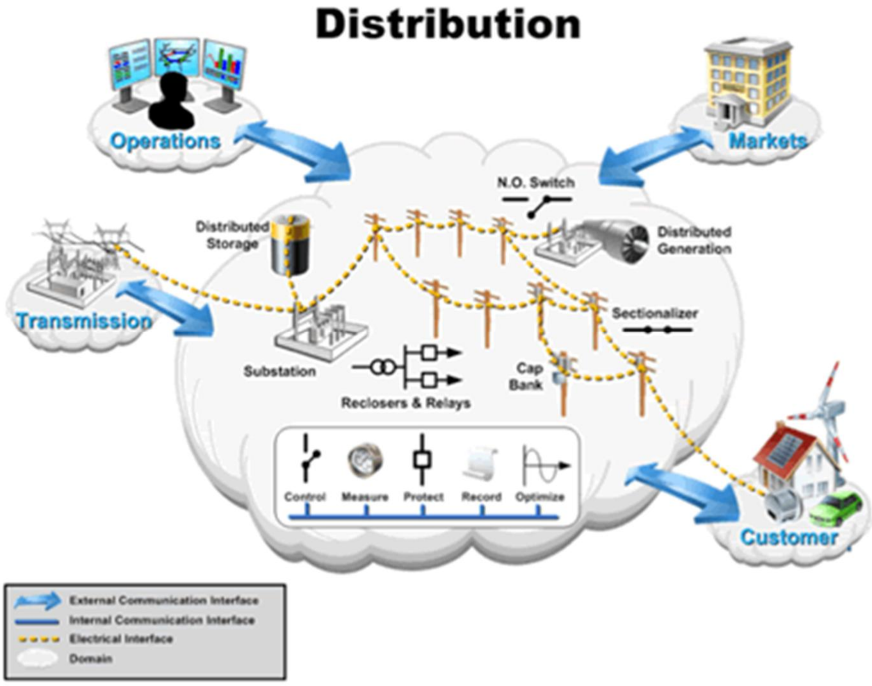


Figure 3.14 Distribution [66]

Customer

The Customer domain (figure 2.15) is electrically connected to the Distribution domain. It communicates with the Distribution, Operations, Markets, and Service Provider domains.

Actors in the Customer domain typically enable customers to manage their energy usage and generation. These actors also provide control and information flow between the Customer and the other domains. The boundaries of the Customer domain are typically considered to be the utility meter and/or an additional communication gateway to the utility at the premises.

There are three types of customers within the Customer domain: industrial, commercial/building, and home. The limits of these domains are typically set at less than 20kW of demand for Home, 20-200kW for Commercial/Building, and over 200kW for Industrial. The electric vehicle is an example of an actor that interfaces with all three domains.

All three domains (industrial, commercial and residential) have a meter actor and a gateway that may reside in the meter or may be an independent actor. The gateway is the primary communications interface to the Customer domains. It may communicate with other domains via AMI or another method such as the Internet. It typically communicates to devices within the customer premises using a home area network or other local area network. The gateway enables applications such as remote load control, monitoring and control of distributed generation, in-home display of customer usage, reading of non-energy meters, and integration with building management systems. It may also provide auditing/logging for security purposes.

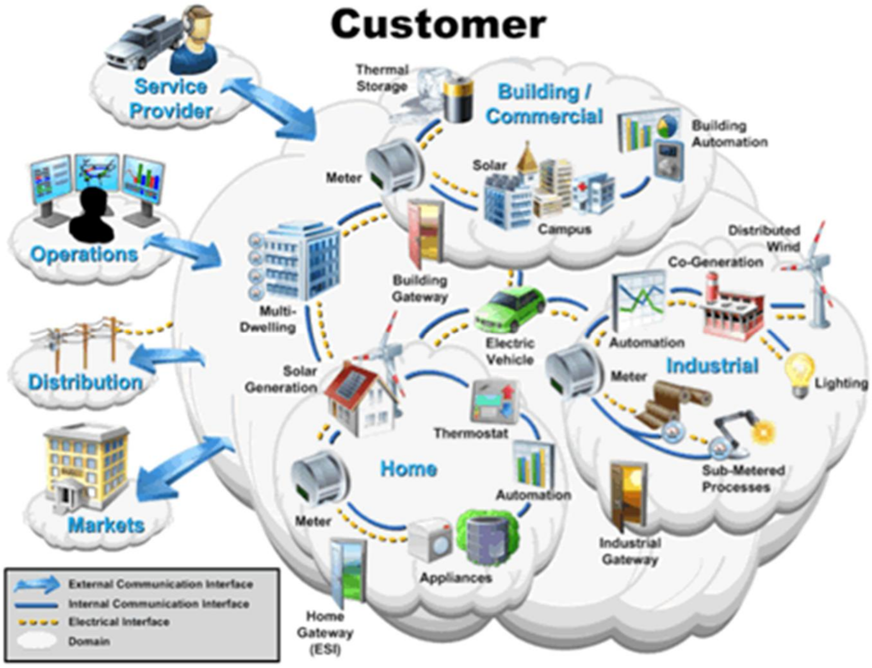


Figure 3.15 Customer [66]

Operations

Actors in the Operations domain (figure 2.16) perform the ongoing management functions necessary for the smooth operation of the power system. While the majority of these functions are typically the responsibility of a regulated utility, many of them may be outsourced to service providers and some may evolve over time. For instance, it is common for some customer service functions to be part of the Service Provider domain or Markets domains.

The typical applications performed within the Operations domain may include: network operation, network operation monitoring, network control, fault management, operation feedback analysis, operational statistics and reporting, real-time network calculation, dispatcher training.

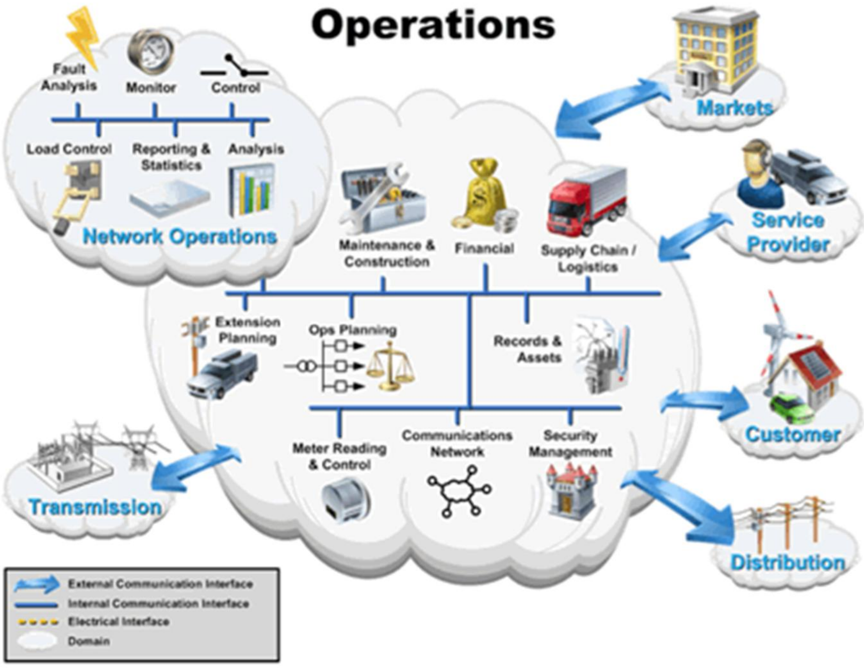


Figure 3.16 Operations [66]

Markets

Actors in the Markets domain (figure 2.17) typically perform pricing or balance supply and demand within the power system. The boundaries of the Markets domain are typically considered to be at the edge of the Operations domain where control happens, and at the domains containing physical assets (e.g. generation, transmission, etc.).

The interfaces between the Markets domain and those domains containing generation are most critical because efficient matching of production with consumption relies on markets. Besides the Bulk Generation domain, electricity generation also takes place in the Transmission, Distribution, and Customer domains and is known as distributed energy resources (DER). NERC CIPs consider suppliers of more than 300 megawatts to be Bulk Generation; most DER is smaller and is typically served through aggregators. DERs participate in markets to some extent today, and will participate to a greater extent as the smart grid becomes more interactive.

Figure 3.17 Markets [66]

Service provider

Actors in the Service Provider domain (figure 2.18) include the organizations providing services to electrical customers and utilities. That is, the actors in this domain typically perform a variety of functions that support the business processes of power system producers, distributors and customers. These business processes range from traditional core services such as billing and customer account management to enhanced customer services such as home energy generation and management. It is expected that service providers will create new and innovative services (and products) in response to market needs and requirements as the smart grid evolves. These emerging services represent an area of significant economic growth. Services may be performed by the electric service provider, by a third party on their behalf, or in support of new services outside of the current business models.

The boundaries of the Service Provider domain are typically considered to be the power transmission and distribution network controlled by the Operations domain. Services provided must not compromise the security, reliability, stability, integrity and safety of the electrical power network.

The Service Provider domain is typically electrically connected at the Customer domain. It communicates with the Markets, Operations and Customer domains. Of these, the interfaces to the Operations domain are critical for system control and situational awareness but the interfaces to the Markets and Customer domains are critical for enabling economic growth through the development of "smart" services. The Service Provider domain may, as an example, provide the "front-end" connection between the customer and the market(s).

Figure 3.18 Service provider [66]

The figure 2.19 provides a diagram to describe Smart Grid information networks; it allows identifying actors and possible communications paths, as well as useful way for identifying potential intra and inter-domain interactions and potential applications and capabilities enabled by these interactions.

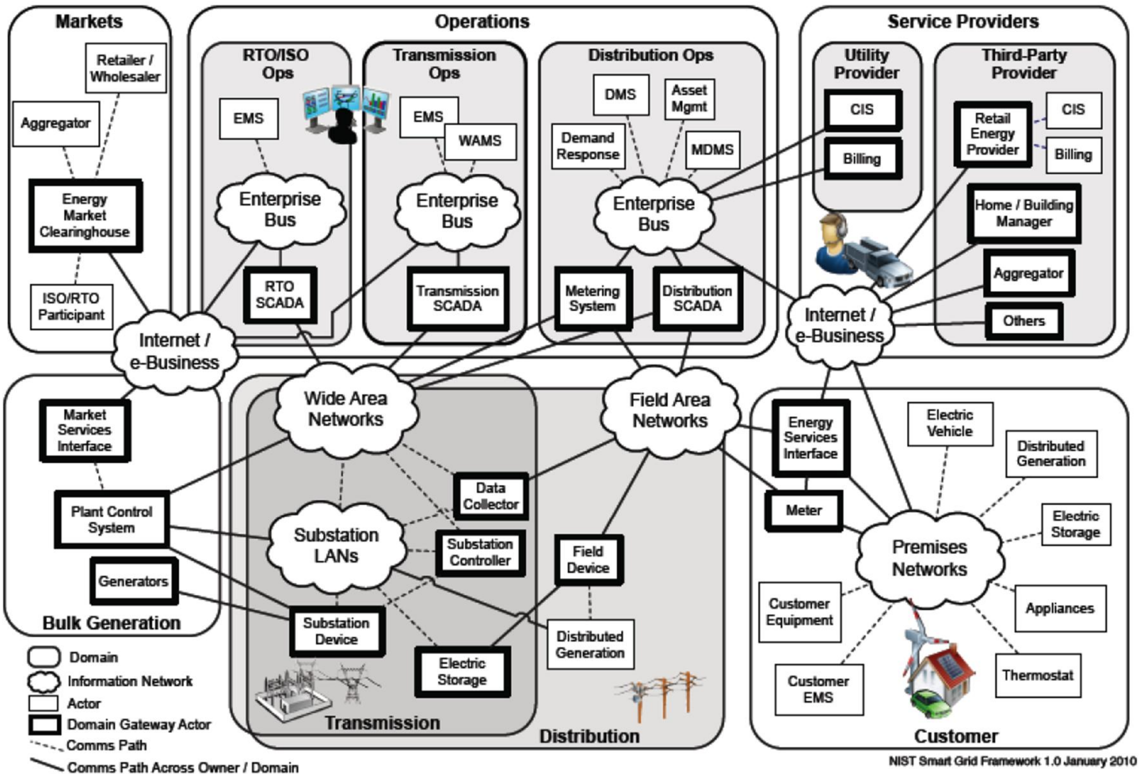


Figure 3.19 Conceptual reference diagram for Smart Grid information networks [65]

- **Domain:** Each of the seven Smart Grid domains (table 2.2) is a high-level grouping of organizations, buildings, individuals, systems, devices or other actors that have similar objectives and that rely on—or participate in—similar types of applications. Communications among actors in the same domain may have similar characteristics and requirements. Domains may contain sub-domains. Moreover, domains have much overlapping functionality, as in the case of the transmission and distribution domains. Transmission and distribution often share networks and, therefore, are represented as overlapping domains.
- **Actor:** An actor is a device, computer system, software program, or the individual or organization that participates in the Smart Grid. Actors have the capability to make decisions and to exchange information with other actors. Organizations may have actors in more than one domain. The actors illustrated here are representative examples but are by no means all of the actors in the Smart Grid. Each actor may exist in several different varieties and may actually contain other actors within them.
- **Gateway Actor:** An actor in one domain that interfaces with actors in other domains or in other networks. Gateway actors may use a variety of communication protocols; therefore, it is possible that one gateway actor will use a different communication protocol than another actor in the same domain, or use multiple protocols simultaneously.
- **Information Network:** An information network is a collection, or aggregation, of interconnected computers, communication devices, and other information and communication technologies. Technologies in a network exchange information and share resources. The Smart Grid consists of many different types of networks, not all of which are shown in the diagram. The networks include the Enterprise Bus that connects control centre applications to markets, generators, and with each other; Wide Area Networks that connect geographically distant sites; Field Area Networks that connect devices, such as Intelligent Electronic Devices (IEDs) that control circuit breakers and transformers; and Premises Networks that include customer networks as well as utility networks within the customer domain. These networks may be implemented using public (e.g., the Internet) and nonpublic networks in combination. Both public and non-public networks will require implementation and maintenance of appropriate security and access control to support the Smart Grid. Examples of where communications may go through the public networks include: customer to third-party providers, bulk generators to grid operators, markets to grid operators, and third-party providers to utilities.
- **Comms (Communications) Path:** Shows the logical exchange of data between actors or between actors and networks.

3.2.1.3 Distributed Energy Resource

With renewable energy prices falling and public concern for the environment rising, communities are driving for increased levels of renewable and distributed generation. The distributed and variable nature of renewables creates challenges for an electrical grid built for centralized generation and predictable loads. At low penetration levels, utilities may be able to ignore the effects of renewable and distributed energy resources (DER) but as levels rise, a new approach is required to integrate and manage the vast amount of DER that is driving the grid of the future.

The optimization of the distributed energy resource scheduling has been gaining increasing relevance due to the fostering of Distributed Generation (DG), storage, and flexible consumption (Demand Response - DR). such resources include wind and photovoltaic generation, electric vehicles, heat pumps, and so on, all related to the distributed level, for instance at the building level.

DR has gained widespread policy support in Europe, namely through the Third Energy Package, the Network Codes, and the Energy Efficiency Directive (EED) [67]. The current difficulties must be overcome and require new visions and further developments in order to widely integrate DR in the power and energy sector and on energy markets, namely in what concerns the participation of buildings that also include DG.

[68] is a relevant position paper concerning the independent aggregation in the European electricity markets. It refers to the EED mention on "Member States shall promote access to and participation of Demand Response in balancing, reserves and other system services markets (…) Such specifications shall include the participation of aggregators" as a clear motivation for the DR to participate alongside generation in electricity markets. Two most critical issues (regulatory and market model) on the electricity consumption and DR aggregation are referred in order to foster the integration of DR.

Several approaches concerning a framework for building controls aiming the success of energy transactions are presented in [69]. This reference guide prepared for the US Department of Energy by the Pacific Northeast National Laboratory (PNNL) proposes, among others, the use of DG, DR, and storage coordinated by an Independent Microgrid Operator (IMO). The IMO is able to make use of the resources owned by the building owner. Resource owners agree to pay (when consuming) or receive (when injecting energy) based on a Real-Time Price (RTP) that varies over time at short intervals (e.g., 5-min) based on a market clearing mechanism operated by the IMO. Such approach can be adapted for the context of several buildings operated by an aggregator (similar to the IMO) in order to enable the participation on the electricity market also. Different IMOs are also able to exchange energy between them.

3.2.1.4 Electricity Markets

All over the world, the restructuring of the power sector placed several challenges to governments and to the companies that are involved in the area of generation, transmission, distribution and retail of electrical energy. One of the most significant changes is the introduction of electricity markets, aimed at providing competitive electricity service to consumers [70]. Potential benefits, however, depend on the efficient operation of the market. Definition of the market structure implies a set of complex rules and regulations that should not encourage strategic behaviours that might reduce market performance.

Electricity Markets are not only a new reality but an evolving one as the involved players and rules change at a relatively high rate [71]. The emergence of a diversity of new players (*e.g*. aggregators) and new ways of participating in the market (distributed energy resources and demand side are gaining a more active role) are signs of this [72].

The restructuring turned electricity markets into an attractive domain for developers of software tools. Simulation and Artificial Intelligence techniques may be very helpful under this context.

Real-world restructured electricity markets are sequential open-ended games with multiple participants trading electric power. Market players and regulators are very interested in foreseeing market behaviour: regulators to test rules before they are implemented and to detect market inefficiencies; market players to understand market's behaviour and operate in order to maximize their profits.

Electrical energy generation is a distributed problem by its own nature. Traditionally, electricity was based on a reduced number of plants (*e.g*. nuclear, other thermal and hydro power plants). However, guaranteeing sustainable development is an enormous challenge for Power Systems. This requires a significant increase in distributed generation, mainly based on renewable sources [73]. However, this leads to a system that is much more complex to control, since it includes many more power generation plants, and the generation is more unpredictable than before, due to the difficulty in forecasting the energy production originated by some renewable sources (*e.g*. wind and photovoltaic).

Electricity markets put a new emphasis on the economic dimension of the problem. However, the basic infrastructure, namely the power system network, has a real physical nature, with specific limitations [74]. The introduction of electricity markets shows us the fragility of power systems infrastructures to operate in a competitive context. Several severe incidents, including blackouts, occurred (*e.g*. the 14th August 2003 Blackout in the US, and the 4th October 2006 quasi-blackout affecting nine European countries and some African nations as well).

Regulatory Models

The market environment typically consists of a pool, which may be symmetric or asymmetric, as well as a floor for bilateral contracts [75]. Additionally, balancing markets are also required. This implies that each market player must decide whether to, and how to, participate in each market type.

Besides the entities that negotiate in the market, namely electricity sellers and buyers, these types of market usually include also a market operator and a system operator. The market operator is the entity responsible for operating the market; it manages the pool using a market-clearing tool to establish the market price and a set of accepted selling and buying bids for every negotiation period. The system operator is usually responsible for the transmission grid management and for the technical constraints. Every established contract, either through bilateral contracts or through the pool, must be communicated to the system operator, who analyses its technical feasibility from the power system point of view.

Spot Market

The spot or day-ahead market [76], [77] is a daily basis functioning market, with the goal of negotiating electric power for each hour, or half an hour of the following day. This type of markets is structured to consider the daily production fluctuations, as well as the differentiated operation costs of production units.

One day is usually divided into 24 or 48 negotiation intervals [74], referring to one hour or half hour periods. The participating entities must present their selling or buying proposals for each of those periods. These proposals, or bids, are typically characterized by (in symmetric pools, the most common type of market clearing mechanism): in the case of a seller, the amount of power to be sold, and the minimum selling price; in the case of a buyer, the amount of desired power, and the maximum accepted price.

When the negotiation is finished, the market operator sets the dispatch for each period. This entity is responsible for the spot market functioning, initiating and controlling the full process. The market operator is also responsible for the definition of the market price, a unique price that is applied to all transactions.

Symmetric Pool

The symmetric pool is characterized by a competition between selling and buying entities. The price definition mechanism is usually based on a double auction.

In this type of pool, suppliers and consumers both submit bids. The market operator orders the selling and demand offers: selling bids start with the lowest price and move up, and demand bids start with the highest price and move down. Then, the proposed bids form the supply and demand step curves, and the point at which the two curves intersect determines the market price, paid to all accepted supplier and consumers. The bids of every supplier offering prices lower than the established market price and every consumer offering prices higher than the market price will be accepted.

Figure 2.20 shows the dispatch procedure of a symmetric market.

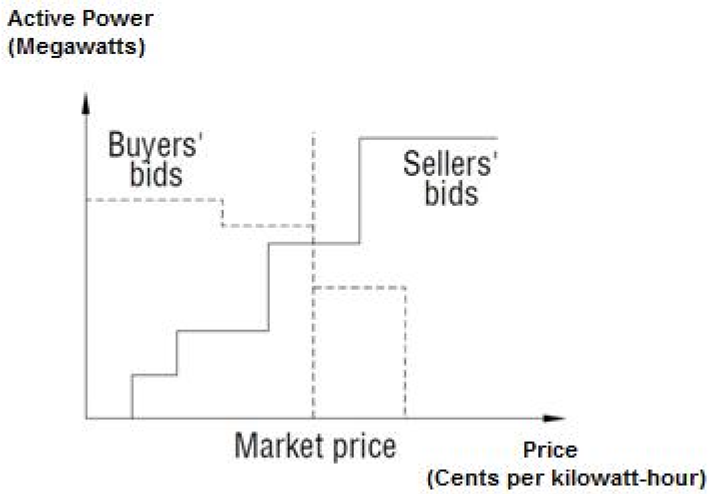


Figure 3.20 Symmetric Pool, adapted from [78].

Sellers and buyers submitting their bids in this type of markets is an indicator of the existence of behaviours with price sensitive consumptions. This observation is related to the dependence of this market's efficiency on the number of participating players, as well as on their proposals provision.

Asymmetric Pool

This type of pool allows the exclusive submission of selling proposals. In the asymmetric markets, the buyer players do not present bids, rather only a demand estimative. This model assumes that buyers are able to pay any price, which results from the market functioning, *i.e.,* the demand is considered inelastic.

In an asymmetric market, the sellers present their bids, and the market operator organizes them starting with the lowest price and moving up. The consumers reveal their needs to set up the demand. Once the market operator knows the demand, it accepts the suppliers' bids starting from the lowest and accepts as many as necessary to fill the demand. The market price - to be paid to all accepted suppliers - is that of the last accepted bid (the one with the highest price).

Figure 2.21 shows the dispatch procedure of a symmetric market.

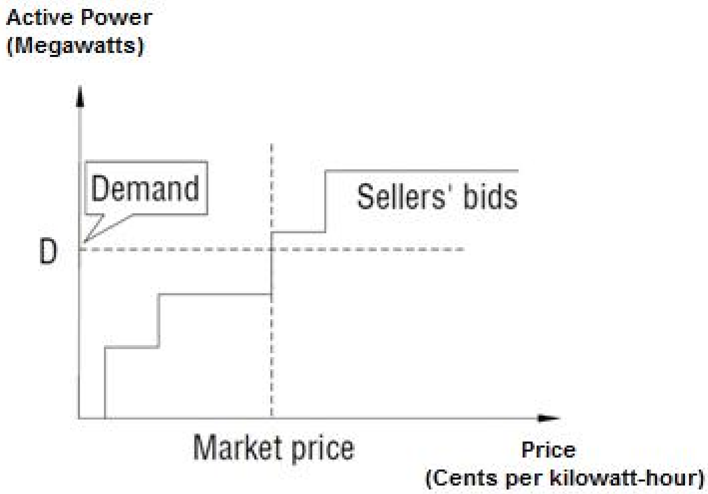


Figure 3.21 Asymmetric Pool, adapted from [78].

In this type of market, the prices are highly influenced by the proposed selling prices and by the level of demand.

Complex Market

The complex market [74], [79] provides the opportunity for the presentation of restrictions that allow players to leave the market if they are not respected, meaning that players are not interested in participating unless those conditions are met. In addition, to meet the requirements of the simple day-ahead pool, complex market includes at least one of the following conditions: *Indivisibility, Charge Gradient, Minimum Income* and *Scheduled Stop*. Market agents as strategies for achieving the highest possible profit also use complex conditions.

The *Indivisibility* condition allows setting a minimum value of operation in the first offer of each period. Below this value, the participation of the production unit on the market is not possible. This condition applies to generating units that cannot work under a technical limit.

The *Charge Gradient* condition allows establishing the maximum difference between the initial and the final power, between periods, for a production unit. This allows avoiding abrupt changes between consecutive periods (resulting from technical impossibility of achieving such changes).

The *Minimum Income* condition is used to ensure that the production unit does not enter the market if it cannot obtain a minimum amount in Euros (€), in the total of all periods, plus a variable fee per transacted kWh. This restriction depends on the sales strategy of each agent.

The *Scheduled Stop* condition is used in situations when the production unit has been withdrawn for not meeting the condition of required Minimum Income. This condition ensures that the production stopping is not done abruptly, rather undertaking a scheduled stop in a maximum time of 3 hours, avoiding production to immediately decrease to zero, from the last period of one day to the first period of the next. This is done by accepting the first offer of the first three periods as a simple offer, with the sole condition that the offered power is decreasing in each period, to smooth the production decrease until it gets to zero.

The market operator must assure the economical dispatch taking into account the specified conditions, which may imply the renegotiation of the period or day in matter, depending on the possible removal of entities that have presented competitive bids but whose complex conditions were not satisfied. In day-ahead market, only seller agents may present complex conditions.

Bilateral Contracts

Bilateral contracts allow players to negotiate directly with each other, outside the spot market. This provides opportunities for reaching advantageous agreements from an economic perspective, but also from a spatial one, when negotiating with players that offer benefits resulting from their location. It is also a good opportunity to establish contracts with varying timelines, resulting on higher security for companies that require constant demands over time.

Bilateral contracts are established through requests for proposals sent by buyers or traders - the demand agents. If a demand agent chooses to participate in the bilateral market, it will first send a request for electricity with its price expectations to the interested sellers. In response, a seller analyses its own capabilities, current availability, and experience. The seller must be sure that it is feasible to deliver energy to the buyer's location. Therefore, it must get the network operator's feedback before reaching an agreement with the demand agent. If the seller can make an offer according to the requested parameters, it formulates a proposal and sends a message to the demand agent. The demand agent evaluates the proposals and accepts or rejects the offers.

Balancing Market

The balancing market's [80], [81] goal is to take care of the necessary adjustments on the viable daily program and the last final hourly program, correcting possible deviations from forecasted production or consumption. It is a market of voluntary participation, in which offers can be presented by all agents qualified for the daily market who have participated in the corresponding session of the same market, which performed a bilateral contract, or whose units were unable to attend due to unavailability in the daily market and return to be available. It is, therefore, a complementary platform to the day-ahead market.

Although usually only sellers may, present complex conditions to the spot market, in the balancing market both sellers and buyers may present such conditions. Another important issue is that sellers may become buyers on the balancing market, while buyers may become sellers, depending on their forecasted production or consumption deviations, and on their strategies. Therefore, players are able to take advantage on this to define negotiation strategies, which contemplate the buying of a higher amount of power when the prices are lower. By knowing that, they can later sell it at higher prices; or vice-versa, selling more than they are capable of producing, when the prices are high, and then buying the extra amount on a complementary market in which the prices are lower.

Complementary Markets

Electric energy is one of the most commonly used forms of energy. However, a substantial part of its production results from the burning of fossil fuels, which originates greenhouse gases emission to the atmosphere, namely carbon dioxide, methane, nitrous oxide and sulphur hexafluoride, originating climatic changes with harmful consequences to the environment [82].

With the shortage perspective of the non-renewable resources and the increasing usage of electric energy, it becomes imperative to develop new methods of energy production, investing in technologies that contribute to a more energetically rational way of living.

The verified growth of the investment on distributed generation, namely in wind and photovoltaic technologies, has been creating new opportunities for the promoters that own such technologies. Besides the selling of electrical energy to the system operators or in energy markets, the promoters can develop their activity in other markets, such as the Carbon Market, the Green Certificates emission, or the selling of water steam and hot water, among others [83]. An alternative potential business is the integration with industries as livestock, the treatment of municipal solid waste, cork, in order to significantly reduce investment and/or operation costs.

These market mechanisms are complementary to the electric market, originating a more dynamic and alternative global market. The complementarity between such different types of markets creates the opportunity for players to improve their negotiating approaches, considering the investments in different markets.

The increasing complexity brought by the conception of such a diversity of market types resulted in high changes concerning the relationship between the electricity sector entities. It also resulted on the emergence of new entities, mostly dedicated to the electricity sector and electricity energy trading management. In what regards the commercial transactions, the analysis of different market mechanisms and the relationship between market entities becomes crucial. Namely in the case of Portugal, where the Iberian market, in partnership with Spain, has materialized not long ago, there are many aspects to analyze, improve, and even redefine. All market participants develop interactions among them, needing information systems for that purpose. As the observed context is characterized as being of significant adaptation and change, the need for decision support tools directed to this markets' analysis is also accentuated. Multi-agent based software is particularly well fitted to analyze systems with such characteristics.

3.2.2 Smart Mobile

3.2.2.1 Smart Cyber Physical

Illustrate a large range of problematics from electronics and sensor platforms, low power wearable devices, short and long-range low power communication methods in various environments on the move.

3.2.2.2 ETSI M2M mobile network

The ecosystem is horizontal and vertical, i.e. horizontal service layers, vertical use cases. The standard model for this horizontal service environment is OneM2M, ETSI standard for M2M, which aims to provide an environment where developers can develop applications for M2M devices, regardless of the device type, the operating system it runs, and the network serving it. The OneM2M architecture requires adaptors to the specific networks, and ETSI defined the MTC-IWF gateway as an adapter that bridges the OneM2M generic service layer with the mobile core network. In order to support OneM2M, the mobile operator must have an Interworking Function, for conversion between OneM2M protocols and the mobile network protocols.

The interworking function is a standard network element, defined by ETSI, which is part of the OneM2M ecosystem. OneM2M is a horizontal layer architecture that aims to encompass all M2M manufacturers and providers, with the goal of opening the M2M devices for application developers via standard APIs, independently of the network type. The IWF GW is designed to bridge between the generic OneM2M environment and the specific mobile network architecture, by converting triggers and events and provides APIs that expose the core network capabilities to the M2M verticals (service providers).

The OneM2M architecture should server multiple network types, such as wireless (GSM, CDMA, LTE, and Wi-Fi), wireline, and short range/personal networks. There is a need for adapters that connect the generic Network Service Capabilities Layer (NSCL) to the various network architectures. The IWF GW is such an implementation, for the cellular network.

The figure 2.22 is taken from the 3GPP standards. And describes the MTC-IWF component, which bridges between the mobile network (at the left side) and the SCS (Service Capabilities Server), usually installed at the cloud, outside of the mobile operator premises.

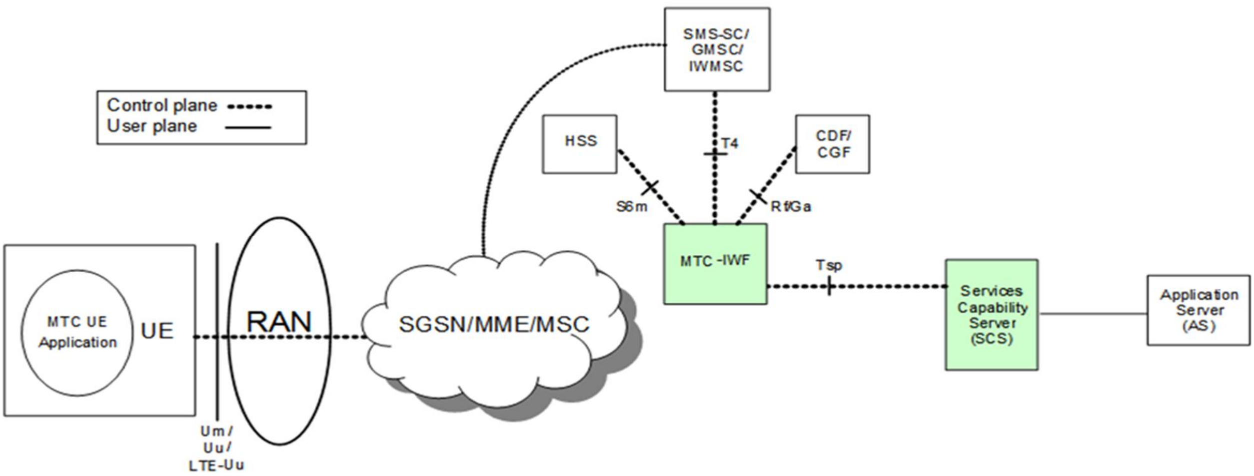


Figure 3.22 ETSI M2M mobile network overview

Triggering via MTC-IWF gateway:

The main functionality of the MTC-IWF is to bridge between the generic SCL (Service Control Layer) and a specific network (in our project – the mobile network) for:

- Hiding the internal topology of the mobile network

- Receiving triggers from the SCS, transmits to the device and reports back
    - Triggers are requested to trigger the device to open an IP channel with the server
- Providing load control for triggers

The triggers are a main issue for MTC-IWF. That means triggers an M2M device to create an IP connection (while it is not connected):

- It allows an MTC server to trigger registered devices (i.e. IMSI attached or GPRS attached) without a PDP connection to establish a connection making communication with the MTC server possible.
- The MTC-IWF would use a standardized protocol over the MTCsp interface point.
- The role of the MTC-IWF is to hide the details of the triggering mechanism in the operator' domain and provide the MTC Server a generalized interface for it to make a device triggering request.

3.2.2.3 Cloud 3D sensing

Softkinetic is designing a game that stimulates collaboration between multiple users (customers/end-users), where the input of the games is gesture based. The gestures are detected based on information from 3D smart camera that communicates to M2M Grid cloud services – the data is computed on the platform through a cloud base services. The main purpose of the games is to accomplish goals of the cooperative game play over M2M cloud services, including manipulation of remote objects.

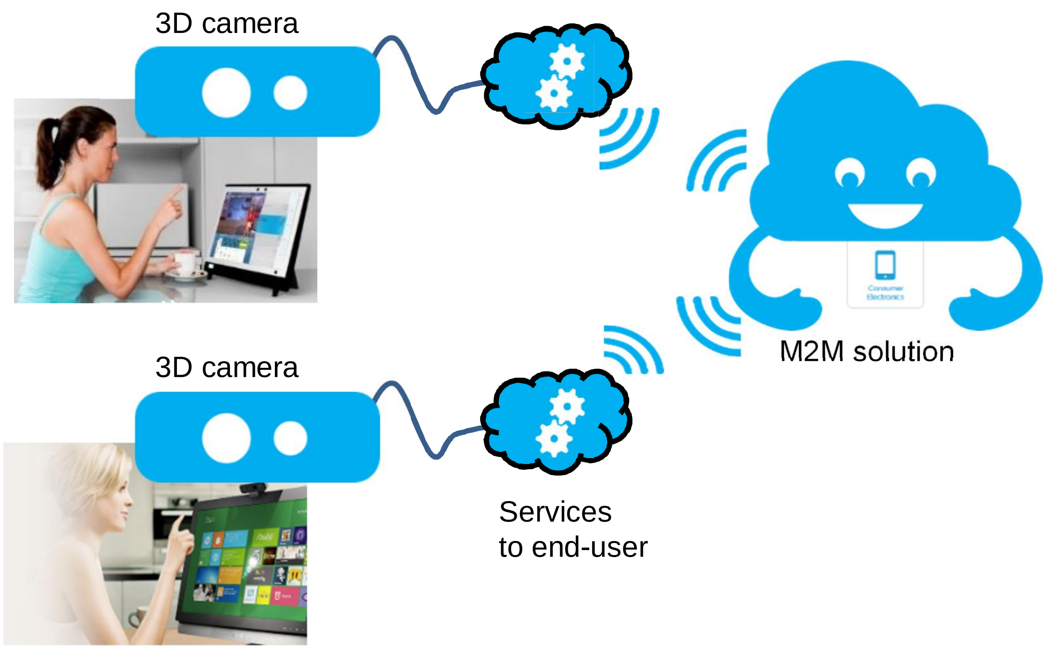


Figure 3.23 Softkinetic 3D sensing game

References

[1] Full Project Proposal Annex, 13011 ITEA 2 M2MGrids, Smart M2M Grids for a Cyber-Physical Information Ecosystem, Juhani Latvakoski, Version 2.0, 6th February 2015

[2] RDF: Resource Description Framework , http://www.w3.org/RDF, 2006.

[3] RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/rdf-schema, 2004.

[4] OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide, 2004.

[5] OWL-S: Semantic Markup for Web Services, http://www.w3.org/Submission/OWL-S/, 2004.

[6] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/, 2008

[7] L. Getoor, et al. (2007). "Probabilistic Relational Models." In L. Getoor and B. Taskar, editors, Introduction to Statistical Relational Learning.

[8] N. Friedman et al., Learning Probabilistic Relational Models, Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99), Volume 2, pp. 1300-1307, August 1999.

[9] Zimmermann, R. et al., "PowerWeb: a tool for evaluating economic and reliability impacts of electric power market designs", IEEE PES Power Systems Conference and Exposition, vol. 3, pp. 1562-1567, 2004.

[10] T. Logenthiran and D. Srinivasan, "Multi-agent system for the operation of an integrated microgrid," Journal of Renewable and Sustainable Energy, vol. 4, p. 013116, 2012.

[11] T. Pinto, et.al., Strategic Bidding in Electricity Markets: An agent-based simulator with game theory for scenario analysis", Integrated Computer-Aided Engineering, IOS Press, vol. 20, no. 4, pág. 335-346, Sept. 2013.

[12] L. Gomes, P. Faria, H. Morais, Z. Vale, C. Ramos, "Distributed, Agent-Based Intelligent System for Demand Response Program Simulation in Smart Grids," IEEE Intelligent Systems, vol. 29, no. 1, pp. 56-65, Jan.-Feb. 2014.

[13] T. Logenthiran, "Multi-Agent System for Control and Management of Distributed Power Systems," Ph.D., National University of Singapore, 2012.

[14] H. Morais, et.al., "Multi-Level Negotiation in Smart Grids for VPP Management of Distributed Resources", IEEE Intelligent Systems, Special Issue "Sustainable Energy and Distributed AI", vol. 27, no. 6, pp.. 8-16, November - December 2012.

[15] Logenthiran, T. & Srinivasan, D., Multi-agent system for the operation of an integrated microgrid. Journal of Renewable and Sustainable Energy, 4, 013116., 2012.

[16] Logenthiran, T., Multi-Agent System for Control and Management of Distributed Power Systems. Ph.D. thesis, National University of Singapore, 2012

[17] Wooldridge, M., An introduction to multiagent systems, Wiley, 2008

[18] Dimeas, A. L. & Hatziargyriou, N. D., Operation of a multiagent system for microgrid control. IEEE Transactions on Power Systems, 20, 1447-1455, 2005

[19] Bo, Z., Xuesong, Z. & Jian, C., Integrated microgrid laboratory system. Power Systems, IEEE Transactions on Sower Systems, 27, 2175-2185, 2012

[20] Fotouhi Ghazvini, M. A., Morais, H. & Vale, Z., Coordination between mid-term maintenance outage decisions and short-term security-constrained scheduling in smart distribution systems. Applied Energy, 96, 281-291, 2012

[21] Duy Thanh, N., Michael, N. & Martin De, G., Walrasian market clearing for demand response exchange. IEEE Transactions on Power Systems, 27, 535-544, 2012.

[22] Harp, S. et al., "Sepia: A Simulator for Electric Power Industry Agents", IEEE Control Systems Magazine, vol. 20, no. 4, pp. 53 – 69, 2000

[23] Zimmermann, R. et al., "PowerWeb: a tool for evaluating economic and reliability impacts of electric power market designs", IEEE PES Power Systems Conference and Exposition, vol. 3, pp. 1562-1567, 2004

[24] Migliavacca, G., "SREMS-electricity market simulator based Game Theory and incorporating network constraints", IEEE Power Tech, Lausanne, Swiss, 2007

[25] Koritarov, V., "Real-World Market Representation with Agents: Modeling the Electricity Market as a Complex Adaptive System with an Agent-Based Approach", IEEE Power & Energy magazine, pp. 39-46, 2004

[26] Li, H. and Tesfatsion, L., "Development of Open Source Software for Power Market Research: The AMES Test Bed", Journal of Energy Markets, vol. 2, no. 2, pp. 111-128, 2009

[27] Cincotti, S., Gallo, G., "Genoa Artificial Power-Exchange," Agents Artificial Intelligence, 2013

[28] Figueiredo, V. et al., "An electric energy consumer characterization framework based on data mining techniques", IEEE Transactions on Power Systems, vol. 20, no. 2, 2005

[29] I. Praça, C. Ramos, Z. Vale, and M. Cordeiro, "MASCEM : A Multiagent System That Simulates Competitive Electricity Markets," Intell. Syst. IEEE, vol. 18, no. 6, pp. 54–60, 2003.

[30] T. Pinto, H. Morais, P. Oliveira, Z. Vale, I. Praça, and C. Ramos, "A new approach for multi-agent coalition formation and management in the scope of electricity markets," Energy, vol. 36, no. 8, pp. 5004–5015, 2011.

[31] B. Teixeira, F. Silva, T. Pinto, I. Praça, G. Santos, and Z. Vale, "Data Mining Approach to support the Generation of Realistic Scenarios for Multi-Agent simulation of Electricity Markets," in Intelligent Agents (IA), 2014 IEEE Symposium on, 2014, pp. 8–15.

[32] T. Pinto, Z. Vale, T. M. Sousa, I. Praça, and H. Morais, "Adaptive Learning in Agents Behaviour : a Framework for Electricity Markets Simulation," Integr. Comput. Aided. Eng., vol. 21, no. 4, pp. 399–415, 2014.

[33] James Momoh Smart Grid: Fundamentals of Design and Analysis April 2012, Wiley-IEEE Press

[34] M.A. Duran and I.E. Grossmann (1986). An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs, Math. Program., 36, 307–339.
[35] R. Fletcher and S. Leyffer (1994). Solving Mixed Integer Programs by Outer Approximation, Math. Program., 66, 327–349.
[36] O.K. Gupta and A. Ravindran (1985). Branch and Bound Experiments in Convex Nonlinear Integer Programming, Manage Sci., 31(12), 1533–1546.
[37] I. Quesada and I.E. Grossmann (1992). An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems, Computers Chem. Eng., 16(10/11), 937–947.
[38] T.Westerlund and F. Petersson (1995). A Cutting Plane Method for Solving Convex MINLP Problems, Computers Chem. Eng., 19, 131–136.
[39] A.M. Geoffrion (1972). A Generalized Benders Decomposition, J. Optim. Theory Appl., 10(4), 237–260.
[40] P. Zhang and A.H. Coonick. "Coordinated Synthesis of PSS Parameters in Multimachine Power Systems Using the Method of Inequalities Applied to Genetic Algorithms". IEEE Transactions on Power Systems 2000, 15, 811 – 816.
[41] Y.L. Abdel - Magid, M.A. Abido, S. Al - Baiyat, and A.H. Mantawy . "Simultaneous Stabilization of Multimachine Power Systems Via Genetic Algorithms", IEEE Transactions on Power Systems 1999, 14, 1428 – 1439.
[42] M. A. Abido. "Robust Design of Multimachine Power System Stabilizers Using Simulated Annealing". IEEE Transactions on Energy Conversions 2000, 15, 297 – 304.
[43] R.A.F. Saleh and H.R Bolton. "Genetic Algorithm- aided Design of a Fuzzy Logic Stabilizer for a Superconducting Generator". IEEE Transactions on Power Systems 2000, 15 1329 – 1335.
[44] M.G. Ippolito, M.L. Di Silvestre, E. Riva Sanseverino, G. Zizzo, G. Graditi, Multi-objective optimized management of electrical energy storage systems in an islanded network with renewable energy sources under different design scenarios, Energy. 64 (2014) 648–662. doi:10.1016/j.energy.2013.11.065.
[45] P. Maghouli, S.H. Hosseini, M. Oloomi Buygi, M. Shahidehpour, A Scenario-Based Multi-Objective Model for Multi-Stage Transmission Expansion Planning, IEEE Trans. Power Syst. 26 (2011) 470–478. doi:10.1109/TPWRS.2010.2048930.
[46] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA II, Ieee Trans. Evol. Comput. 6 (2002) 182–197. doi:Pii S 1089-778x(02)04101-2 Doi 10.1109/4235.996017.
[47] Optimization of Alternative Energy Generator Placement in Electrical Power Distribution Networks using Mixed Integer Nonlinear Programming, (n.d.) 0.
[48] P. Siano, Assessing the Impact of Incentive Regulation for Innovation on RES Integration, (2014) 1–10.
[49] M. Basu, Combined heat and power economic emission dispatch using nondominated sorting genetic algorithm-II, Int. J. Electr. Power Energy Syst. 53 (2013) 135–141. doi:10.1016/j.ijepes.2013.04.014.
[50] X.-D. Yuan, W.-L. Cheng, Multi-objective optimization of household refrigerator with novel heat-storage condensers by Genetic algorithm, Energy Convers. Manag. 84 (2014) 550–561. doi:10.1016/j.enconman.2014.04.086.
[51] Goebel, M. & Gruenwald, L., "A Survey of Data Mining and Knowledge Discovery Software Tools", in SIGKDD Explorations, June, 1999.
[52] Frawley, W.J., Piatetsky-Shapiro, G. & Matheus, C., "Knowledge Discovery in Databases: An Overview", Technical Report, 1995.
[53] Han, Jiawei; Kamber, Micheline; "Data mining: concepts and techniques", The Morgan Kaufmann series in data management systems, San Francisco, 2006.
[54] Dietterich, T., "Bridging the gap between specification and implementation", IEEE Expert , vol. 6, no. 2, pp. 80-82, 1991
[55] Simon, H., "Administrative Decision Making", *IEEE Engineering Management Review,* vol. 1, no. 1, pp. 60-66, 1973
[56] Minsky, M., "A Selected Descriptor-Indexed Bibliography to the Literature on Artificial Intelligence", *IRE Transactions on Human Factors in Electronics,* vol. 2, no. 1, pp. 39-55, 1961
[57] Michalski, R., "Pattern Recognition as Rule-Guided Inductive Inference", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, pp. 349-361, 1980
[58] Winston, P., "Artificial Intelligence", *Addison- Wesley*, pp. 15-228, 1992
[59] Rich, E. and Knight, K., "Artificial Intelligence", McGraw-Hill, 1991
[60] Ginsberg, A., "A unified approach to automatic indexing and information retrieval", *IEEE Expert* , vol. 8, no. 5, pp. 46-56, 1993
[61] Pinto, Tiago et al."Adaptive learning in agents behaviour: A framework for electricity markets simulation". Integrated Computer-Aided Engineering. 2014, 21(4). 399-415. Available: 10.3233/ICA-140477
[62] http://energy.gov/oe/services/technology-development/smart-grid
[63] Jian Linni, Xue Honghong, Guoqing Xu, Zhu Xinyu, Zhao Dongfang, Shao ZY. Regulated charging of plug-in hybrid electric vehicles for minimizing load var- iance in house hold smart microgrid. IEEE Trans Ind Electron 2013; 60 (8):3218–26.
[64] http://powertown.no/wp-content/ uploads/2011/11/SmartGrid_Ueberblick_ohneLegende.jpg
[65] NIST Special Publication 1108, "NIST Framework and Roadmap for Smart Grid Interoperability Standards", 2010
[66] https://www.sgiclearinghouse.org/ConceptualModel
[67] European Commission. Incorporing demand side flexibility, in particular demand response, in electricity markets. 2013. Available: http://ec.europa.eu/energy/gas_electricity/doc/com_2013_public_intervention_swd07_en.pdf
[68] SEDC. Enabling independent aggregation in the European electricity markets - Roles and Responsibilities: Keeping the BRP whole after a demand response event. February 2015. Available: http://sedc-coalition.eu/wp-content/uploads/2015/04/SEDC-Enabling-Independent-Aggregation.pdf
[69] PNNL. Reference Guide for a Transaction-Based Building Controls Framework. April 2014. Available: http://energy.gov/sites/prod/files/2014/06/f16/PNNL-23302_draft.pdf

[70] Meeus, L. et al., “Development of the Internal Electricity Market in Europe”, The Electricity Journal, vol. 18, no. 6, pp. 25-35, 2005
[71] Ilic, M. and Galiana, F., “Power Syst Restructuring: Engineering and Economics”, Kluwer Academic Publishers Group, 1998
[72] Hatziargyrious, N. et al., “Distributed energy sources: technical challenges” IEEE Power Engineering Society Winter Meeting, vol. 2, pp. 1017-1022, 2002
[73] Shahidehpour, M. et al., “Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management”, Wiley-IEEE Press, pp. 233-274, 2002
[74] Operador del Mercado Ibérico de Energia – homepage, http://www.omel.es
[75] Electricity Advisory Committee, “Keeping the Lights On in a New World”, January 2009. Available: http://www.oe.energy.gov/eac.htm
[76] Sheblé, G., “Computational Auction Mechanisms for Restructured Power Industry Operation”, Kluwer Academic Publishers, 1999
[77] Klemperer, P., “Auction Theory: A Guide to the Literature,” Journal of Economic Surveys, vol. 13, no. 3, pp. 227–286, 1999
[78] Praça, I. et al., “MASCEM: A Multi-Agent System that Simulates Competitive Electricity Markets”. IEEE Intelligent Systems, vol. 18, no. 6, pp. 54-60, Special Issue on Agents and Markets, 2003
[79] Santos, G. et al., “Complex Market integration in MASCEM electricity market simulator”, International Conference on the European Energy Market 11 – EEM, 2011
[80] Olsson, M. and Söder, L., “Modeling Real-Time Balancing Power Market Prices Using Combined SARIMA and Markov Processes”, IEEE Transactions on Power Systems, vol. 23, no. 2, 2008
[81] Veen, R. van der, and Vries, L. De, “Balancing market design for a decentralized electricity system: Case of the Netherlands”, First International Conference on Infrastructure Systems and Services: Building Networks for a Brighter Future, INFRA, 2008
[82] Paul Scherrer Institut - PSI, “CO2 -Emissionen aus dem nuklearen Kreislauf”, PSI Fachinfoszu Energiefragen, Villigen, Switzerland, 2007
[83] Matos, F., “Mercados Ambientais e Certificados Verdes”, Workshop Sistemas de Energia - Instituto Superior de Engenharia de Lisboa, 2007
[84] Amelie Gyrard, “A machine-to-machine architecture to merge semantic sensor measurements”, WWW '13 Companion Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 2013. pp 371-376. ISBN: 978-1-4503-2038-2
[85] ETSI, “Machine-to-Machine Communications (M2M); Study on Semantic support for M2M Data”, Technical Report, December 2013.
[86] Laura Daniele, Frank den Hartog, Jasper Roes, “Study on Semantic Assets for Smart Appliances Interoperability”, D-S4: FINAL REPORT: A study prepared for the European Commission DG Communications Networks, Content & Technology, 2013. [Online] Available: http://www.etsi.org/images/files/Events/2015/201502_SMARTAPP/D-S4%20-%20SMART%202013-0077%20-%20Smart%20Appliances%20-%20Final%20Study%20Report_v1.0.pdf
[87] Slobodanka Tomic, Anna Fensel, Tassilo Pellegrini, "SESAME Demonstrator: Ontologies, Services and Policies for Energy Efficiency", In I-SEMANTICS (2010), 2010.
[88] Ye Wang, Yan Hao Huang, Yan Song Li, Yan Yan Wang, "Overview of Ontology-Based Power System Applications", Applied Mechanics and Materials, Vols. 291-294, pp. 2346-2351, Feb. 2013
[89] Mohamed Gaha, Arnaud Zinflou, Christian Langheit, Alexandre Bouffard, Mathieu Viau, Luc Vouligny, “An Ontology-Based Reasoning Approach for Electric Power Utilities”, Proceedings of the 7th International Conference, RR 2013, Mannheim, Germany, July 27-29, 2013.
[90] IETF RFC 2119 “Key words for use in RFCs to Indicate Requirement Levels”, https://tools.ietf.org/html/rfc2119, 1997.

4 State-of-the-Art Analysis on M2M Service Platforms

This chapter discusses the existing technologies, standards, frameworks and systems that are expected to be relevant to the different M2M service aspects to be covered by the layer.

4.1 M2M service level existing solutions and architecture standards

4.1.1 OneM2M Service level architecture

Since early 2015, the OneM2M specification body has released a first mature version of the OneM2M function architecture [58], as well as a number of related, lower-level specifications, e.g. mapping the work to OMA and BBF management models and binding to protocols such as CoAP, HTTP and MQTT [57]. The OneM2M specifications count as ‘umbrella’ reference specifications, that are receive a standard identification number under key M2M standardization organizations such as ETSI and ATIS. It thus comes as no surprise that there is a full alignment with (and on top of) ETSI M2M communication network standards. As we will show, the

OneM2M specification may provide an abstraction foundation for the envisioned functionality of the M2M Service Platform layer of the M2MGrids infrastructure.

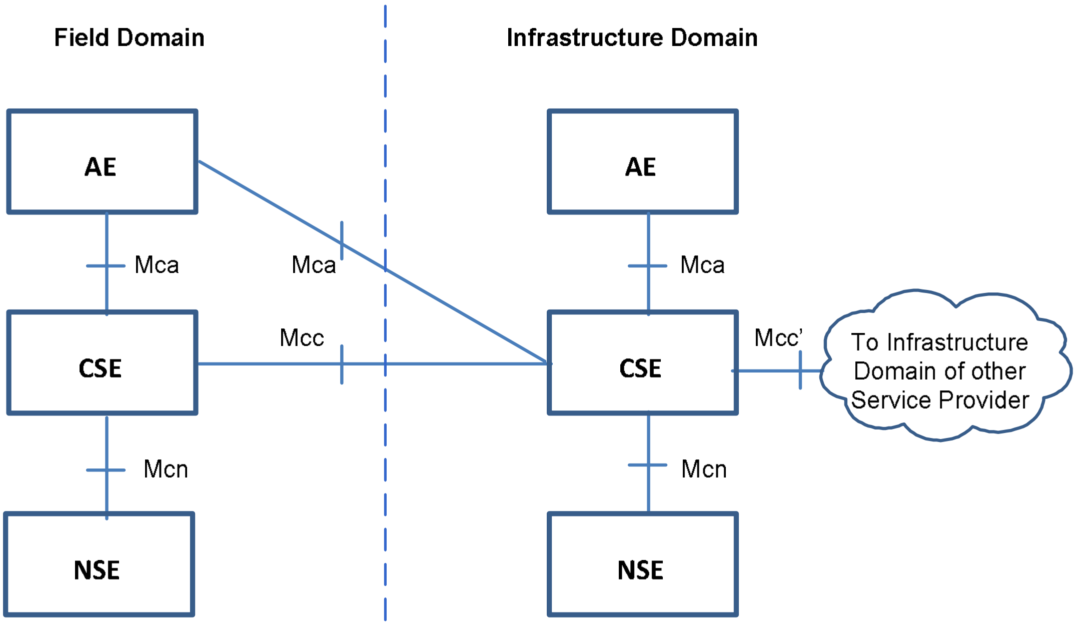


Figure 24. OneM2M basic function instance classes

As illustrated by Figure 24, the OneM2M architecture considers essentially 3 types of possible function instances:

- Application Entities (AEs), which each contain an execution instance of particular application logic and are identified by an AE-ID,
- Common Services Entities (CSEs), which each contain an execution instance of a common infrastructure service and are identified by an CSE-ID, and
- Network Service Entities (NSEs), which each represent instances of abstracted network functions (except the actual transport), such as device management.

These function instances can reside in the *Infrastructure Domain* (i.e. the operated network and cloud equipment and central or edge services), an Infrastructure Domain of another operator, or the *Field Domain*, which e.g. includes Customer Premises Equipment (CPE) or mobile subscriber devices. The specification defines interface reference points between the 3 function classes as indicated in the Figure 24.

As illustrated by Figure 25, the function instances are deployed (configured, upgraded, software-lifecycle-managed) to particular *Nodes*, distinguishing **Infrastructure Nodes (IN)** and various types of **device-dedicated Node types (MN, ASN, ADN)** which correspond typically to physical (constrained or other) M2M Devices or M2M Gateways. However, Nodes are in principle logical entities, meaning that e.g. INs may be distributed over several physical machines, e.g. in a load-balancing, multi-site or elastic cloud distribution.

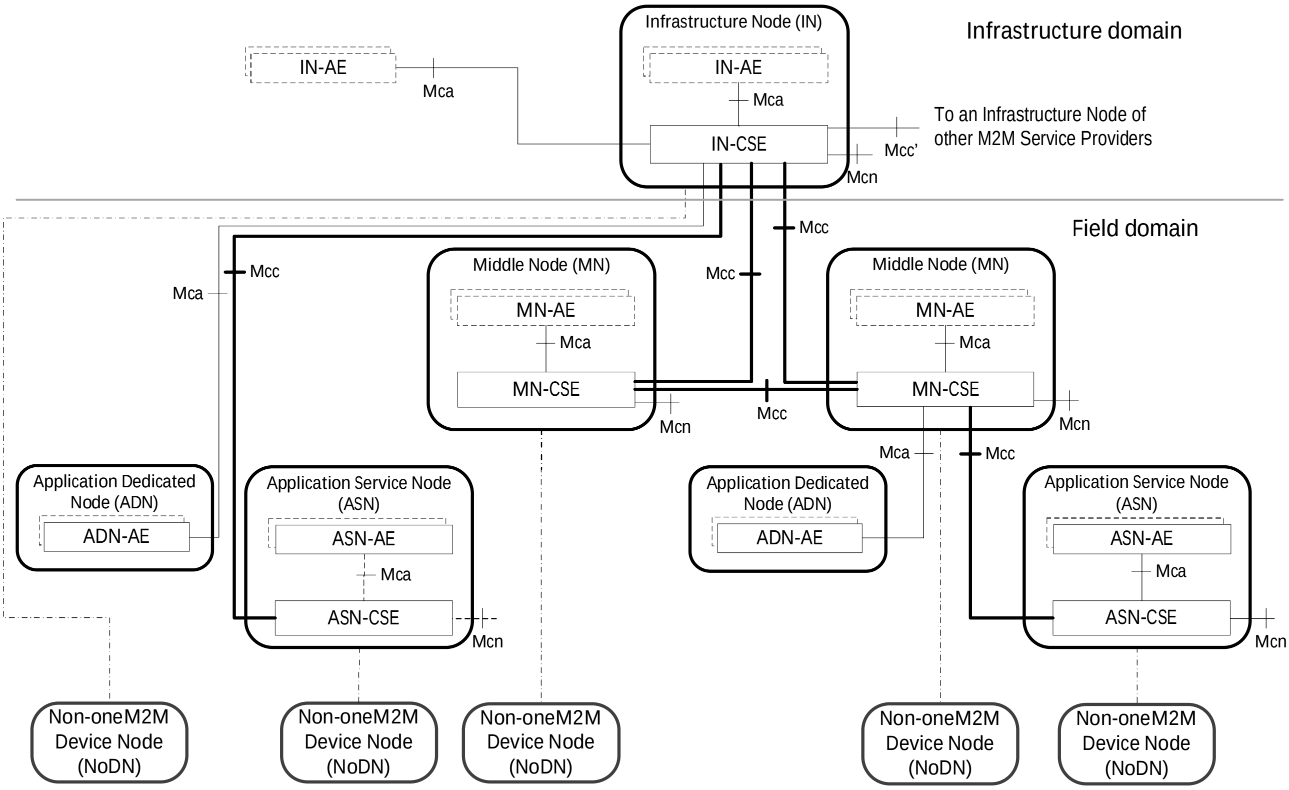


Figure 25. Supported OneM2M node configurations

The OneM2M specification foresees a number of Common Service Functions (CSFs) that can reside in a CSE. Figure 26 gives an overview of these. It is expected that, among the other standards-inspired management functions, e.g. **Registration, Discovery, Location, Group Management and Subscription CSF (REG , DIS, LOC, GMG, SUB)** and especially **Application and Service Layer Management (ASM)**, which is in particular also managing the deployment and lifecycle of AEs, will be relevant base abstractions for the service dynamicity and adaptive placement envisioned to be needed for the most innovative scenarios in the M2MGrids infrastructure.

When the operator / service provider, or 3rd parties via AEs, offer services on the infrastructure towards subscribers, OneM2M foresees an **M2M Service Subscription management**, assigning service and subscription identifiers (M2M-Serv-ID, M2M-Sub-ID).

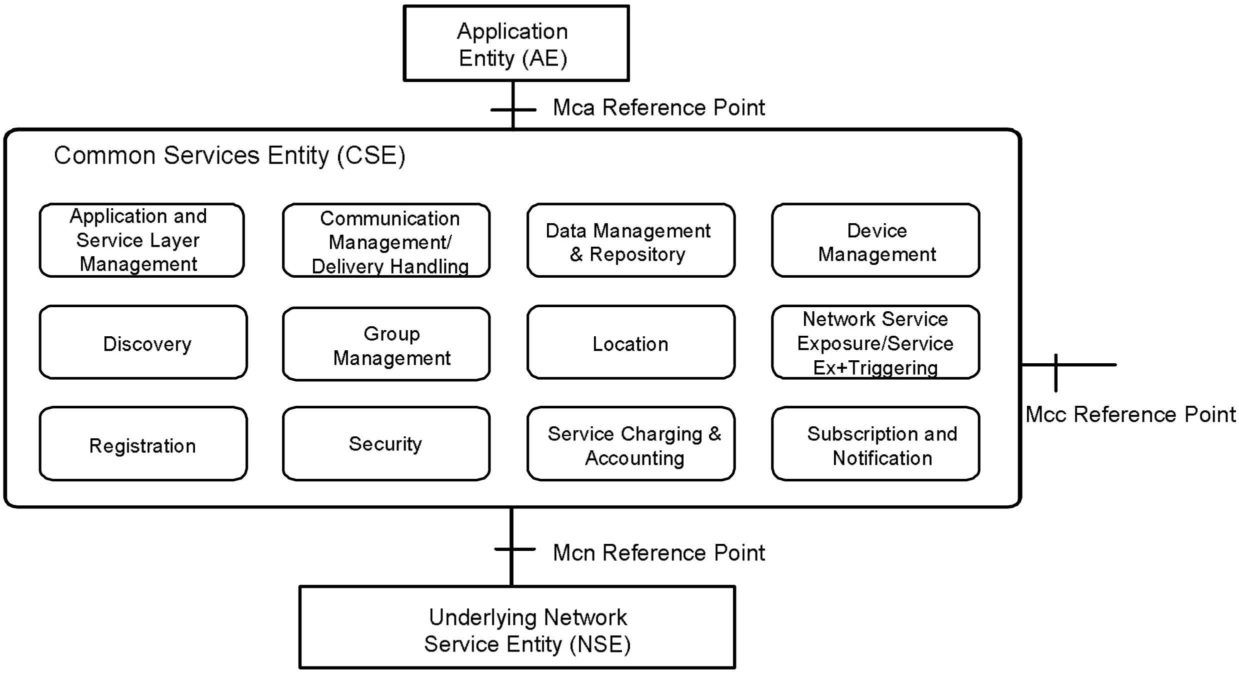


Figure 26. OneM2M Common Service Functions

The **Device Management CSF (DMG)** is abstracting the ETSI M2M device management, TR-069, OMA DM, and LWM2M client-server management structure, considering both M2M Devices and M2M Gateways. The **Communication Management and Delivery Handling CSF (CMDH)** is offering an abstraction for any communication needs among the AE, CSE or NSE entities, be it as control signaling or bearer-type of data streams (but anyhow unaware of the data content), depending on delivery handling parameters and **CMDH policies** expressing the usage conditions and constraints on using the underlying network, and the CSF being aware of the availability of the underlying network. Figure 27 illustrated the example of a 3GPP network, where a CSE at an M2M Device or Gateway eventually communicates with a CSE in the network/cloud infrastructure, after the serving entity has been discovered and the underlying network channels have been established. (As we will discuss in section , the serving CSE in such case is clearly mapable on an ETSI Service Capability server.)

In a OneM2M system, everything (AEs, CSEs, other entities, data, policies, subscriptions, etc.) is represented as **uniquely addressable resources** according to a **type hierarchy** which can be extended as the need arises (i.e. using URIs as an addressing means). Based on this resource type structure, the OneM2M specification describes many procedures generically expected to be needed in an M2M system, on the basis of the general **Create-Retrieve-Update-Delete-Notify (CRUDN)** principle of managing resources, e.g. a *remoteCSE* can be registered with a Registrar CSE, so that the remote resource is known and addressable at the Node of the Registrar.

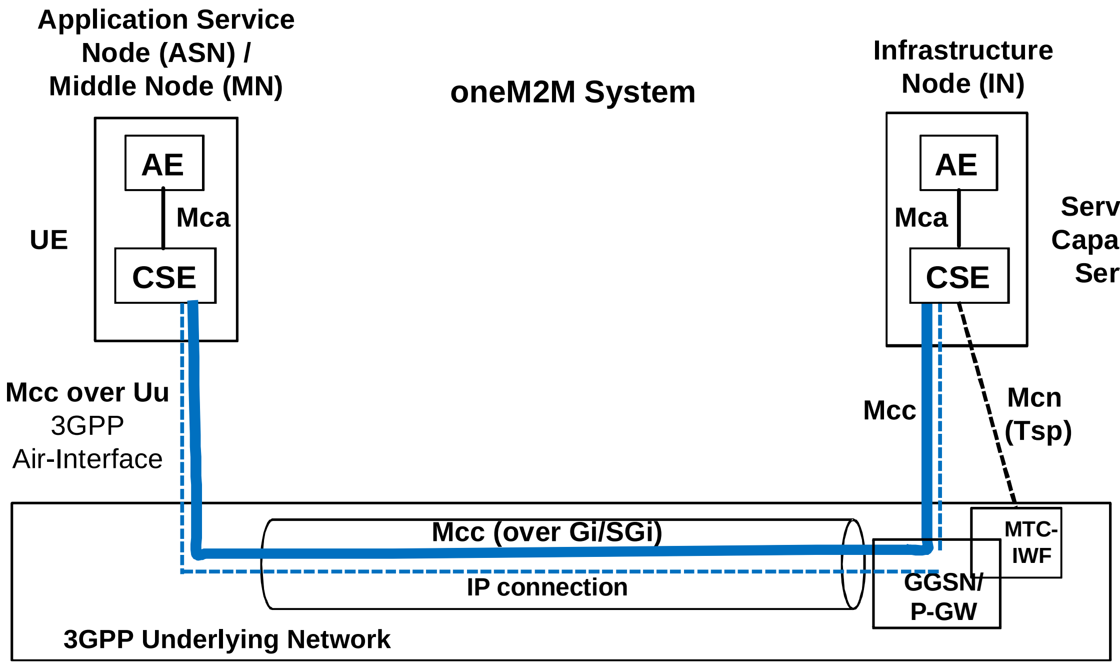


Figure 27. 3GPP connectivity example between Nodes via their respective CSEs

With the OneM2M specifications having been extrapolated from the ETSI M2M specifications, in the next section we zoom in further on the specifics of the ETSI M2M service level itself.

4.1.2 ETSI M2M service level

The core objective of the M2MGrids project is composed of the horizontal structure of service enablers and the vertical structure of the business cases. WP3 aims to provide a horizontal, “A la carte”, dynamic, standard-based (e.g. enhancements to horizontal M2M service capabilities of oneM2M/ETSI) and SOA-based platform, sizeable to devices/gateways/server (cloud). ETSI/M2M standardizes the service layers for the M2M/IoT such that an application developer can develop an application regardless of its operating system (iOS, Android, etc.), and regardless of the network the device receives connectivity services (3G, 4G, Wi-Fi, Bluetooth, fix-line, etc.).

ETSI/M2M defines 3 layers of **SCLs (Service Capabilities Layer)** – Network, Gateway and Device. An application can be developed with APIs on top of each one of the 3 layers. The following figure describes the 3 layer services, taken from ETSI TS 102 690 V1.1.1 (2011-10) – as well as the following figures of this section:

Where:
NSCL: Network Service Capabilities Layer refers to M2M Service Capabilities in the Network Domain.
GSCL: Gateway Service Capabilities Layer refers to M2M Service Capabilities in the M2M Gateway.
DSCL: Device Service Capabilities Layer refers to M2M Service Capabilities in the M2M Device.

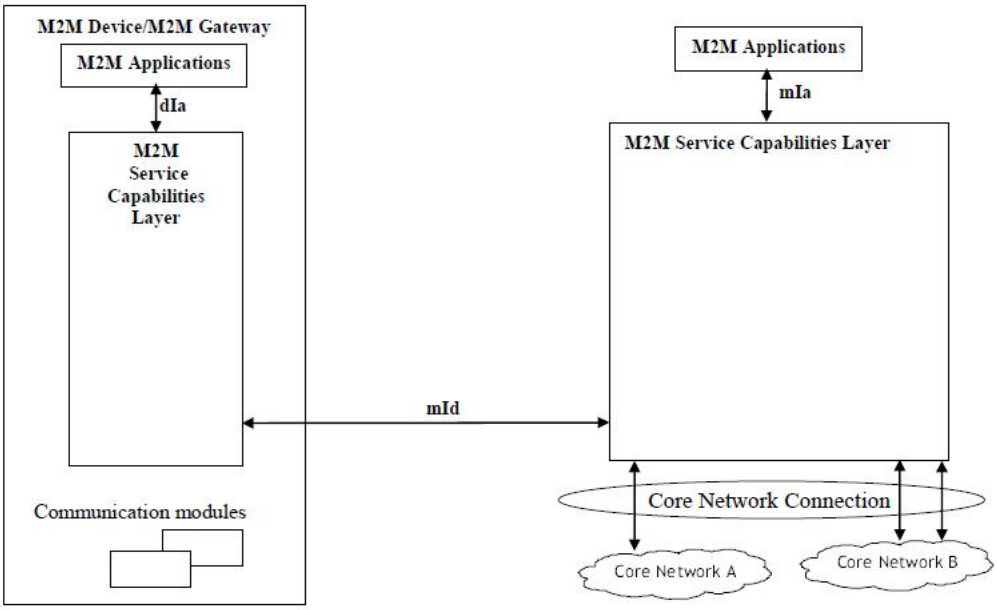


Figure 28. ETSI/M2M Overall High Level Architecture

ETSI/M2M defined 12 service capabilities for the time being:

• Application Enablement (xAE);

• Generic Communication (xGC);

• Reachability, Addressing and Repository (xRAR);

• Communication Selection (xCS);

• Remote Entity Management (xREM);

• SECurity (xSEC);

• History and Data Retention (xHDR);

• Transaction Management (xTM);

• Compensation Broker (xCB);

• Telco Operator Exposure (xTOE);

• Interworking Proxy (xIP);

The xSCL layers manage the complete life cycle of the M2M devices (see commercial exploitations for life cycle management in the next section). That includes the service bootstrap, communication, registration in the N/D/G/SCL. The following figure details the life cycle management:

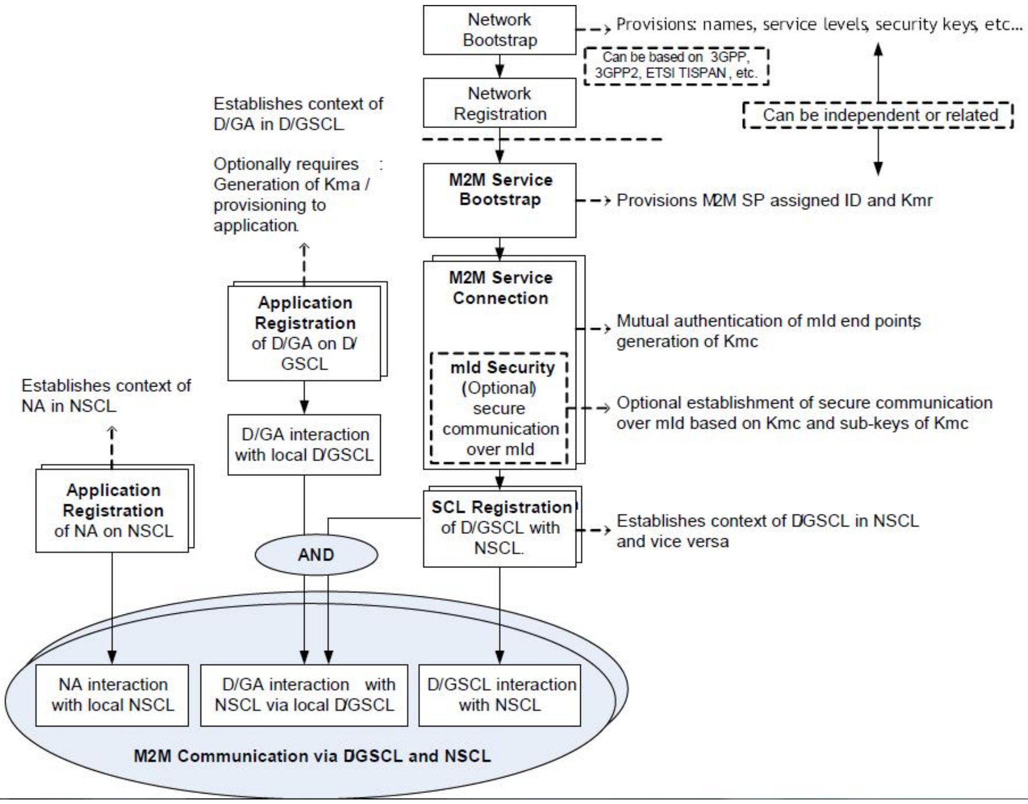


Figure 29. Service establishment scenario in ETSI/M2M

Various interfaces are defined between the SCLs and the communication networks. Starhome, for example, is implementing in the M2MGrid project parts of the MTC-IWF (Machine Type Communication – Inter Working Function), which is an interworking between the mobile network and the external NSCL (Network Service Capabilities Layer).

4.1.3 Existing commercial solutions

The commercial solutions for IoT service management are mainly M2M platforms, known as CDP or ADP/AEP.

CDPs (Connectivity Device Platforms) – which provide global connectivity for M2M verticals (service providers, such as automotive, energy, eHealth), as well as tariff management, life-cycle management for SIM card and device activation, and troubleshooting, The CDP therefore manages all the service relations with the wireless carriers from one hand, and with the service providers ("verticals") from the other hand. These platforms are installed either in the cloud or at the mobile operator premises and are connected to mobile operators via network APIs that the operator provides. There are many MVNOs (Virtual MNOs) for M2M in the market, which are concentrated solely in M2M. The MVNOs sign roaming agreements with real MNOs, and provide global connectivity and tariff management to the verticals, using the multiple roaming relations Such platforms are Jasper (www.jasper.com), Kore (www.kore.com), , Wyless (www.wyless.com). The large mobile operator groups develop such platform on their own, such as Vodafone, Orange, Telefonica, and Deutche Telecom.

Another type of platforms is AEP/ADP, Application Enablement Platforms/Application development Platforms. These platforms enable the development of applications, using APIs and development environment. Such as www.thingworx.com or www.axeda.com. Such platforms may implement the Open APIs defined by OneM2M towards the verticals, and enable them to develop generic applications, that can run on top of any operating system and regardless of the network type. Fraunhofer Fokus is developing OpenMTC, and open source version of some modules of the ETSI M2M (http://www.open-mtc.org/index.html#home)

M2MGrids partner Starhome's plan is to provide APIs from the mobile network towards these platforms, CDPs and ADP/AEPs, APIs that will externalize and expose mobile network parameters, such as location, coverage status.

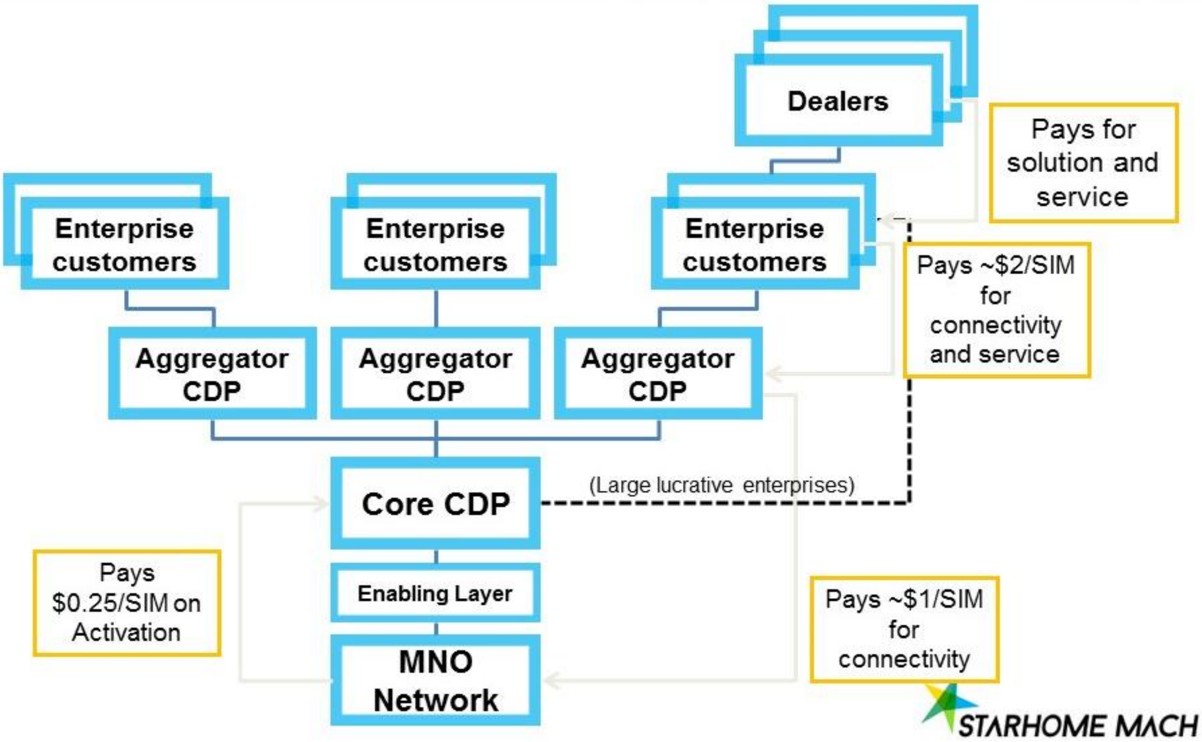


Figure 30. CDPs – Market value chain

Both these CDPs and ADPs/AEPs have coverage in the ETSI M2M; under the SCL (Service Capabilities Layer) components, but the industry adoption of these standards is progressing slowly.

The market value chain for CDPs is described by Figure 30. The enterprise customers are the verticals mentioned before. Aggregator CDPs are mainly MVNO CDPs that can work with many MNO CDPs, where core CDP is the CDP installed at the MNO (Mobile Network) core network.

4.2 Virtualization & container technologies

Containers and virtualization of cloud resources could be assigned to particular service components dynamically by the mechanisms to be explored in the M2M Service Platform layer. Therefore, we discuss virtualization from a general perspective here.

Given a generic machine architecture (for example, the stack shown in Figure 38), virtualization is a technology that introduces software abstractions for various layers in between a machine's physical hardware and the applications running on top of it.

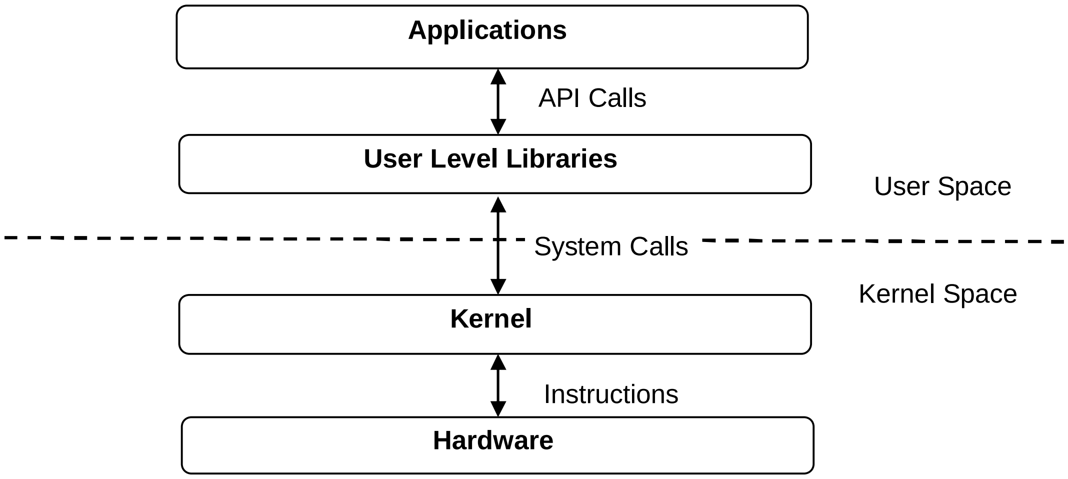


Figure 38. Stacked Machine Architecture [117]

Some of the primary benefits of virtualizations are as follows [118][119]:

- Security: isolation --which is the defining property of virtualization-- makes it easier to impose additional security checks and access control.
- Efficiency: virtualization allows consolidating resources and hence reducing system underutilization; it also improves the performance of system start-up and migration.
- Troubleshooting: through virtualization it is possible to create different test scenarios and run them safely.

Virtualization has been applied across multiple Information Technology (IT) aspects such as systems, storage, networks, security and applications. Today we are witnessing the widespread growth of virtualization technology (especially Virtual Machines and Linux Containers) in testing, training, development and even production environments.

In the following sections we first briefly describe the main types of virtualization and then present some high-level analysis and empirical studies reported in the literature.

4.2.1 Abstraction Levels / Virtualization Types

Based on the stacked architecture in Figure 38, typically the following five types of virtualization can be defined [117][118][119][120]:

Instruction Set Level: Emulation is a virtualization method in which a complete hardware architecture may be created in software. This software is able to replicate the functionality of a designated hardware processor and associated hardware systems. This method provides tremendous flexibility in that the guest OS may not have to be modified to run on what would otherwise be an incompatible architecture. Emulation features tremendous drawbacks in performance penalties as each instruction on the guest system must be translated to be understood by the host system. The main applications of this type of virtualization are debugging and teaching.

Hardware Level: Virtual Machines were introduced on IBM mainframes in the 1970s and then reinvented on x86 by VMware in the late 1990s. Xen and KVM brought VMs to the open source world in the 2000s. The overhead of virtual machines was initially high but has been steadily reduced over the years due to hardware and software optimizations. Hence, virtual machines are used extensively in cloud computing. In particular, the state-of-the-art in Infrastructure as a Service (IaaS) is largely synonymous with virtual machines. Cloud platforms like Amazon EC2 make VMs available to customers and also run services like databases inside VMs. Many Platform as a Service (PaaS) and Software as a Service (SaaS) providers are built on IaaS with all their workloads running inside VMs.

The core component of VMs is Virtual Machine Monitor (VMM) which is a highly privileged piece of software that runs either alongside or under an operating system, it is designed to be “an efficient, isolated duplicate of the real [physical] machine”. A virtual machine monitor is distinct from an emulator. An emulator intercepts all instructions, whereas a virtual machine monitor need only intercept sensitive instructions (those which could interfere with the operation of the VMM itself). All non sensitive instructions should be executed directly on the hardware where possible.

There are two main types of virtual machine monitor: 1) Bare Metal which is installed as the primary boot system on the hardware, and 2) Hosted (as popularized by products such as VMware Workstation) which sits alongside or above a host operating system above the hardware, and may share drivers from the host operating system to handle I/O. This cooperative model results in a VMM system that does not require hardware-specific drivers for VMM I/O operations, and allows the use of virtual machines within an existing environment. As a result entry barriers to VM use are reduced, as the existing OS does not need to be overwritten or migrated to a multiple boot arrangement.

In order to be virtualizable, an architecture must be capable of trapping all sensitive instructions and calling the VMM. The x86 architecture is not fully virtualizable, so various methods have been taken to achieve the widespread virtualization of these systems now in use. The most important of these are paravirtualization, binary translation, and hardware-assisted virtualization. For more in-depth discussion on each of these methods and their relative performances refer to [121].

Operating System Level: Containers are operating environments on top of OS. In other words, an operating system kernel provides for multiple isolated user-space instances. This is not true virtualization, however it does provide the ability for user-space applications (that would be able to run normally on the host OS) to run in isolation from other software. Most implementations of this method can define resource management for the isolated instances. Virtualized SysCall Interface (may be same). May or may not provide all the device abstractions. Easy to manipulate (create, configure, destroy). Processes, File System, Network resource (IP address), Environment variables, System call interface. Applications: Sandboxing, Fine grain access control (root in the container).

Linux containers have a long and winding history. The Linux-VServer project was an initial implementation of “virtual private servers” in 2001 that was never merged into mainstream Linux but was used successfully in PlanetLab. The commercial product Virtuozzo and its open-source version OpenVZ have been used extensively for Web hosting but were also not merged into Linux. Linux finally added native containerization starting in 2007 in the form of kernel namespaces and the LXC userspace tool to manage them.

Within the last two years, Docker [124] has emerged as a standard runtime, image format, and build system for Linux containers.

1. **Resource Level**: is a method in which specific resources of a host system are used by the Guest OS. These may be software based resources such as domain names, certificates, etc. or hardware based such as shared storage space. This form of virtualization is largely used in the HPC (High Performance Computing) community because of its advantages in forming a single logical computer across multiple nodes. Specific examples include Storage Area Networks (SAN) and Virtual Private Networks (VPN).

2. **Application Level**: provides smaller single application virtual machines that allow for emulation of a specific environment on a client system. For example a Java Virtual Machine allows disparate

operating systems such as Windows and Linux to run the same Java program as long as they have the Java VM installed. This form of virtualization is limited in that it only provides single program isolation from the host, but is useful when testing programs out without installing them.

4.2.2 Analysis and Comparison

In this section, we present a summary of some of the high-level analysis and empirical studies regarding virtualization types and software that have been reported in the existing literature.

4.2.2.1 Virtualization Types

Given the aforementioned virtualization types, authors in [117] have made the high-level comparison summary presented in Table 1, where they make the comparison between the virtualization types of main importance.

Table 3. Overall picture of virtualization types [117]

	Instruction Set	Hardware	Operating System	Applications
Performance	*	****	****	**
Flexibility	****	***	**	**
Ease of Implementation	**	*	***	**
Degree of Isolation	***	****	**	***

Note: qualitative scale (meaning poor support, **** meaning good support)*

4.2.2.2 Virtualization Software

Table 2 (from [122]), depicts a performance overview of some of the most widely-used virtualization software that exist today. While looking at this set of results, however, it's important to note that not all of this software are at the same level of abstraction (as explained before, their design goals are different).

Table 4. Existing Virtualization Software Overview Analysis [122]

Product	Type	Performance
Bochs	Emulator	Very Slow
QEMU	Emulator/Native	Close to 100%
VMWare	Native Virtualization	Close to 100%
VirtualBox	Native Virtualization	Close to 100%

Xen	Paravirtualization	100%
Open VZ	OS Level Containers	100%
User Mode Linux	Paravirtualization	Close to 100%

4.2.2.3 Virtual Machines vs. Containers

Currently, Cloud computing makes extensive use of virtual machines (VMs) because they permit workloads to be isolated from one another and for the resource usage to be somewhat controlled. However, the extra levels of abstraction involved in virtualization reduce workload performance, which is passed on to customers as worse price/performance. Newer advances in container-based virtualization (i.e., Linux containers and the Docker ecosystem [123]) have simplified deployment of applications while continuing to permit control of the resources allocated to different applications.

Hence, it's important to know how these two virtualization technologies compare against each other. To this end, the authors of a recent empirical study reported in [123] have explored the performance of traditional virtual machine (i.e., KVM as a representative hypervisor) deployments, and contrast them with the use of Linux containers (i.e., Docker as a container manager). They have used a suite of workloads that stress CPU, memory, storage, and networking resources. In addition to a set of benchmarks that stress different aspects such as compute, memory bandwidth, memory latency, network bandwidth, and I/O bandwidth.

Based on their results, in general, Docker equals or exceeds KVM performance in every case. The results also show that both KVM and Docker introduce negligible overhead for CPU and memory performance (except in extreme cases). For I/O-intensive workloads, both forms of virtualization should be used carefully.

In terms of implications for future cloud architectures, the authors argue that if containers are to be widely adopted they must provide advantages other than steady-state performance. We believe the combination of convenience, faster deployment, elasticity, and performance is likely to become compelling in the near future.

4.3 Complex event processing and other stream processing mechanisms/engines

The M2M Service Platform layer of the M2MGrids architecture would derive/deploy/distribute the executable code for stream processing engines to execute, and potentially also considers dynamic deployment/distribution of the engines themselves. So, in this section, we discuss these as existing mechanisms/platforms/engines for complex event and stream processing.

Applications that require real-time or near-real-time processing of high-volume of data streams are pushing the limits of traditional data processing infrastructures. These event-based applications include market feed processing and electronic trading on financial markets, network and infrastructure monitoring, cloud computing applications, fraud detection, and command and control in military environments. Furthermore, M2M and ubiquity of sensors in the real world can lead to a large field of monitoring and control applications with high-volume and low-latency processing requirements.

M2M systems use various communication technologies for automatically monitoring and controlling variety of systems. In M2M systems, each "machine" emits a continuous stream of data records, which must be analyzed in real-time. Intelligent M2M systems should be able to diagnose their actual states and trigger appropriate actions as soon as critical situations occur.

Complex event processing for M2M differs from existing applications due to the following critical factors [153]:

- **Volume**: Growing volumes of data and how much data needs to be processed within a time window. For M2M volumes start at the border of big data.
- **Variety**: The variety of data range from structured tables, documents, email, metering data, video/audio, stock ticker data, & thousands of proprietary outputs from a multitude of sensors. M2M arguably has one of the widest ranges
- **Velocity**: How fast data is produced and processed to meet demand and of course, the ability to respond once an opportunity or problem is detected.

These factors are out of the comfort zone for typical business solutions for Complex Event Processing (CEP).

4.3.1 Complex Event Processing

Complex Event Processing (CEP) systems are responsible for filtering and combining such notifications to understand what is happening in terms of higher-level events (sometimes also called composite events or situations) to be notified to sinks, which act as event consumers.

CEP systems put great emphasis on the issue that represents the main limitation of most DSMSs: the ability to detect complex patterns of incoming items, involving sequencing and ordering relationships. Indeed, the CEP model relies on the ability to specify composite events through event patterns that match incoming event notifications on the basis of their content and on some ordering relationships on them.

In addition, Carvalho et. al. [154] summarize current initiatives that aim to improve and evolve the existent event processing systems. They discuss the evolution in the event processing field and show the path of development of these tools has changed since integrating the *MapReduce* model into the event processing domain.

4.3.1.1 Overview of Existing CEP Systems

In the following, based on the survey conducted in [133], we briefly describe a set of widely-known CEP systems (both academic/open-source and commercial) as potentially relevant to the M2MGrids work. (For more details as well as a more complete list of CEP systems, see this survey [133].)

Rapide [125] is considered one of the first steps toward the definition of a complex event processing system. It consists of a set of languages and a simulator that allows users to define and execute models of system architectures. Rapide is the first system that enables users to capture the timing and causal relationships between events: in fact, the execution of a simulation produces a causal history, where relationships between events are made explicit.
Rapide models an architecture using a set of components, and the communication between components using events. It embeds a complex event detection system, which is used both to describe how the detection of a certain pattern of events by a component brings to the generation of other events, and to specify properties of interest for the overall architecture. The pattern language of Rapide is quite powerful, in particular it defines conjunctions, disjunctions, negations, sequences, and iterations, with timing constraints. Notice that Rapide does not assume the existence of an absolute time (however, event can be time-stamped with the values of one or more clocks, if available): as a consequence sequences only take into account a causal order between events. During pattern evaluation, Rapide allows rules to access the state of components; we capture this interaction with the presence of the Knowledge Base, in our functional model. The processing model of Rapide captures all possible set of events matching a given pattern, which means that it applies a multiple selection and a zero consumption policies. Notice that complex events can be re-used in the definition of other events.

Since Rapide also uses its pattern language to define general properties of the system, it embeds operators that are not found in other proposals, like logical implications and equivalences between events. To capture these relations, Rapide explicitly stores the history of all events that led to its occurrence. This is made easy by the fact that simulations are executed in a centralized environment.

Cayuga [126] is a general purpose event monitoring system. It is based on a language called CEL (Cayuga Event Language). The structure of the language strongly resembles that of traditional declarative languages for database: it is structured in a SELECT clause that filters input stream, a FROM clause that specifies a streaming expression, and a PUBLISH clause that produces the output. It includes typical SQL operators and constructs, like selection, projection, renaming, union, and aggregates. Despite its structure, we can classify CEL as a detection language: indeed, the streaming expression contained in the FROM clause enables users to specify detection patterns, including sequences (using the NEXT binary operator) as well as iterations (using the FOLD operator). Notice that CEL does not introduce any windowing operator. Complex rules can be defined by combining (nesting) simpler ones. Interestingly, all events are considered as having duration, and much attention is paid in giving a precise semantics for operator composability, so that all defined expressions are left-associated, or can be broken up into a set of left-associated ones. To do so, the semantics of all operators is formally defined using a query algebra; the authors also show how rules can be translated into non deterministic automata for event evaluation.
Different instances of automata work in parallel for the same rule, detecting all possible set of events satisfying the constraints in the rule. This implicitly defines a multiple selection policy; Cayuga uses a zero consumption policy, as events can be used many times for the detection of different complex event occurrences. Since Cayuga is explicitly designed to work on large scale scenarios, the authors put much effort in defining efficient data structures: in particular, they exploit custom heap management, indexing of operator predicates and reuse of shared automata instances. Since detecting automata of different rules are strictly connected with each other, Cayuga does not allow distributed processing.

Sase [127] is a monitoring system designed to perform complex queries over real-time flows of RFID readings. Sase defines a detecting rules language based on patterns; each rule is composed of three parts: event, where and within. The event clause specifies which information items have to be detected and which are the relations between them; relations are expressed using logic operators and sequences. The where clause defines constraints on the inner structure of information items included into the event clause. Finally, the within clause expresses the time of validity for the rule; this way it is possible to define time-based, sliding windows. The language adopted by Sase allows only detection of given patterns of information items; it does not include any notion of aggregation.
Sase compiles rules into a query plan having a fixed structure: it is composed of six blocks, which sequentially process incoming information elements realizing a sort of pipeline: the first two blocks detect information matching the logic pattern of the event clause by using finite state automata. Successive blocks check selections constraints, windows, negations, and build the desired output. Since all these operators explicitly specify the set of events to be selected, it is not possible to capture unbounded sequences of information items (Seq is bounded).

Sase+ [128] is an expressive event processing language from the authors of Sase. Sase+ extends the expressiveness of Sase, by including iterations and aggregates as possible parts of detecting patterns. Non deterministic automata are used for pattern detection, as well as for providing a precise semantics of the language. An interesting aspect of Sase+ is the possibility for users to customize selection policies using strategies. Selection strategies define which events are valid for an automaton transition: only the next one (if satisfying the rule's constraints), or the next satisfying one, or all satisfying ones that satisfy. Consumption of events, instead, is not taken into account.

Esper [129] is considered the leading open-source CEP provider. Esper defines a rich declarative language for rule specification, called EPL (Event Processing Language). EPL includes all the operators of SQL, adding a ad-hoc construct for window definition and interaction, and for output generation. The Esper language and processing algorithm are integrated into the Java and .Net (NEsper) as libraries. Users can install new rules from their programs and then receive output data either in a push-based mode (using listeners) or in a pull-based one (using iterators).
EPL embeds two different ways to express patterns: the first one exploits so called EPL Patterns, that are defined as nested constraints including conjunctions, disjunctions, negations, sequences, and iterations. The

second one uses at regular expressions. The two syntaxes offer the same expressiveness. An interesting aspect of Esper patterns is the possibility to explicitly program event selection policies, exploiting the every and every-distinct modifiers. Esper supports both centralized and clustered deployments; in fact, using the EsperHa (Esper High Availability) mechanisms it is also possible to take advantage of the processing power of different, well-connected, nodes, to increase the system's availability and to share the system's load according to customizable QoS policies.

Coral8 CEP Engine [130] uses a processing paradigm in which flows of information are transformed through one or more processing steps, using a declarative, SQL-like language, called CCL (Continuous Computation Language). CCL includes all SQL statements; in addition it offers clauses for creating time-based or count-based windows, for reading and writing data in a defined window, and for delivering items as part of the output stream.

CCL also provides simple constructs for pattern matching, including conjunctions, disjunctions, negations, and sequences; instead, it does not offer support for repetitions. Like in Aleri, the Coral8 engine does not rely upon the existence of an absolute time model: users may specify, stream by stream, the processing order (e.g., increasing timestamp value, if a timestamp is provided, or arrival order). The results of processing can be obtained in two ways: by subscribing to an output flow (push), or by reading the content of a public window (pull).

Together with its CEP Engine, Coral8 also offers a graphical environment for developing and deploying, called Coral8 Studio (now part of SAP's Event Stream Processor). This tool can be used to specify data sources and to graphically combine different processing rules, by explicitly drawing a plan in which the output of a component becomes the input for others. Using this tool, all CCL rules become the primitive building blocks for the definition of more complex rules, specified using a graphical, plan-based, language.

The Coral8 CEP engine may execute in a centralized or clustered environment. The support for clustered deployment is used to increase the availability of the system, even in presence of failures.

Oracle CEP [131] is a system that provides real time information flow processing. Oracle CEP uses CQL as its rule definition language, but, similarly to Coral8 and Tibco, it adds a set of relation-to-relation operators designed to provide pattern detection, including conjunctions, disjunctions, and sequences. An interesting aspect of this pattern language is the possibility for users to program the selection and consumption policies of rules.

Like in Coral8 and Tibco, a visual, plan-based language is also available inside a development environment based on Eclipse. This tool enables users to connect simple rules into a complex execution plan. Oracle CEP is integrated with existing Oracle solutions, which includes technology for distributed processing in clustered environment, as well as tools for analysis of historical data.

Tibco Business Events [132] is another widespread complex event processing system. It is mainly designed to support enterprise processes and to integrate existing Tibco products for business process management. To do so, Tibco Business Events exploits the pattern-based language of Rapide, which enables the specification of complex patterns to detect occurrences of events and the definition of actions to automatically react after detection. Interestingly, the architecture of Tibco Business Events is capable of decentralized processing, by defining a network of event processing agents: each agent is responsible for processing and filtering events coming from its own local scope. This allows, for example, correlating multiple RFID readings before their value is sent to other agents.

4.3.1.2 Comparison

The Tables below (adopted from [133]) provide a few high-level comparison matrixes for the CEP systems described above.

Table 5. Functional and Processing Models [133]

	Functional Model	Processing Model

Name	Clock	K. Base	Seq	Recursion	Select. Policy	Consum. Policy	Load Shedding
Rapide	Absent	Present	Unbounded	Yes	Multiple	Zero	No
Cayuga	Absent	Absent	Unbounded	Yes	Multiple	Zero	No
Sase	Absent	Absent	Bounded	No	Multiple	Zero	No
Sase+	Absent	Absent	Unbounded	No	Program.	Zero	No
Coral8 CEP	Present	Present	Unbounded	No	Program.	Program.	Yes
Oracle CEP	Absent	Present	Unbounded	No	Program.	Program.	?
Esper	Present	Present	Unbounded	No	Program	Program	No
Tibco BE	Absent	Absent	Unbounded	Yes	Multiple	Zero	?

Table 6. Deployment and Interaction Models [133]

	Deployment	Interaction Model		
Name	**Deployment Type**	**Observation**	**Notification**	**Forwarding**
Rapide	Centralized	Push	Push	n.a.
Cayuga	Centralized	Push	Push	n.a.
Sase	Centralized	Push	Push	n.a.
Sase+	Centralized	Push	Push	n.a.
Coral8 CEP	Clustered	Push	Push/Pull	Push
Oracle CEP	Clustered	Push	Push/Pull	Push
Esper	Clustered	Push	Push/Pull	Push
Tibco BE	Networked	Push	Push	Push

Table 7. Data, Time, and Rule Models [133]

	Data Model			Time Model	Rule Model
Name	**Nature of Items**	**Format**	**Nature of Flows**	**Time**	**Rule Type**
Rapide	Events	Records	Heterogeneous	Causal	Detecting
Cayuga	Events	Tuples	Homogeneous	Interval	Detecting
Sase	Events	Records	Heterogeneous	Absolute	Detecting
Sase+	Events	Records	Heterogeneous	Absolute	Detecting
Coral8 CEP	Data	Tuples	Homogeneous	Stream-only	Transforming
Oracle CEP	Data	Tuples	Homogeneous	Stream-only	Transforming
Esper	Data	Tuples	Homogeneous	Stream-only	Transforming
Tibco BE	Events	Records	Homogeneous	Interval	Detecting

Table 8. Language Model [133]

	Single Item			Logic						Window						Flow Management											
Name	**Selection**	**Projection**	**Renaming**	**Conjunction**	**Disjunction**	**Repetition**	**Negation**	**Sequence**	**Iteration**	**Fixed**	**Landmark**	**Sliding**	**Pane**	**Tumble**	**User Defined**	**Join**	**Union**	**Except**	**Intersect**	**Remove Dup**	**Duplicate**	**Group By**	**Order By**	**Parameterizatio**	**Flow Creation**	**Detection Aggr**	**Production Aggr**
Rapide	X			X	X	X	X	X	X	X		X				X	X							X		X	
Cayuga	X	X	X	X	X			X	X							X								X		X	X
Sase	X			X	X		X	X				X												X			
Sase+	X			X	X		X	X	X			X												X		X	X
Coral8 CEP	X	X	X	X	X		X	X		X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
Oracle CEP	X	X	X	X		X		X	X			X				X	X	X	X	X	X	X	X	X	X	X	X
Esper	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X					X	X	X	X	X	X
Tibco BE	X			X	X	X	X	X	X	X		X				X	X							X		X	

4.3.2 Improvements and Evolution of Complex Event Processing

Beyond the detailed summary on CEP systems of the previous section, Carvalho at. al. [154] further studied current initiatives that aim to improve and evolve the existent event processing systems, identifying that the path of development of these tools has changed since integrating the MapReduce model into the event processing domain is being considered.

So, their survey studies not only the CEP capability of these systems, but also aspects from perspectives such as data stream management systems (DSMS) – which are like database managements systems (DBMS) but focus on continuous data streams – and whether these systems have provisions for distributed processing systems, for very high data throughput. This being important to the adaptation of CEP in the M2M domain, Table 7 shows systems characterized by these aspects, in comparison.

(And, in the next section, we will fully focus on the distributed stream processing platforms.)

Table 9. Distributed and DSMS-oriented CEP systems comparison

Name	Release Year	Description	DBMS	DSMS	CEP	Distribute
Google Photon	2013	Distributed stream processing system		x		x
Walmart Muppet	2012	Distributed event processing system		x		x
StreamDrill	2012	Stream processing system			x	
SAP HANA	2011	In-memory database	x	x		x
Apache Storm	2011	Distributed stream processing system		x		x
Apache YARN	2011	Distributed general-purpose processing system	x	x	x	x
Apache Flume	2011	Distributed stream processing system		x		x
Apache Kafka	2011	Distributed stream processing system		x		x
Apache S4	2011	Distributed event processing system		x		x
Apache Chukwa	2010	Distributed stream processing system		x		x
HStreaming	2010	Distributed stream processing system		x		x
AMPLab Spark	2010	Distributed general-purpose processing system	x	x		x
VoltDB	2010	In-memory distributed database	x	x		x
Esper	2006	Complex Event Processing			x	
StreamBase CEP	2003	Distributed Complex Event Processing			x	x
SQLstream	2003	Distributed stream processing system		x		x

Google Photon is a distributed system for joining multiple continuously flowing streams of data in real-time with high scalability and low latency, where the streams may be unordered or delayed.

Apache YARN is a framework that facilitates writing arbitrary distributed processing frameworks and applications. YARN provides the daemons and APIs necessary to develop generic distributed applications of any kind, handles and schedules resource requests (such as memory and CPU) from such applications, and supervises their execution. YARN's execution model is more generic than the earlier MapReduce implementations. (See also section 4.3.3 for further discussion, under Apache Samza and Flink.)

StreamBase is a complex event processing platform for the rapid building of systems that analyze and act on real-time streaming data for instantaneous decision making and combines a rapid application development environment, an ultra low-latency high-throughput event server, and connectivity to real-time and historical data.

Apache Chukwa is a scalable distributed system for monitoring and analysis of log-based data. Log processing was one of the original purposes of MapReduce.

Apache Flume is a distributed system for collecting log data from many sources, aggregating it, and writing it to the Hadoop file system. highly available, while providing a simple, flexible, and intuitive programming model based on streaming data flows.

SAP HANA is a general purpose and ANSI standards compliant in-memory database. Because of its design, it allows transactional and OLAP reporting in a single system.

Apache Storm is a scalable, fault-tolerant distributed stream processing platform. Similar to tools like Apache S4, it provides a distributed stream processing system, based on a particular architecture, formed by elements called bolts and spolts, that together form a processing in topology design. (See also next section 4.3.3 for further discussion.)

Apache Kafka is a distributed publish-subscribe messaging system. It is designed to provide high throughput persistent messaging that is scalable and allows for parallel data loads into Hadoop. (See also section 4.3.3 for further discussion, under Apache Samza and Flink.)

AMPLab Spark is a general-purpose open source in memory cluster computing system. Its main difference is that its system is based on an in-memory model, providing better throughputs than Hadoop-based systems. (See also next section 4.3.3 and section 4.5.2 for further discussion, as Apache Spark Streaming.)

VoltDB is an ACID-compliant in-memory database. It represents a new type of databases that focus on maintain the guarantees that traditional relational databases offer, but also provides a scalable and fault-tolerant system

Apache S4 is a general-purpose distributed stream processing platform. It focuses on providing a middleware for the development of applications that process continuous unbounded streams of data, and a way similar to tools like Apache Storm. (See also next section 4.3.3 for further discussion.)

Walmart Muppet is a stream processing tool, similar to Apache S4 and Apache Storm. It focuses on distributed parallel processing of streams of continuously flowing data, proposing a model called MapUpdate, which is a MapReduce based model for stream processing.

HStreaming is a general-purpose distributed data analytics platform for streaming data. It runs over Hadoop and aims to provide real-time processing over streams in a more performant manner than Hadoop's common MapReduce

SQLstream is an in-memory data-analytics platform. It operates in a similar way as SAP Hana and AMPLab Spark, but focusing mainly on stream processing. As VoltDB, the system provides queries in standard SQL language, and executes the queries before the data arrives in data warehouse systems.

Streamdrill is a complex event processing system. It is oriented for real-time data analytics, focusing mainly on the top- k problem. The problem consists of continuously updating a user generated query with streaming tuples, generating a trending list with k updated entries.

4.3.3 Distributed Stream Processing Platforms

Distributed stream processing platforms solve the problem of how to process data streams that are too large to be processed by any single machine. The solution is to spread the processing over a cluster of interconnected machines. This, however, exposes the programmer to all the difficulties of distributed computing, including dealing with load imbalance among machines, dealing with machine failures, unreliable message transmission between machines, etc. The goal of distributed stream processing platforms is to hide this complexity from the programmer, much in the spirit of how Google's MapReduce [59] framework hides the complexities of spreading a batch processing job across a cluster of machines. One of the first such distributed stream processing platforms was Apache S4 [60], originally developed by Yahoo!, but now largely superseded by more recent stream processing frameworks discussed below.

Google Millwheel [61] is a general-purpose stream processing framework for low-latency data processing applications built by Google. It is part of what powers Google's Cloud Dataflow "stream processing as a service" platform, and is used internally at Google for tasks such as anomaly detection. Stream processing queries are conceived as a graph of "operators" where the operators are written in a general-purpose programming language like C++ or Java. These operators are deployed on a cluster of commodity machines. The framework provides strong fault-tolerance guarantees: any part of the graph may fail at any time without disrupting the stream processing pipeline. This is achieved by regularly checkpointing operator state, and by providing exactly-once delivery guarantees of the "records" sent between stream operators. Millwheel enables parallel processing of large data streams by splitting a single large stream into many smaller streams. It does this by partitioning the stream based on keys from the stream's records. Processing stages for records with the same key are sequentialized, but records with different keys can be processed in parallel.

Apache Storm [62] is an open source distributed stream processing framework for the Java Virtual Machine, originally developed by Twitter. Stream processing pipelines are written as Java components called "bolts", wired together in a directed acyclic graph. A Storm stream is a sequence of data "tuples". Each bolt processes input tuples and emits output tuples. Special input components called "spouts" inject external data streams as tuples into the system. Storm manages the scheduling and deployment of bolts across a cluster. By default, Storm provides at-least-once delivery guarantees of tuples among bolts. A higher-level framework called Trident provides transactional exactly-once guarantees, ensuring that each tuple is processed exactly once.

Apache Samza [63] is an open source distributed stream processing framework originally developed by LinkedIn. It uses Apache Kafka [64], a fault-tolerant publish/subscribe message broker, to connect distributed stream operators. Apache Kafka was designed from the get-go for high-throughput workloads and a clustered design. Similar to Google's Millwheel, large streams are partitioned into smaller streams based on keys. Kafka persists all routed messages to a log file on disk, achieving functionality similar to a database commit log. Samza provides support for stream processing on top of Kafka, by deploying so-called "containers" across a cluster of machines. It integrates with multiple possible data center schedulers, but uses YARN [65] by default (YARN is also used to schedule Hadoop MapReduce jobs). Each Samza container can run multiple "stream tasks". These tasks can be stateful and have access to local storage. Updates to a task's state are streamed out to Kafka as a changelog, where it is persisted. If the container dies, its replacement can stream the changelog from a previous checkpoint to restore the local storage of the task. This allows local state to scale horizontally with processing. In addition, each task can just read and write to and from local state, which is an order of magnitude faster than having to access a centralized remote store.

Amazon Kinesis [66] is the core technology that powers distributed stream processing on Amazon's Elastic Compute Cloud (EC2). While it is a closed-source product, its architecture appears to be similar to Kafka. **Amazon Lambda** [67] is a stream processing service that lets developers easily process Kinesis streams using stream operator functions written in high-level languages such as JavaScript (node.js) or Java. The function is called each time a preconfigured event occurs (e.g. an entry was added to a Kinesis stream). Unlike in the case of Millwheel or Samza, a Lambda function must be stateless, that is, all persistent data must be stored in an external database. Amazon Lambda automatically replicates the user-provided function as the number of input events increases. Users are charged per function call invocation and per time quantum that the function executes, leading to a very fine-grained service cost model.

Apache Spark Streaming [68] is an open source distributed stream processing framework designed to process data streams in real-time, such as for collecting site activity statistics (clickstream data), spam detection, cluster health monitoring and network intrusion detection. Spark Streaming is built as an extension to the Apache Spark framework. Like Hadoop, Spark is designed for batch processing, but unlike Hadoop, it aims to keep the data in-memory as much as possible (rather than reading from and writing to external storage systems such as HDFS). This makes it ideal for running iterative algorithms that reuse the same data set over and over again. The basic unit of processing in Spark is an RDD (resilient distributed dataset), which is a collection of elements partitioned across a cluster of machines. An RDD can then be transformed or processed in parallel across the cluster. Spark Streaming adapts Spark's batch processing to streaming queries, by breaking up an incoming data stream into little batches that are individually fed into the Spark engine. Such streams are called DStreams (discretized streams) and are essentially a sequence of RDDs. Spark Streaming guarantees that all DStream transformations have exactly-once semantics. Incoming DStream data is replicated to survive machine failures.

Apache Flink [69] is a large-scale data processing engine. It is very similar to, but less mature than Apache Spark. Like Spark, Flink offers support for both batch and stream processing and aims to keep as much of the data in-memory so that cyclic or iterative computations can be performed faster on the same data. Like Spark, it has both a Scala and a Java API. Flink uses Hadoop's YARN as the scheduler and offers I/O connectors to components that can feed in a large volumen of data such as Kafka, RabbitMQ, Flume, etc. Flink's main differentiator, compared to Spark, is that it offers actual low-latency stream processing. In Flink, incoming data elements are immediately pipelined through the stream computation as they arrive. By contrast, Spark Streaming collects arriving data elements into small batches which it then processes as a regular batch job.

Microsoft Azure Service Fabric [70] is essentially Microsoft's response to Amazon Lambda. It is targeted at so-called microservices architectures, where software is written as many small independent components interacting via the exchange of messages. Key differentiators of Azure Service Fabric with respect to Amazon Lambda is that microservices can be stateful, and that the platform can be hosted as a service or deployed on premises. Azure Service Fabric is based on an open source programming framework for Microsoft's .NET platform called Orleans [71]. In Orleans, services are defined as Actors [72], independent units of computation interacting via asynchronous message passing. A key contribution of Orleans is its so-called "virtual actor" abstraction: an actor always exists, virtually, and will be created on-the-fly if it's not yet already running when a request for that actor enters the system. Like Amazon Lambda, Orleans automatically replicates functions (actors) to keep up with increasing load.

Vert.x [73] is an open source JVM-based event-driven application platform. One of its main selling points is its support for "polyglot" programming, allowing the easy set up of data processing pipelines where the stream operators themselves can be written in a variety of different programming languages (as long as the runtime of the language is JVM-based). Distributed components are called "Verticles" and follow a strict single-threaded event loop architecture. Verticles communicate via a distributed event bus (also known as a message broker) supporting publish/subscribe or point-to-point interaction between components. Verticles can be packaged into modules so that they can be easily reused in other projects. Vert.x supports external endpoints via standard protocols such as TCP, HTTP, Websockets or SockJS.

4.3.4 Business and Enterprise Platforms that offer Integrated CEP for IoT

Today, specifically in the enterprise realm, there exist cost effective ways for business systems and applications to talk to each other. A good example of this kind of solution is the recently evolved IBM Integration Bus. This is a solution that can allow disparate IT systems to share and disseminate data where it can be best utilized within the organization. This is built on a software architecture model commonly known as an Enterprise Service Bus (ESB). However some of these solutions (based on enterprise middleware such as ESBs) are not designed to deal with the challenges facing IoT platforms.

Mnubo is an SaaS solution providing a comprehensive Big Data platform catering to the Internet of Things via three solutions: mnubo smartobjects cloud, mnulabs and mnubo smartobjects analytics. Mnubo facilitates business logic modeling and Big Data analytics, speeding up your time to market by allowing you to focus on what you do best — building intelligent objects.

Oracle IoT is the vendor's Java Embedded solutions aim to reign in the massive amounts of data required for and created as a result of the Internet of Things by facilitating seamless communications between all elements of the IoT architecture.

Swarm is an IoT development platform that facilitates adding new services to products easily. Swarm Dashboards serve as central, device-specific home pages, offering real-time, visual access to device features. Dashboards add value to connected products, enabling event notifications and alerts, historical data, analytics and reporting and other features for turning machine data into actionable insights.

Axeda is a comprehensive cloud-based platform for managing connected products and machines and implementing IoT and M2M applications. The platform is used to transform machine data into valuable insights, build and run applications and integrate machine data with other applications and systems to optimize business processes.

SAP's IoT solutions facilitate connectivity and multi-directional communication to enable users to interact with their devices in new ways. Transforming operations in field service and remote asset management, providing supply chain visibility and predicting and remedying logistics bottlenecks are just some of the challenges solved by SAP's remote maintenance and service, connected logistics and connected retail solutions for the IoT.

ThingWorx aims at rapid, streamlined creation of end-to-end smart applications for agriculture, cities, grid, water, building and telematics. Traditional industries are transformed and equipped with modern-day connectivity and smarter solutions through connected devices that provide comprehensive data collection and analysis for data-driven decision-making.

Arrayent is an IoT platform for connected objects, enabling major brands like Whirlpool, Maytag and First Alert to bring smart, connected devices to consumers.

4.3.5 Message brokers for M2M and IoT

M2M systems use various communication technologies for automatically monitoring and controlling machines. In M2M systems, each sensor may emit a continuous stream of data records, which must be analyzed in real-time. Intelligent M2M systems should be able to diagnose their actual states and to trigger appropriate actions as soon as critical situations occur. Speed is important, and so lightweight messaging middleware and protocols are critical to event processing and high performance streaming. Here a few important examples of message brokers complementing the above stream processing systems.

IBM Message Sight is a specialty appliance and secure gateway to the IoT and high-performance mobile messaging that can scale up to millions of messages a second.

MQTT (MQ Telemetry Transport) is an M2M and IoT connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport, for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. This makes it suitable for M2M messaging such as with low power sensors or mobile devices such as phones, embedded computers or microcontrollers, like Arduino and many other small sensing and computing boards that have emerged in recent years. MQTT has therefore become a popular choice for M2M/IoT message brokering.

Mosquitto is an open source message broker that implements the MQTT protocol.

4.4 Intelligent service resource allocation procedures

Another key responsibility of the M2M Service platform layer is the intelligent (and dynamic) allocation of network and processing resources for service components / stream processing operators. Allocation of network resources or network flow problem has wide range of applicability such as the transportation network problem where the flow of the vehicles through the railway or road network is considered; the wireless sensor networks where the flow of measured data through a wireless channel is considered; and the Internet, where the information packets flow through the wired network. Allocation of computing resources [2, 3, 4], allocation of network resources [2, 5] in computer or communication networks, and machine allocation in factories [6] are other examples of network resource allocation. In most real-world cases resources are limited. Resources are any means that enable operations of an engineered system or process, e.g., energy, information, materials, capacities of plants, bandwidth, and machines [1].

High quality computing, storage resources and high-speed networks are being installed worldwide. Network bandwidths are no longer the bottleneck in many high performance computational workflows due to their evolution. Nowadays, many data intensive applications data can be retrieved over high speed network links from remote machines at higher data rates than from local disks. Although these resources are rare, and mostly not available locally, parallel and high performance storage systems can pump data at high rates for computational and analysis tasks. Large network bandwidths reduce the difference between a local and a remote storage system, enabling access to remote storage systems through network links [23].

Performance models give reasonable predictions on how resources perform with certain workloads under certain conditions, being a requirement for solving the problem of intelligent resource brokering. Both the local application component performance, and the performance of the network links connecting the nodes where the application is running must be accurately modeled by the distribution application performance model [23].

Internet is the most popular and widely used distributed system that includes a large number of computers and users connected together through a network. Recently, the power grids also fit into the large-scale networked system category. The Smart Grid (SG) is proposed to address the drawbacks of the current power grid. It includes improved computing, communication and incorporation of sensors in different parts of the network. The future SG, with decision making capability and increased communications, that will be more distributed, dynamic and unpredictable in nature because of the incorporation of renewable and distributed energy resources, may be considered a good example of large-scale networked systems. Cloud computing is another example of a large-scaled network system. These systems consist of multiple service users and service providers that are geographically distributed. Cyber-physical systems, cognitive networks, participatory or opportunistic sensing network, software defined communications, among others, are other potential networked systems, and their number is expected to grow with the development in the fields of sensing, communication and computing [1].

In this section we discuss the different, central or non-central, static or dynamic approaches to optimizing this resource allocation for a requested collection of M2M service instances to be executed. Virtualization, as discussed in section 4.2, can form a basis for flexibly adding or removing processing capacity. In line with the OneM2M service layer standard (section 4.1.1), next to the application service component instances / stream processing operator instances, also infrastructure functions can flexibly be allocated to particular resources.

In the next sub-sections, we first sample the state of the art of central, often applied non-online in a one-shot way, optimization techniques that algorithmically optimize resource allocations among given criteria and

constraints (section 4.4.1), and then we devote a major part of the overall section to distributed, dynamic agent-based solutions (section 4.4.2).
In a final sub-section, we discuss pattern matching and forecasting based on very large dataset processing as a potentially related enabler (section 4.4.3).

4.4.1 Centralized, offline optimization methods

The network flow problem deals with the flow of resource within the network in an optimal manner which can be interpreted as the evaluation of the optimal point of operation of the network, and is one of the main issues of many research fields such as mathematical optimization, operations research, computer science and many engineering and real world problems. A part of the resource management problem is the network resource allocation, which is concerned with goals like task planning, resource deployment and resource planning, so that the system attains its objectives. The resource allocation involves the assignment of available resources to tasks in an optimal manner, and is a lower level function of the resource management [1].

In cases where the service collection to be executed is rather fixed, or changes only slowly, offline placement of the correspondingly needed processing can be considered as an option. Most notably, this option is considered for the placement of major telecom services, such as NFV (Network Function Virtualization), where a telecom network virtualizes the main telecom network functions (from central functions related to multimedia session management and user and device registries, down to even virtualized customer premises equipment, vCPE), or dedicated content distribution network functions (CDN). In general, virtual cloud service functions (vCSFs) are assumed to be placed elastically and on-demand, according to a central placement algorithm.

Yet, also for (M2M) service platforms, i.e. by which application service components are deployment as well as related service management functions part of the (distributed) platform, this option may become feasible, under the conditions that (1) new service types are introduced at a reasonably slow pace (e.g. a few service types a day, which is acceptable in many scenarios), and (2) that sufficiently fast algorithms/solver tools are applicable and incremental adaptation is possible in such semi-online setting.

A vast range of techniques can be applied to such resource allocation problems under various types of constraints and conditions, as researched in the longstanding Operations Research (OR) mathematics field. We discuss a few popular examples in the next paragraphs. In the M2MGrids project, we will need to research the exact problem formulation, assuming that the formulation will include additional real-world constraints which are passed on from the M2M Information Platform layer (WP2) for potentially taking into account for the resource allocation decision. E.g., certain behavior of the physical energy grid, demand behavior of consumers or response capacities of producers, as well as actual M2M service behavior in the energy domain, may translate into domain-agnostic constraints to which the M2M processing and networking resources will be provisioned (closer to) optimally.

A first popular class of approaches aims at deriving a **multi-constraint objective function minimization problem** for the placement decision of processing components and corresponding network transport, that can typically be considered to be solved as a **(Mixed) Integer Linear Programming ((M)ILP)** formulation, which can be solved by specialized Integer Programming tools such as CPLEX [156] or Xpress-MP [157]. The issue, however, is that for many problem formulations, especially if there are integer or non-linear properties to it, the complexity and computational cost quickly becomes high, preventing easy online application of the algorithms and tools. It is therefore important to formulate the problem in such a way as to make it reasonably efficiently solvable, with the aim for online application. E.g. in [158], an ILP approach is described to model and optimize cloud service distribution in terms of computer, storage and transport resources, relaxing the problem formulation in such a way as to make it solvable in polynomial time.

Another class of approaches expresses the (distributed) utility of a network/cloud/service platform's (distributed) resources for each actual user or instantiated service, as a market/economy-oriented approach optimizing the value that the resource spending results in. **Utility max-min fairness** of resource allocation to particular flows or processing parts as required by the services is one popular allocation criterion, whit max-min fairness defined as such allocation in which it is not possible to increase the allocation for one service or party without penalizing another which is already obtaining less of the resource. This property makes that the problem can typically be solved by means of an **iterative water-filling algorithm** variant [159]. While utility max-min fairness may clearly be a valid assumption for the assignment of resources in energy distribution, it may also be reflected in the overlay M2M communication network, and in particular in the M2M resource allocation decisions to be taken at the M2M Service Platform layer. Of this class of algorithms also distributed variants exist in literature [160].

For resource allocation problems, also methods deriving from Game Theory often also apply. An example is **Bilevel Programming/Optimization**, which considers the Stackelberg Game assumptions that two ('leader' vs. 'follower') levels make decisions sequentially [161]. For M2MGrids, this type of optimization problem can be considered at various levels. E.g., within the M2M Service Platform level, M2M infrastructure may be decided to be provisioned in a particular way which is optimal according to some overall assumptions, e.g. based on investment cost, offers to 3rd parties using the infrastructure, followed by cost-effective placement of the service components within that constrained provisioning. Another such duality may emerge from the combined action of managing a smart energy grid while serving and controlling it from an M2M service infrastructure.

A further step coping with multi-criteria, in a distributed way, are multi-agent systems, providing another range of approaches that may be leveraged in the M2M Service Platform layer.

4.4.2 Distributed optimization through multi-agent systems

The networked optimization problem is challenging since the network users with different objectives compete for shared resources [7], and it becomes more complex with the uncertainty and dynamic nature of the system, presence of distributed resources and the absence of central information in the system. This brings the need for the use of distributed optimization methods that can operate in parallel and independently utilizing only local information [8]. Some of the known challenges associated with these systems include: i) the assurance of network control performance under communication constraints between network nodes, ii) the effect of noise, delay, network overhead, latency, and packet dropouts, iii) coverage, consensus, and optimized cooperation in the system, and iv) information patterns for distributed and decentralized control of the network [9]. The determination of optimal operation point of these networks to ensure its efficiency/performance, robustness to uncertainty, and responsiveness to dynamic changes becomes of particular interest [1].

The network resources allocation and the optimization problems are generally dealt in a centralized manner [6]. To do so, the central processing node requires perfect information of the system (i.e., knowledge of current states of resources and tasks at every instance of decision-making) that we want optimize [2]. The most fundamental challenge to carry out optimization in these large scale networks is the high dimensionality. The computational complexity increases with the number of decision variables for the centralized optimization methods and thus the centralized optimization methods behave poorly for these kind of problems. For some complex problems, it is often impossible to define a single system wide metric for the optimization process [2]. Reliability is another issue associated with the centralized methods since these method have a single point of failure, i.e., if the main node fails, the whole system will fail. Due to the lack of global information, a centralized optimization method cannot also be performed in networks where it is not efficient or possible to share the objective function due to privacy or energy constraints [1][10].

In order to overcome these issues, the concept of decentralized and distributed optimization has emerged. A distributed optimization and resource allocation method, in addition to relieve the problem of high dimensionality and single point of failure, also addresses the issue of scalability and flexibility. Another advantage of decentralizing the optimization is when the resources and tasks are geographically separated, have different characteristics and abilities and different information.

The goal of distributed resource allocation is no longer to maximize utilization. A policy is needed to make resource allocation decisions when demand exceeds supply and not all needs can be met. More intelligent ways of allocating resources than simple best effort, or randomized allocation schemes are now considered by researchers. A social policy for resource distribution can be involved in these methods. A policy is simply a set of rules for allocation when resource demand exceeds resource supply. A blend of policies can also be implemented to meet a complex social goal [11]..

4.4.2.1 Agent characteristics

As people become aware of the software agents role importance, they also have become increasingly popular over the last few years. Software agents essentially are reasonable advanced and sophisticated computer programs capable of achieving certain tasks with some degree of autonomy, i.e. without constant human guidance or intervention [34].

Initially, most distributed problem-solving systems were based on a distinguished and independent software agent [36]. More recently, the focus has shifted to more autonomous agents with coordinated actions [37]. It has been proven that the real potential of this technology becomes unleashed when several software agents are put to use in the same environment. A group of agents is usually conceived as a multi-agent system, where the successful completion of their tasks is subject to the decisions and actions from more than one agent. Agents are forced to coordinate their activities so as to avoid negative interactions with their acquaintances and to exploit synergic potentials [35]. Also an important reason for the growing success of multi-agent technology is the potential to deal with high complexity problems [38].

A Distributed Problem Solving is based on the assumption that the problem solving task can be sub-divided in a set of modules or nodes, that will cooperate mutually, sharing knowledge about the problem and the solution as this is produced. The way of analysing, designing and building of distributed problem-solving systems has been changed due to the rapid growth of internet, mobile technologies and the design of intelligent applications. Most of these applications are intelligent, provided by distributed software agents.

When we say that an Agent acts on behalf of someone, we are implicitly admitting that it has a great degree of autonomy. According to Franklin and Gasser (1996) definition [39] an autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, in pursuit of its own agenda and so as to effect what it senses in the future.

In Multi-Agent Systems (MAS) the focus is on the coordination of the intelligent behaviours shown by a community of agents (autonomous or semi-autonomous) in such a way that they will be able to share knowledge, abilities, goals and plans in order to take actions or solve problems. Individual agents should be able to reason about the coordination processes involved. These autonomous or intelligent agents have the capability to observe, act and learn from their past experience.

Autonomy is essential in situations where the real-time human intervention is difficult or even impossible, like for instance controlling an aircraft or a space robot. Autonomy can also be key in situations where human operators can become “frozen” due to psychological pressure under the critical nature of the possible situation where a wrong decision could bring out heavy consequences. However, Agents may also be semi-autonomous. In this case they will partially depend on others and on human beings. Semi-autonomous agents don’t make sense individually because they depend on others to fulfil different functions or to validate agents’ decisions.

Usually, an agent doesn’t exist in isolation. It is common for it to have to interact with other agents, similar or different. In the same way that we talk about societies of individuals, we can also consider communities of agents with social skills. This is the basis for the concepts of MAS. Agents need to gather information and knowledge and therefore they need to establish communication between them. Several technological alternatives for communication are available: peer-to-peer (one to one) communication, broadcast (one for all) communication, blackboard based communication. Sockets, shared memory and other techniques can be used.

Cooperation is a concept that is defined at a higher level than communication. Agents cooperate in order to try and achieve their common goals or to get some benefit. Obviously, cooperation needs some kind of communication. A cooperative agent needs to know what its skills are and to have an idea about what tasks can be accomplished by other agents. This information can be stored in agent's data or be obtained by asking a specialized agent about it. Cooperative agents are able to share both tasks and results (data and knowledge).

Also agents can compete between themselves (they can represent, for instance, electronic commerce companies selling identical products in the web). In that case, agents must have increased abilities to monitor the environment evolution, namely of being able to watch closely over its competitors. Agents with social skills should be able to negotiate. Negotiation is based on announcements, proposals, offers and decisions and is usually bound by several restrictions (cost, time, quality, etc), conditions and penalties. Negotiation between agents has been identified since the beginning of the Distributed Artificial Intelligence, as can be seen in the Contract Net Protocol [41].

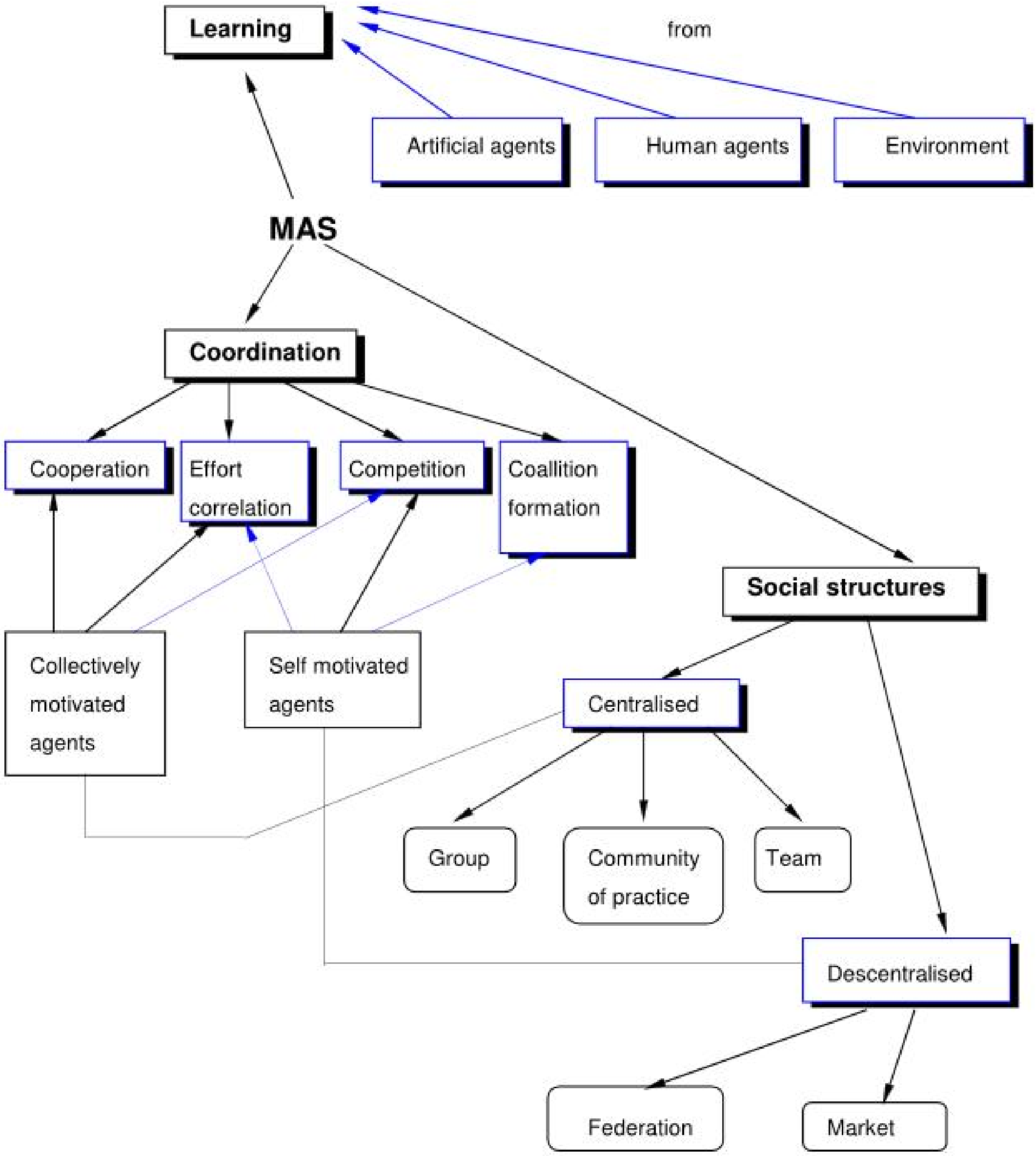


Figure 39. Cognitive interactions in a MAS [42]

At a decentralized level, the predominant MAS structure is the market, which organization implies the existence of suppliers, able to perform tasks to produce goods or services, and of buyers, namely agents that need the goods or services produced by the suppliers. The basic model associated with such a structure is the competitive MAS, with self-interested agents that are competing either to supply or to buy goods or services. Such a model is well suited for open systems. In such organizations, the agents in the system are dived into groups, each group having associated a single “facilitator” to which the agents surrender a degree of autonomy. A facilitator serves to identify the agents that join or leave the system and enables the communication with agents located in other groups [42].

Figure 39 depicts the basic aspects to be considered when studying, designing and developing MAS.

From J. Z. Hernández technical point of view, the inherent distribution allows for a natural decomposition of the system into agents that interact so as to achieve a desired global functionality. By this, reusability is promoted, not just by the agent’s modularity, but also based on their autonomy. In addition, the scalability of systems operating in highly complex domains can be improved by choosing among specific coordination models that harmonize agent activities with respect to the multi-agent system’s task. Through an adequate design of coordination mechanisms for a particular problem, he also believes that, from an economic point of view, multi-agent systems may tackle complex problem with an acceptable degree of performance, but at a lower cost than traditional solutions [35].

4.4.2.2 Cooperative problem solving in multi-agent systems

Distributed Constraint Optimization Problems (DCOPs) have been studied [25], [26], [27], [28] as a basic framework of cooperative problem solving in multi-agent systems [24]. DCOP provides a framework for multiple agents to coordinate and together compute the optimal choice over a set of alternatives encoded as a constraint network [29, 30]. The states of agents and the relationships between them are formalized into a constraint optimization problem that is solved by distributed search algorithms. The optimization problems and distributed search algorithms that are mainly contained in cooperative protocols of the multi-agent systems are the focus of these studies [24]. DCOP has many practical applications such as distributed resource scheduling [28], distributed meeting scheduling [31], distributed planning, resource allocation and sensor networks [32, 29]. The representation of DCOPs can be extended to meet particular problems. In these special cases, the solver is also adapted to the specific problem [24]. Resource Constrained DCOPs (RCDCOPs) make up a dedicated class of problems that explicitly represents shared resources as global constraints that can be decomposed into agents [33].

RCDCOP models resources by introducing virtual variables corresponding to each resource and imposes nary constraints among virtual variables and agents to guide the search mechanism. Agents must reason about the global optimality as well as the resources they consume which mean that adding resources significantly increases the combinatorial aspect of the problem. Some problem domains associate additional structure with resources and exploiting it can yield better efficient algorithms [34].

Distribution networks describe the flow of "a product" from a source to a destination. These networks can be modelled for many scenarios as, for instance, liquids flowing through a pipe, parts through assembly lines or current through a power network. Each edge in this network is a conduit for the "product" and has a certain capacity, which limits the amount of flow through it, and there's also the possibility of incurring a cost when the "product" passes through it. Vertices are the conduit junctions, which may consume some of the incoming flow and forward the rest on a subset of its incident edges, also known as flow conservation. In addition, the flow from the source to the destination must take the form of a tree, i.e. it must be acyclic – in the context of power distribution networks is also known as feeder tree [34].

The optimization problem to be addressed in such networks is to determine the configuration of the least cost feeder tree such that every sink in the network gets fed without violating the capacity of any edge, considering there is at least one such tree. This is essential for reconfiguring the network after its structure gets disrupted due to line failures. The resource constraints, i.e. capacity of edges, are the key source of complexity in modelling these networks [34].

4.4.2.3 Multi-agent systems in Microgrid management

Next to the allocation of resources for services executed in the M2MGrids Service Platform, another important example for the M2MGirds project is the application of agent principles in the actual services executed towards the energy grid resources, and in particular in Microgrid cases. While also applicable in scope of application domain information and knowledge (WP2), due to its importance as service resource consumption behaviour, and performance and distribution requirements, we discuss software agents also in the M2M Service Platform layer context here.

The power systems' sector is one of the main fields of application of multi-agent systems approaches [48, 49, 50, 51]. Approaching the power network as a set of sub-systems is being largely discussed among the scientific community, as it improves features such as network security, reliability, stability and minimization of the energy price against the traditional power systems [8] and consequently reduce the energy bill for consumers. These sub-systems can be referred as Microgrids, where a Microgrid can be defined as an enclosed area of the network, connected to it from a single point, that is capable of working isolated from the main network, and that responds to its orders and requests as a single entity, increasing the satisfaction and the quality of its services [52].

The use of multi-agent systems technology in controlling a Microgrid solves a number of specific operational problems, for instance the achievement from individual and collective energy efficiency increase, providing solutions with reasonable cost and retrofitting ability so that the solution rollout is economically viable, which creates the opportunity for an intelligent management of the system.

In the scope from M2MGrids is expected an active participation of the consumers in the available resource management targets not only with their individual benefits but also their contribution for the increase of the overall system efficiency. First of all, small DG (Distributed Generation) units have different owners, and different players operating the Microgrid, such as system operators, VPPs, energy suppliers, and consumers. Therefore several decisions should be taken locally, in a place where a centralized control is difficult to accomplish. Furthermore, Microgrids operating in a market require that the actions of the controller of each unit participating in the market should have a certain degree of intelligence.

Simulation combined with distributed artificial intelligence techniques appears to be an adequate form to study these markets and context evolution. Modelling the Microgrid environment with multi-agent systems enables model enlargements to include new players and allows studying and analysing both the individual and internal performance of each distinct player, as also the global and specific interactions between all the involved players [53].

Micro-agent systems will also have an important role in generation monitoring, forecasting and storage monitoring, as also in consumption monitoring and profiling and users preferences and context awareness.

Physical entities that can be considered as agents in our Microgrid are such as a protection relay, a sensor, a smart meter or any controller that controls directly particular power system component or part of the system. A Virtual entity that can be considered as an agent is a piece of software that receives inputs from an environment and produces outputs that initiate acting over it. Also an agent can be a combination of physical (computation architecture) and virtual one (a piece of software running on the computational architecture).

In conclusion, the goals from the integration with a multi-agent system platform within the scope from M2MGrids are related to the residential energy consumption optimization system, with learning and user modelling abilities, providing residential energy consumers with an automated easy to use system to ensure the optimization of the house energy resources, for instance the electric vehicles, which in this specific case will provide EVs negotiation, monitoring, charge and discharge management, guaranteeing to drivers a certain level of confidence while travelling with an EV.

4.4.2.4 Agent negotiation protocols and techniques

As a side note to this section, we introduce **GOAL**, **agent programming for decision making** for the nodes in the M2M Grids. GOAL [143] is a high level programming language to program rational agents that derive their choice of action from their belief and goals. The language provides the basic building blocks to design and implement rational agents by means of a set of programming constructs. These programming constructs allow and facilitate the manipulation of an agent's beliefs and goals and to structure its decision-making. The programming concepts of belief and goal incorporated into GOAL provide the basis for this form of reasoning and are similar to their common sense counterparts used every day to explain the actions that we perform. In addition, GOAL provides the means for agents to focus on their attention on specific goals and communicate at the knowledge level. This provides an intuitive basis for writing high-level agent programs.

Negotiation is one of the widely used techniques for allocating resources efficiently in distributed agent systems [144][145]. Agents in smart grids can negotiate on resources and make agreements on their allocation automatically. In such a dynamic environment, agents might not know each other's preferences and negotiation strategies. They need to interact with each other according to a predefined negotiation protocol. In the following part, multilateral negotiation protocols that can be applicable on resource allocation have been reviewed.

Mediated single text negotiation protocol [146] is a mediated negotiation protocol where the mediator initially generates a bid randomly and asks the negotiating agents to vote for this bid. Each agent can vote to either ``accept'' or ``reject'' in accordance with its negotiation strategy. If all negotiating agents vote to accept, the bid is labeled as the most recent mutually accepted bid. In further rounds, the mediator modifies the most recent mutually accepted bid by exchanging one value with another randomly in the bid and asks negotiating agents to vote for the current bid. This process continues iteratively until a predefined number of bids are reached.

For this protocol, two negotiating strategies are defined: "Hill-climber" and "Annealer". An agent employing hill-climber strategy only accepts a bid if its utility is greater than the utility of the most recent mutually accepted bid. If all agents employ hill climber strategy and the utility of initial bid is quite high for one of the negotiating agents, that agent may not accept other bids even though those bids might be better for the majority. By contrast, the agent employing annealer strategy calculates the probability of acceptance for the current bid based on the utility difference and a virtual temperature, which gradually declines over time. There is a higher probability when the difference is small and virtual temperature is high. That is, an agent employing annealer has a tendency to accept individually worse bids earlier so that the agents can find win-win bids later. Towards to the end of the negotiation, the agent has a tendency to accept only the bids whose utility is greater than the utility of the most recent mutually accepted bid.

Feedback and voting based protocol [147] is a variant of the mediated single text mediator. It allows negotiating agents not only to vote the mediator's bid but also to give some feedback (e.g. better, worse, same). Consequently, the mediator models agent's preferences during the negotiation and generate better bids for all agents over time. This protocol consists of two phases: "Feedback" and "Voting". In the first phase, the mediator generates bids and updates preference models for each agent based on their feedback. In the second phase, it generates estimated Nash bids maximizing the product of the estimated utilities of the agents with respect to the learnt model and asks the negotiating agents to vote them either to reject or accept. Similar to mediated single text mediator protocol, the most recently accepted bid is taken as the final agreement.

Consensus policy based negotiation protocol [148] is a multiparty negotiation protocol using Generalized Pattern Search (GPS), an optimization technique. According to the proposed protocol, a mediator agent chooses an initial contract randomly and accordingly proposes a mesh, a set of contracts, from the initial contract. Each agent privately informs the mediator about their preferences on these contracts. The mediator agent aggregates the individual preferences on each contract and finds the preferred contract with respect to the aggregated preferences. By applying the GPS, the mediator agent decides whether to continue the negotiation by generating a new mesh or complete the negotiation with the current preferred contract.

Mediated pre-negotiation protocol [149] is a multiparty negotiation protocol, in which the mediator proposes a large number of random contracts and the agents perform a runoff voting, i.e., they continue eliminating the candidates with the least votes until just one contract is left.

Sequential-offer protocol [150] is a generalization of the alternating offers protocol for multiparty negotiations. The agents make sequential offers in predefined turns or accept the underlying offer according to this protocol. In this work, a sequential projection method is suggested to employ to generate agent's bids.

Stacked alternating offers protocol [151] is an extension of sequential-offer protocol, which allows negotiating agents to walk away during the negotiation; thus, they can end the negotiation at any time.

Alternating multiple offers protocol [151] is a variant of alternating offers protocol for multilateral negotiations. Here, the emphasis is that all players will get the same opportunities with respect to bidding. A negotiation round consists of two phases: bidding and voting. In the bidding phase, all negotiating agents put their offer on the negotiation table and in the voting phase, all participants vote on all of these bids. An agreement is reached if all of the participants accepts an offer on the table. This is an iterative process continuing until reaching an agreement or reaching the deadline.

Unstructured Communication Protocol [152] is a novel multilateral protocol inspired from human negotiations. This protocol does not structure the negotiation process. That is, any agent may propose an offer at any time and offers can be retracted at any time. Agents can accept a given offer by repeating the same offer. When all agents propose the same offer, this offer is considered an agreement. In this protocol, agents can remain silent and wait for the other agents. Designing an agent having the intelligence to deal with the uncertainties in UCP is quite a challenge.

4.4.3 Pattern matching and forecasting for very large datasets

According to the U.S. Chamber of Commerce 90% of all the digital data in the world has been created in the last two years. The exponential growth of the volume, velocity and complexity of data collections [59] in such a short period has posed a set of new challenges in the field of data analysis and computational intelligence. The two most pressing issues identified are a) the volume of data that can be stored locally is often much smaller than the size of the whole dataset and b) the speed with which new data arrives is often greater than the processing speed that even supercomputers can achieve. For a big part of computational intelligence techniques, these issues are vital since these techniques are based on the principal of "stationary datasets that are always available in their entirety", which is no longer valid. Very large datasets need to be processed in distributed environments, where either the data storage/sourcing or the processing is distributed.

The variety of events in the M2M/IoT context also makes it harder to define patterns for analysis; Bruns et al. [155] have defined models for the M2M domain, M2M machine states and M2M events based on DS-EPL. They emphasize limitations such as the difficulty for non-technical users to define event patterns by using a textual notation, except for event pattern actions that are supported. This thus is clearly an area where the M2MGrids project has an opportunity to innovate beyond the state of the art.

In the following sub-sections, we look at the distribution issue from consecutive historic perspectives of very large dataset handling.

4.4.3.1 Local parallel processing

The first attempts to create distributed versions of pattern matching algorithms and methods followed the path of traditional parallelism computing techniques [82] which can be applied to most types of algorithms. Under this setting, all the computing units share the same memory and the communication cost among computing units is negligible. Most data analysis / pattern matching algorithms can be rewritten on the fine-grained parallelism level. That is, a level of parallelism that allows the subtasks to communicate many times per second. Large multi-core computing clusters are nevertheless quite expensive and thus the alternative of developing parallel versions of data analysis algorithms on graphics processing units (GPU) composed of hundreds of cores.

4.4.3.2 Distributed processing

Although GPU processing offered a temporary solution to the analysis and mining of large datasets, it remained an expensive option with limited scalability. In order to achieve, at least theoretically, unlimited scalability a decentralized solution was needed. Such a solution should operate on the principal of a number of equal processing peers that are able to communicate within network limitations. Already in 1997 Guo et al. [83] and Stolfo et al. [84] presented the first frameworks of this type. The common characteristic of these solutions was the fact that the data was partitioned in non-overlapping subsets and each processing unit had access to one of them. The partitioning of the data can be either horizontal (subsets of instances) or vertical (subsets of attributes). In such a scheme, each processing unit would produce a model of the available data and at the end a central node would combine those partial results. The combination of partial results can be implemented in two ways: a) with non-adaptable methods or b) with meta-learning methods.

Under the family of non-adaptable methods fall all fixed rules that are defined as functions, receive as inputs the outputs of the set of learnt models and produce a unique output when combined [85]. Some of the most well-known rules that are being used in pattern analysis and matching for this purpose are: majority voting, the sum and product rules and the max and median rules. Meta-learning, on the contrary, combines several models by building a separate meta-model with inputs that are the outputs of the various models and the output is the desired outcome [86]. In the last few years different variations of meta-learning based techniques have appeared each one with a variety of advantages and disadvantages [87]. The most well-known are: stacked generalization [88], knowledge probing [89], effective stacking [90] and distributed boosting [91] The "meta-learning" approach was widely adopted [92][93] since it was based on the already known idea of ensemble learning [94], it did not however take into account all the possibilities of node failures, network link failures etc., still assuming total availability of the partitioned data at all times.

Grid computing solutions began to gain attention recently. In this state of the art review we do not consider such solutions an entirely different family, but rather an option that utilizes the modern network and hardware technologies to address the issues of the already existing distributed pattern analysis and matching algorithms. The main idea of such an approach is that one does not necessarily need to have their own distributed computing infrastructure since there are billions of PCs around the world, and many are idle most of the time. Internet computing exploits these idle workstations and PCs to create powerful distributed computing systems with global reach. The most popular grid-based approaches have been based on the well-known pattern analysis and machine learning toolkit WEKA [95][96]. These solutions extend the algorithms included in the WEKA toolkit into services that can be distributed across computers in an ad hoc grid environment.

4.4.3.3 Sensor data

Sensor data have always been a separate category of data because of their functional nature (time series). Although the storage of time series data is not particularly demanding, the indexing and retrieval of such data have become very important bottlenecks.

The first important breakthrough in the field came only ten years ago when Sakurai et al. [97] proposed FTW, the first scalable method for the computation of similarities between data sequences based on dynamic time-warping (DTW). The most important advantages of FTW were the fact that it was significantly faster than the original DTW, it could guarantee no false dismissals and it could handle sequences of arbitrary lengths. Lin et al. [98] presented in 2007 the SAX time series representation. The most important novelty introduced by SAX was the representation of a real-valued data sequence of arbitrary length w through a discrete "alphabet" (set of symbols), thus reducing its length to a number $n<<w$. Being symbolic SAX also allowed for the use of data mining algorithms that are not well defined for real-valued data (i.e. Markov models). Shieh et al. [99] extended the SAX representation to a multi-resolution variant called iSAX in order to make it faster and scalable. iSAX allows for efficient indexing of data sequences since the number of bits used for the representation can be dynamically changed, corresponding to a desired resolution. Furthermore, iSAX has the intrinsic properties to allow comparisons of sequences of different resolutions and can even compute lower bounds of comparison metrics, thus facilitating fast approximate searches. Camerra et al. [100] achieved another major breakthrough and further extended the SAX representation to its iSAX 2.0 variant, which further reduced the I/O costs of indexing using a) a new algorithm for time series bulk loading and b) a new splitting policy for the internal nodes of the index. SAX and its variants have been extensively used in a variety of time series data mining tasks such as ranking, classification [101] and forecasting [102].

Finally, one of the most recent developments in the field concerns the Hadoop TS [103] framework. Hadoop TS stores data based on the concept of application-specific collections of time series called “Buckets”. These “Buckets” allow basic but important operations such as changing the sampling rate and filtering of the time series. Data synchronization, consistency checks as well as tracking of processing steps have also been implemented as a part of Hadoop TS. Thus, a complete framework is being offered for the development and application of time series data mining algorithms.

4.4.3.4 M2M Grids data streams and services

At this stage of the project we expect the possibility of various data streams of different nature being utilized within the M2M Service Platform layer. The exact data streams and the exact services that the final demonstrator will consist of are not yet fully defined, but the dynamic allocation of resources and deployment of modules is a desired result. This section has therefore focused on the span of algorithms that may become relevant in the later stages of the work. Indeed, in the context of M2M Grids, techniques related to the analysis and modelling of large data streams (time series data) have the potential to become essential parts of the service platform, and may e.g. be deployed as components / stream processing operators in the M2M Service Platform layer. Especially algorithms related to load balancing (evolutionary computing), predictions (forecasting) and data aggregation (data modelling) may become key enablers for efficiently executing M2M services.

4.5 M2MGrids Service Platform-relevant programming languages, frameworks and service modeling

For the M2M Service Platform layer, we will also need to decide on languages and knowledge/information representation conventions for service logic, at least when passed on from the WP 2 layer, i.e. as a service collection specification to be fulfilled and executed by the Service Platform layer in the underlying execution environments, engines and communication network resources. Various existing representation formalisms and languages may prove to be valid candidates for this, ranging from RDF, various graph representations, service specification languages, specialized XML schemes, up to (dedicated or general purpose) programming languages for the considered stream processing frameworks. We discuss a first sample of such representation formalisms and languages in this section.

4.5.1 Base Knowledge Representation Languages

In this section, we describe the base ontology languages to increase the machine interpretability in the service platform layer of the M2MGrids. As a representation of knowledge with its semantics, ontology provides a common understanding of knowledge concerning the domain of interests. It contains the specification of concepts and their relationship. There are a variety of markup languages to represent an ontology such as SHOE [134], DAML [135], OIL [136], RDF [137], RDF Schema [138], OWL [139]. Among these languages, we briefly address RDF, RDF Schema and OWL, as being the most widely used in the community.

RDF, Resource Description Framework [137], is a W3C recommendation for describing resources on the Web. It is worth noting that any concrete/physical or abstract concept can be treated as an RDF resource. Each resource is identified by a unique name, URI. RDF consists of statements which are in a form of triples. A triple consists of subject, predicate and object. A predicate establishes binary relations between subjects and objects. Subject and predicate should be a resource while the object can be a resource or a literal such as a string or a float [109]. RDF documents are structured as XML documents.

RDF Schema [138] is language for describing RDF vocabularies. This language can be considered as an extension of RDF. While RDF itself does not allow us to represent a hierarchy of classes or define our RDF vocabularies, RDF Schema support to create our RDF vocabularies and present hierarchical organization by using “subClassOf” property. With RDF Schema, we can use some constraints on properties such as domain and range. Domain denotes the type of subject and range specifies the set of all legal values for the object. The expressiveness of RDF and RDF Schema is inadequate for more sophisticated tasks. We need a more expressive language that provides additional properties and vocabularies.

The Web Ontology Language (OWL) [139] is a W3C standard recommendation for formalizing an ontology. Expressions in OWL is based on description logics. It enhances RDF and RDF Schema by introducing additional constructs such as advanced properties, constraints, restrictions and so on. These additional features of OWL let us perform more sophisticated reasoning. In OWL, we have two types of properties: *object property* and *data property*. The former relates the instances of two objects where the later relates an instance with a data type such as string, float and so on [140]. Particular to object properties, there are some characteristics such as symmetric, transitive, functional, and inverse functional properties.

In order to obtain a particular information from the given RDF statements (e.g. RDF triples –subject, predicate and object), we need a query language
Simple Protocol and RDF Query Language (SPARQL) [141] is a W3C recommendation to query the data stored in RDF format. Similar to SQL, it has ``Select'' and ``where'' clauses. In “Select” clause we use the variables to be returned in a result set and in “where” clause we specify some triple patterns as a condition. Moreover, SPARQL provides some filters for RDF literals such as restricting the values of strings/ numeric values.

To increase the expressivity even more, we can add additional rules with a semantic rule language. **Semantic Web Rule Language (SWRL)** [142] is an OWL-based rule language. It is worth noting that the ontology reasoners such as KAON2, Jena, Pellet, have limited support for the rule languages. Selecting the rule language and compatible ontology reasoner depends on what rules required to express in a given domain.

4.5.2 Stream Processing Frameworks and Languages

In section 4.3.3 we discussed state of the art distributed stream processing frameworks, focusing on their distributed architecture and provided fault tolerance guarantees. Here we describe specifically the frameworks and languages used to write the actual stream processing operators or stream transformations. Unsurprisingly, many stream processing frameworks have their roots in the functional programming paradigm, conceiving computations as transformations on (potentially infinite) input streams.

In **Apache Spark Streaming** [68], stream operators are written in a general-purpose programming language like Java or Scala. Recall that Spark Streaming is built on Spark. The basic unit of processing in Spark is an RDD (resilient distributed dataset), which is a collection of elements partitioned across a cluster of machines. RDDs can be operated on in parallel. RDDs support two kinds of operations: *actions*, which transform an RDD into a concrete value (example operators include “reduce”, “count”, “foreach”, etc.), and *transformations*, which transform an RDD into a derived RDD (example transformations include “map”, “filter”, “groupBy”, “sort”, etc.). RDD transformations are "lazy" in the sense that they are recomputed for each action, unless they are explicitly cached, in which case they are kept in memory so that later processing does not need to reconstruct the RDD. The cache is fault-tolerant: if a partition of the RDD is lost, that part is recomputed from the original transformations. By default, transformations such as map, filter, etc. can only operate on local data. There are no shared variables among multiple transformations. Spark does offer “accumulators”, which are special shared variables that can be used to implement system-wide counters or sums that can be aggregated in parallel. Spark Streaming builds directly on Spark’s abstractions. A DStream supports many of the transformations of RDDs, e.g. map, filter, union, count, reduce, etc. DStream processing is stateless by default, but state can be kept between successive RDDs using a transformation known as “updateStateByKey”. As in many other stream processing systems, Spark Streaming supports window operations where the window length and sliding interval are configurable. Spark SQL [74] is a Spark extension allowing the use of declarative SQL queries to query RDDs. A third-party open source project maintained by Intel known as Spark-StreamingSQL [75] aims to bring the benefits of declarative SQL queries to Spark Streaming.

Google Cloud Dataflow [76] is a stream processing platform offered as a managed service. Internally it uses Google Millwheel [61] and FlumeJava [77]. Developers construct data processing “pipelines” of “transforms” (stream operators) in Java. The core data type exposed by the API is a PCollection, supporting a variety of “transforms” which Cloud Dataflow classifies as either “core” or “composite” transforms. Core transforms include “ParDo”, “Combine”, “GroupByKey” and “Flatten”. “ParDo” transforms can include general-purpose Java code. Composite transforms provide “out of the box” complex transformations such as counting elements in a collection, dividing a collection into quantiles, finding the top (or bottom) N elements in a collection, and performing basic mathematical combinations on numerical data. Google’s Cloud Dataflow API has been open sourced, and next to deploying Cloud Dataflow pipelines on Google’s hosted Compute infrastructure, pipelines can also be run on one’s own Apache Spark or Apache Flink infrastructure.

A key issue with APIs that express data processing pipelines as a series of functional transformations is that each transformation introduces overhead related to I/O (due to checkpointing to stable storage, network communication, data serialization, etc.). Often, reducing the amount of transformations in a pipeline can drastically reduce the end-to-end processing latency. FlumeJava [77] is a compiler developed by Google that allows chaining together series of transformations using a high-level API that implicitly builds up a query plan. This query plan is subsequently rewritten to reduce the total number of stages. Apache Crunch [78] is an open source project with the same goals as FlumeJava. Both FlumeJava and Crunch were designed primarily for MapReduce batch jobs, but the same principles carry over to operators in stream processing pipelines.

CQL (Continuous Query Language) [79] is a SQL-like declarative query language designed for processing unbounded data streams, rather than bounded tables in a relational database. It is used in a.o. Oracle’s commercial Complex Event Processing (CEP) product known as WebLogic Event Server. CQL exposes three types of operators: stream-to-relation, relation-to-relation and relation-to-stream operators. Stream-to-relation operators use sliding windows to turn an unbounded stream into a finite bag of tuples. Relation-to-relation operators include the standard relational algebra operators such as selection, projection and join. Relation-to-stream operators turn a relation into one of three possible types of streams: IStreams model insertions (new tuples not present at an earlier time), DStreams model deletions (old tuples that were present at an earlier time, but not anymore) and RStreams model relations (all tuples of the relation at each time step).

Microsoft’s Reactive Extensions (Rx) for .NET [80] is a collection of APIs that enable the asynchronous processing of data streams without having to resort to traditional callback-based programming. Instead, developers can use the framework’s many high-level functions to transform the input stream. Microsoft’s Orleans framework [71] extends its Virtual Actors abstraction to Streams with an Rx-style API. Like their virtual actors, discussed earlier, Orleans streams are virtual and conceptually always exist. They are created on-demand and are sufficiently lightweight to be created and destroyed at a high update rate. A runtime component called Streaming Pub Sub can be used as a publish/subscribe system to match stream consumers with stream producers.

EP-SPARQL [104] is a unified language for complex event processing and stream reasoning, grounded in logic programming. The ETALIS engine is a module running on top of a prolog runtime. Standard RDF predicates are extended with a timestamp tuple. These timestamp facilitate the time-related processing. Also background knowledge is represented as prolog rules using the semweb package of prolog. Complex events become prolog goals. Rules are binarized to ease the the implementation of the operators. Rules are fired as soon events arrive. This helps also in the reuse of subexpressions. The backward-chaining rules compute goals in a forward chaining way, to compute the complex events incrementally.

The **SpaceTime Oriented Macroprogramming language (STOP)** [105] is macro programming language to program WSN applications. The language threats space-time as first class citizens. This allows the user to specify the data collection and processing from a global viewpoint and abstract away the sensor nodes. The main advantage if this approach is the procedural programming style for the top layer application, cfr, dataflow and declarative programming with the other frameworks.

4.5.3 Standards for service specification

Open Geospatial Consortium (OGC) [106] provides a full suite of open standards for geospatial services and GIS data processing. Even though the focus to geospatial content, most of the standards have a broader applicability.

SensorML [107] provides and XML encoding for describing sensors and measurement processes. A process is defined by its inputs, outputs, parameters and metadata. Once describe in SensorML the process is discoverable and executable. Processes can be chained in complex graph and form the base for higher level standards like the **Sensor Observation Service (SOS)**, **Sensor Planning Service (SPS)** and **Sensor Alert Services (SAS)**. All these services provide a standard interface between application/services and the sensor and processing network.

Sensor Event Service (SES) is a Pub/sub messaging framework that allows to use filter criteria to subscribe to particular events. It combines OASIS Web Service Notification, Observation&Models and the OGC filter encoding specification (FES). The SES executes the filters, as event stream processing units described in EML (Event pattern Markup Language). Topics are used as identifiers. Topics are grouped as: measurements, sensor management, and expiration information. Topics are defined according the WS-topics standard. SOAP is used as the standard encoding for all message send to the SES. Optionally, XMPP bindings are available as well. SES supports multi-stage content based filtering: level 1, uses xpath expressing to define the filter; level 2: uses OGC filter encoding, supporting temporal, spatial and boolean filtering (similar to the filtering in SAS); level 3: uses complex event processing and event stream processing to derive events from the lower level events, the event pattern are encoded in EML.

Web Processing Service (WPS) provides an API for request/response based process chaining. Even though its original purpose was to standardize complex batch processes, e.g. gcube provides a WPS backend on top of Hadoop [108], it was demonstrated that the WPS can support data streaming and near real-time geoprocessing. [109]. A proposal was made [110] to extend OGC with an event-driven architecture around and Enterprise Service bus, that controls both event-processing request on live data via the SES API and combine this with static data fetches through the Web Coverage Service (WCS)

Ontology Web Language for web services (OWL-S) [111] is a W3C recommendation to semantically describe web services in a computer interpretable way. The aim of the semantic web is to automate the process of web service discovery, invocation and composition and interoperation. The ontology covers three aspects of the services: what it does (ServiceProfile), how to access it (ServiceGrounding), how it works (ServiceModel). The services are described as composition of processes, including control constructs that instrument the execution environment to invoke the processes (within a so called process bag) in a particular way e.g. Sequence, for sequential execution; Split for parallel execution with join after all processes are completed; If-Then-Else for conditional execution. Competing ontologies are WSMO [112], SAWSDL [113] targeting WSDL services and hRESTS [114] and WSMO-Lite [115][116] targeting services with a REST API.

References

[1] HomChaudhuri, Baisravan ,Price-Based Distribution Optimization in Large-Scale Networked SystemsPh.D., UNIVERSITY OF CINCINNATI, 2013,
[2] R. A. Giliano and P. Mitchem, Market based control: A paradigm for distributed resource allocation, ch. Valuation of Network Computing Resources, pp. 28–52. World Scientific Publishing, 1996.
[3] K. Harty and C. D., Market based control: A paradigm for distributed resource allocation, ch. A market approach to operating system memory allocation, pp. 126–155. World Scientific Publishing, 1996.
[4] D. F. Ferguson, C. Nickolaou, J. Sairamesh, and Y. Yemini, Market based control: A paradigm for distributed resource allocation, ch. Economic Models for Allocating Resources in Computer Systems, pp. 156–183. World Scientific Publishing, 1996.
[5] K. Kuwabara, T. Ishida, Y. Nishibe, and T. Suda, Market based control: A paradigm for distributed resource allocation, ch. An equilibratory market-based approach for distributed resource allocation and its application to communication network control, pp. 53–73. World Scientific Publishing, 1996.
[6] A. D. Baker, Market based control: A paradigm for distributed resource allocation, ch. A case study where agents bid with actual costs to schedule a factory, pp. 184–223. World Scientific Publishing, 1996.
[7] H. Terelius, U. Topcu, and R. Murray, “Decentralized multi-agent optimization via dual decomposition,” in World Congress of the International Federation of Automatic Control, IFAC, 2011.
[8] D. P. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” Selected Areas in Communications, IEEE Journal on, vol. 24, no. 8, pp. 1439–1451, 2006.
[9] J. Baillieul and P. Antsaklis, “Control and communication challenges in networked real-time systems,” Proceedings of the IEEE, vol. 95, no. 1, pp. 9–28, 2007.
[10] S. Sundhar Ram, A. Nedi´c, and V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” Journal of optimization theory and applications, vol. 147, no. 3, pp. 516–545, 2010.
[11] Jeffrey Shneidman, Chaki Ng, David C. Parkes, Alvin Auyoung, Alex C. Snoeren, Amin Vahdat, Brent Chun, Why Markets Could (But Don’t Currently) Solve Resource Allocation Problems in Systems, Proceedings of the 10th USENIX Workshop on Hot Topics in Operating Systems, 2005.
[12] Y. Fu, J. Chase, B. Chun, S. Schwab, and A. Vahdat, “SHARP: An architecture for secure resource peering,” in Proceedings of the 19th ACM Symposium on Operating Systems Principles, October 2003.
[13] S. Grinstein, J. Huth, and J. Schopf, “Resource predictors in hep applications,” in Proc. of Computing in High Energy and Nuclear Physics (CHEP), September 2004.
[14] G. Hardin, “The tragedy of the commons,” in Science, 162(1968):1243-1248.
[15] J. Momoh, “Smart grid design for efficient and flexible power networks operation and control,” in Power Systems Conference and Exposition. PSCE’09. IEEE/PES, pp. 1–8, Ieee, 2009.
[16] E. Santacana, G. Rackliffe, L. Tang, and X. Feng, “Getting smart,” IEEE Power and Energy Magazine,, vol. 8, no. 2, pp. 41–48, 2010.
[17] P. Nguyen, W. Kling, G. Georgiadis, M. Papatriantafilou, L. Bertling, et al., “Distributed routing algorithms to manage power flow in agent-based active distribution network,” in Innovative Smart Grid Technologies Conference Europe (ISGT Europe), IEEE PES, pp. 1–7, IEEE, 2010.
[18] G. Lee, N. Tolia, P. Ranganathan, and R. H. Katz, “Topology-aware resource allocation for data-intensive workloads,” in Proceedings of the first ACM asia-pacific workshop on Workshop on systems, pp. 1–6, ACM, 2010.
[19] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, “Dominant resource fairness: Fair allocation of heterogeneous resources in datacenters,” University of California, Berkeley, EECS Department, Technical Report UCB/EECS-2010-55, 2010.
[20] H. N. Van, F. D. Tran, and J.-M. Menaud, “Sla-aware virtual resource management for cloud infrastructures,” in Computer and Information Technology, 2009. CIT’09. Ninth IEEE International Conference on, vol. 1, pp. 357–362, IEEE, 2009.
[21] Gartner Says Nearly Half of Large Enterprises Will Have Hybrid Cloud Deployments by the End of 2017. https://www.gartner.com/newsroom/id/2599315. Consulted May 2015
[22] M. Steiner, B. G. Gaglianello, V. K. Gurbani, V. Hilt, W. D. Roome, M. Scharf, and T. Voith, "Network-aware service placement in a distributed cloud environment," in Proc. ACM SIGCOMM'12, 2012, pp. 73-74.
[23] Cornelius Toole, Jr.Andrei Hutanu, Network Flow Based Resource Brokering and Optimization Techniques for Distributed Data Streaming Over Optical Networks, Mardi Gras Conference, 2008.
[24] Matsui, T.; Matsuo, H., "Considering Equality on Distributed Constraint Optimization Problem for Resource Supply Network," Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on , vol.2, no., pp.25,32, 4-7 Dec. 2012
[25] R. Mailler and V. Lesser, “Solving distributed constraint optimization problems using cooperative mediation,” in 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, 2004, pp. 438–445.
[26] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo, “Adopt: Asynchronous distributed constraint optimization with quality guarantees,” Artificial Intelligence, vol. 161, no. 1-2, pp. 149–180, 2005.
[27] A. Petcu and B. Faltings, “A scalable method for multiagent constraint optimization,” in 19th International Joint Conference on Artificial Intelligence, 2005, pp. 266–271.

[28] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg, "Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks," Artificial Intelligence, vol. 161, no. 1-2, pp. 55–87, 2005.
[29] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. AI Journal, 161:149–180, 2005.
[30] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In IJCAI, Edinburgh, Scotland, Aug 2005.
[31] R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham. Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling. In AAMAS, 2004.
[32] R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham, "Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling," in 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, 2004, pp. 310–317.
[33] T. Matsui, M. Silaghi, K. Hirayama, M. Yokoo, and H. Matsuo, "Resource constrained distributed constraint optimization with virtual variables," in 23rd AAAI Conference on Artificial Intelligence, 2008, pp. 120–125.
[34] KUMAR, Akshat; Faltings, B.; and Petcu, A.. Distributed Constraint Optimization with Structured Resource Constraints. (2009). , 2 , 923-930. Research Collection School Of Information Systems.
[35] Hernández, Josefa Z., Sascha Ossowski, and Ana Garcıa-Serrano. Multiagent architectures for intelligent traffic management systems. Transportation Research Part C: Emerging Technologies 10.5 (2002): 473-506.
[36] Von Martial, F., 1992. Co-ordinating Plans of Autonomous Agents. LNAI 610. Springer, Berlin.
[37] Ephrati, E., Pollack, M., Ur, S., 1995. Deriving multi-agent co-ordination through filtering strategies. In: Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI-95). Morgan Kaufmann, Los Altos, CA, pp. 679–685.
[38] Ossowski, S., Garc_ıa-Serrano, A., 1999. In: Social Structure as a Computational Co-ordination Mechanism in Societies of Autonomous Problem-solving Agents. Intelligent Agents V. Springer-Verlag, pp. 133–148.
[39] S. Franklin, A. Graesser; Is it an Agent or just a Program?: A Taxonomy for Autonomous Agents; Third International Workshop on Agent Theories, Architectures and Languages; Springer-Verlag; 1996
[40] Zou, Xi and Levinson, D. (2003), Vehicle Based Intersection Management with Intelligent Agents, ITS America Annual Meeting Proceedings.
[41] R. Davis; Report on the Workshop on Distributed Artificial Intelligence; SIGART Newsletter, n. 73, pp. 42-52; October 1980
[42] Adina Magda Florea, Introduction to Multi-Agent Systems, International Summer School on MultiAgent Systems, Bucharest, 1998.
[43] Nguyen-Duc, M., Guessoum, Z., Mari, O., Perrot, J.-F., Briot, J.-P., Duong, V., 2008. Towards a reliable air traffic control. In: AAMAS'08: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 101–104.
[44] Callantine, T. CATS-based Air Traffic Controller Agents. NASA Contractor Report 2002-211856. Moffett Field, CA: NASA Ames Research Center, 2002.
[45] Adrian K. Agogino, Kagan Tumer, A multiagent approach to managing air traffic flow. Autonomous Agents and Multi-Agent Systems Journal, January 2012, Volume 24, Issue 1, pp 1-25.
[46] Danko A. Roozemond, Jan L.H. Rogier., 2000. Agent controlled traffic lights. ESIT 2000, 14-15 September 2000, Aachen, Germany.
[47] Zou, Xi and Levinson, D. (2003), Vehicle Based Intersection Management with Intelligent Agents, ITS America Annual Meeting Proceedings.
[48] T. Logenthiran, Multi-Agent System for Control and Management of Distributed Power Systems, Ph.D., National University of Singapore, 2012.
[49] Ming, Zhou, et al. A multi-agent based dispatching operation instructing system in electric power systems. Power Engineering Society General Meeting, 2003, IEEE. Vol. 1. IEEE, 2003.
[50] Nagata, Takeshi, and Hiroshi Sasaki. A multi-agent approach to power system restoration. Power Systems, IEEE Transactions on 17.2 (2002): 457-462.
[51] H. Morais, et.al., Multi-Level Negotiation in Smart Grids for VPP Management of Distributed Resources, IEEE Intelligent Systems, Special Issue "Sustainable Energy and Distributed AI", vol. 27, no. 6, pp.. 8-16, November - December 2012.
[52] B. Lasseter,Microgrids [distributed power generation], Power Engineering Society Winter Meeting, 2001 (1), pp. 146-149. Lasseter, R., Akhil, A., Marnay, C.,
[53] Oliveira, Pedro, et al. A multi-agent based approach for intelligent smart grid management. Power Plants and Power Systems Control. Vol. 8. No. 1. 2012.
[54] S. Russel, P. Norvig, Artificial intelligence – A modern approach, Prentice Hall, 1995.
[55] P. Maes, "Artificial life meets entertainment: Life like autonomous agents", Communications of the ACM, vol. 38, no. 11, pp. 108-114, 1995.
[56] Ferrando, Sergio Pajares, and Eva Onaindia. Context-aware multi-agent planning in intelligent environments. Information Sciences 227 (2013): 22-42.
[57] OneM2M Published Specifications, http://www.onem2m.org/technical/published-documents
[58] OneM2M TS-0001 Functional Architecture, V1.6.1, http://www.onem2m.org/images/files/deliverables/TS-0001-Functional_Architecture-V1_6_1.pdf, January 2015

[59] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107-113.
[60] L. Neumeyer and A. Nair, 2010. S4: Distributed Stream Computing Platform. KDCloud 2010.
[61] Tyler Akidau, Alex Balikov, Kaya Bekiroğlu, Slava Chernyak, Josh Haberman, Reuven Lax, Sam McVeety, Daniel Mills, Paul Nordstrom, and Sam Whittle. 2013. MillWheel: fault-tolerant stream processing at internet scale. Proc. VLDB Endow. 6, 11 (August 2013), 1033-1044.
[62] Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthik Ramasamy, Jignesh M. Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat, Sailesh Mittal, and Dmitriy Ryaboy. 2014. Storm@twitter. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data (SIGMOD '14).
[63] Apache Software Foundation. Apache Samza. http://samza.apache.org/ [Retrieved July 2015]
[64] Apache Software Foundation. Apache Kafka. http://kafka.apache.org/ [Retrieved July 2015]
[65] Apache Software Foundation. YARN and Apache Hadoop. http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html [Retrieved July 2015]
[66] Amazon. Amazon Kinesis. http://aws.amazon.com/kinesis/ [Retrieved July 2015]
[67] Amazon. Amazon Lambda. http://aws.amazon.com/lambda/ [Retrieved July 2015]
[68] Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, and Ion Stoica. 2013. Discretized streams: fault-tolerant streaming computation at scale. In Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles (SOSP '13).
[69] Apache Software Foundation. Apache Flink. http://flink.apache.org [Retrieved July 2015]
[70] Microsoft. Azure Service Fabric. http://azure.microsoft.com/en-us/campaigns/service-fabric/ [Retrieved July 2015]
[71] Sergey Bykov, Alan Geller, Gabriel Kliot, James R. Larus, Ravi Pandya, and Jorgen Thelin. 2011. Orleans: cloud computing for everyone. In Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC '11).
[72] Carl Hewitt, Peter Bishop, and Richard Steiger. 1973. A universal modular ACTOR formalism for artificial intelligence. In Proceedings of the 3rd international joint conference on Artificial intelligence (IJCAI'73). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 235-245.
[73] Eclipse Foundation. Vert.x. http://vertx.io [Retrieved July 2015]
[74] Apache Software Foundation. Spark SQL. https://spark.apache.org/sql/ [Retrieved July 2015]
[75] Spark-StreamingSQL. https://github.com/Intel-bigdata/spark-streamingsql [Retrieved July 2015]
[76] Google Inc. Cloud Dataflow https://cloud.google.com/dataflow/ [Retrieved July 2015]
[77] Craig Chambers, Ashish Raniwala, Frances Perry, Stephen Adams, Robert R. Henry, Robert Bradshaw, and Nathan Weizenbaum. 2010. FlumeJava: easy, efficient data-parallel pipelines.SIGPLAN Not. 45, 6 (June 2010), 363-375.
[78] Apache Software Foundation. Apache Crunch. https://crunch.apache.org/ [Retrieved July 2015]
[79] Arvind Arasu, Shivnath Babu, and Jennifer Widom. 2006. The CQL continuous query language: semantic foundations and query execution. The VLDB Journal 15, 2 (June 2006), 121-142.
[80] Microsoft. Rx (Reactive Extensions). https://rx.codeplex.com/ [Retrieved July 2015]
[81] Chen C.L.P., Zhang C.-Y.: Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences 275, 314–347 (2014)
[82] Kumar V., Grama A., Gupta A., Karpis G.: Introduction to parallel computing: design and analysis of parallel algorithms. Addison Wesley, Reading (2003)
[83] Guo Y., Ruger S., Sutiwaraphun J., Forbes-Millott J.: Meta-learning for parallel data mining. In: Proceedings of the seventh parallel computing workshop, pp 1–2 (1997)
[84] Stolfo S., Tselepis A., Lee W., Fan D., Chan P.: JAM: java agents for meta-learning over distributed databases. In: Proceedings of the third international conference on knowledge discovery and data mining (1997)
[85] Kittler J., Hatef M., Duin R.P.W., Matas J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20(3), 226–239 (1998)
[86] Zeng L., Li L., Duan L., Lu K., Shi Z., Wang M., Wu W., Luo P.: Distributed data mining: a survey. Information Technology and Management, pages 403–409 (2012)
[87] Peteiro-Barral D., Guijarro-Berdias, B.: A survey of methods for distributed machine learning. Progress in Artificial Intelligence, Vol. 2, No. 1, pp. 1-11 (2013)
[88] Wolpert D.H.: Stacked generalization. Neural Netw. 5(2), 241–259 (1992)
[89] Guo Y., Sutiwaraphun J.: Probing knowledge in distributed data mining. In: Zhong, N., Zhou, L. (eds.) Methodologies for Knowledge Discovery and Data Mining, pp. 443–452. Springer, Berlin (1999)
[90] Tsoumakas G., Vlahavas I.: Effective stacking of distributed classifiers. In: ECAI, pp. 340–344 (2002)
[91] Lazarevic A., Obradovic Z.: Boosting algorithms for parallel and distributed learning. Distrib. Parallel Databases 11(2), 203–229 (2002)
[92] Kubota K., Nakase A., Sakai H., Oyanagi S.: Parallelization of decision tree algorithm and its performance evaluation. In: Proceedings of the the fourth international conference on high performance computing in the Asia-Pacific region, vol 2. IEEE, pp 574–579 (2000)
[93] Wolff R., Schuster A.: Association rule mining in peer-to-peer systems. IEEE Trans Syst Man Cybern B Cybern 34(6): 2426–2438 (2004)

[94] Drucker H., Cortes C., Jackel L., LeCun Y., Vapnik V.: Boosting and Other Ensemble Methods. Neural Computation 6(6) (1994)
[95] Khoussainov R., Zuo X., Kushmerick N.: Grid-enabled weka: a toolkit for machine learning on the grid. ERCIM News No. 59 (2004)
[96] Perez M., Sanchez A., Herrero P., Robles V., Pena J.: Adapting the weka data mining toolkit to a grid based environment. Lect Notes Comput Sci 3528:819–820 (2005)
[97] Sakurai Y., Yoshikawa M., Faloutsos C.: FTW: fast similarity search under the time warping distance. In Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS 2005), 326–337 (2005)
[98] Lin J., Keogh E., Wei L., Lonardi S.: Experiencing SAX: a novel symbolic representation of time series. Data Min Knowl Disc 15:107–144 (2007)
[99] Shieh J., Keogh, E.: iSAX: disk-aware mining and indexing of massive time series data sets. Data Mining and Knowledge Discovery 19 (1), 24–57 (2009)
[100] Camerra A., Palpanas T., Shieh J., Keogh E.: iSAX 2.0: Indexing and Mining One Billion Time Series, International Conference on Data Mining. (2010)
[101] Senin P., Malinchik S.: Sax-vsm: Interpretable time series classification using sax and vector space model. In Data Mining (ICDM), 2013 IEEE 13th International Conference on. IEEE, pp. 1175–1180. (2013)
[102] Oates T., Mackenzie C., Stansbury L., Aarabi B., Stein D., Hu P.: Predicting patient outcomes from a few hours of high resolution vital signs data. In Machine Learning and Applications (ICMLA), 2012 11th International Conference on, volume 2, 192–197 (2012)
[103] Kӓmpf M., Kantelhardt J.W.: Hadoop.TS: Large-Scale Time-Series Processing. International Journal of Computer Applications (0975 - 8887) Volume 74 - No. 17 (2013)
[104] EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning, D. Anicic, 2011, http://www.aifb.kit.edu/images/c/c0/Www29-anicic.pdf
[105] A SpaceTime Oriented Macroprogramming Paradigm for Push-Pull Hybrid Sensor Networking, H. Wada, 2007, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04317927
[106] Open Geospatial Consortium, http://www.opengeospatial.org/
[107] Introduction to SensorML, http://www.ogcnetwork.net/SensorML_Intro
[108] WPS-hadoop, http://gcube.wiki.gcube-system.org/gcube/index.php/Geospatial_Data_Processing
[109] Full Streaming WPS: Near real-time Geoprocessing with WPS, 2012, http://blog.52north.org/2012/10/04/full-streaming-wps-near-real-time-geoprocessing-with-wps/
[110] An Event Driven Architecture for Decision Support, T. Everding, 2011
[111] OWL-S: Semantic Markup for Web Services, 2004, http://www.w3.org/Submission/OWL-S/
[112] Web Service Modeling Ontology (WSMO), 2005, http://www.w3.org/Submission/WSMO/
[113] Semantic Annotations for WSDL and XML Schema (SAWSDL), 2007, http://www.w3.org/TR/sawsdl/
[114] hRESTS: an HTML Microformat for Describing RESTful Web Services, Jacek Kopecky, 2008
[115] WSMO-Lite Annotiations for Web Services, Tomas Vitvar, 2008
[116] WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web, 2010, http://www.w3.org/Submission/2010/SUBM-WSMO-Lite-20100823/
[117] Chiueh, Susanta Nanda Tzi-cker, and Stony Brook. "A survey on virtualization technologies." RPE Report (2005): 1-42.
[118] Bazargan, Fatma, Chan Yeob Yeun, and Mohamed Jamal Zemerly. "State-of-the-art of virtualization, its security threats and deployment models." International Journal for Information Security Research (IJISR) 2.3/4 (2012): 335-343.
[119] Pearce, Michael, Sherali Zeadally, and Ray Hunt. "Virtualization: Issues, security threats, and solutions." ACM Computing Surveys (CSUR) 45.2 (2013): 17.
[120] White, Joshua, and Adam Pilbeam. "A survey of virtualization technologies with performance testing." arXiv preprint arXiv:1010.3233 (2010).
[121] Adams, Keith, and Ole Agesen. "A comparison of software and hardware techniques for x86 virtualization." ACM Sigplan Notices 41.11 (2006): 2-13.
[122] Dueňas, Juan C., et al. "System virtualization tools for software development." Internet Computing, IEEE 13.5 (2009): 52-59.
[123] Felter, Wes, et al. "An updated performance comparison of virtual machines and Linux containers." technology 28 (2014): 32.
[124] Docker, https://www.docker.com/
[125] Luckham, D. C. and Vera, J. 1995. An event-based architecture definition language. IEEE Transactions on Software Engineering, 21, 717--734.
[126] Brenna, L., Demers, A., Gehrke, J., Hong, M., Ossher, J., Panda, B., Riedewald, M.,Thatte, M., and White, W. 2007. Cayuga: a high-performance event processing engine. In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, New York, NY, USA, 1100--1102.

[127] Wu, E., Diao, Y., and Rizvi, S. 2006. High-performance complex event processing over streams. In SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data. ACM, New York, NY, USA, 407--418.
[128] Agrawal, J., Diao, Y., Gyllstrom, D., and Immerman, N. 2008. Efficient pattern matching over event streams. In SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, New York, NY, USA, 147--160.
[129] Esper. http://www.espertech.com/, Visited July 2015.
[130] Coral8 CEP Engine. http://help.sap.com/event-stream-processor, Visited July. 2015.
[131] Oracle CEP. http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/complex-event-processing-088095.html, Visited July 2015
[132] Tibco. http://www.tibco.com/products/event-processing/complex-event-processing/streambase-complex-event-processing, Visited July 2015.
[133] Cugola, Gianpaolo, and Alessandro Margara. "Processing flows of information: From data stream to complex event processing." ACM Computing Surveys (CSUR) 44.3 (2012): 15.
[134] Heflin, J., J. Hendler, and S. Luke, "SHOE: A Blueprint for the Semantic Web", In Spinning the Semantic Web, pp. 29-63, MIT Press, Cambridge, 2003.
[135] DAML, http://www.daml.org/about.html, 2003.
[136] Horrocks, I., D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. Van Harmelen, M. Klein, S. Staab, and R. Studer, OIL: The Ontology Inference Layer, http://www.ontoknowledge.com/oil, Vrije Universiteit Amsterdam, September, 2000.
[137] RDF: Resource Description Framework , http://www.w3.org/RDF, 2006.
[138] RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/rdf-schema, 2004.
[139] OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide, 2004.
[140] Singh, M. P. and M. N. Huhns, Service-Oriented Computing: Semantics, Processes, Agents, John Willey & Sons, Ltd, England, 2005.
[141] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/, 2006.
[142] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.,*et al*.: SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission 21, 79 (2004).
[143] Hindriks, Koen, "Programming Rational Agents in GOAL", In Multiagent Programming (editors El Fallah Seghrouchni, Amal and Dix, Jürgen and Dastani, Mehdi and Bordini, Rafael H), pp.119- 157, Springer, 2009.
[144] An, Bo, "Automated Negotiation for Complex Multi-Agent Resource Allocation" (2011). Dissertations. Paper 329.
[145] Endriss, U., N. Maudet, F. Sadri, and F.Toni, " Negotiating Socially Optimal Allocations of Resources", Journal of Artificial Intelligence Research, Volume 25, pages 315-348, 2006.
[146] Klein, M., Peyman Faratin, Sayama Hiroki and Yaneer Bar-Yam, " Protocols for Negotiating Complex Contracts", IEEE Intelligent Systems, Vol. 18, pp. 32-28, IEEE Computer Society, 2003.
[147] Aydoğan, R., Koen Hindriks, and Catholijn Jonker, "Multilateral Mediated Negotiation Protocols with Feedback", In I. Marsa-Maestre, M. A. Lopez-Carmona, I. Takayuki, M. Zhang, Q. Bai and K. Fujita (editors). Novel Insights in Agent based Complex Automated Negotiation, Studies in Computational Intelligence, p. 43-59, Springer Japan, 2014.
[148] De la Hoz, E., Miguel A. Lopez-Carmona, Mark Klein, and Ivan Marsa-Maestre, "Consensus Policy Based Multi-agent Negotiation", In Proceedings of the 14th International Conference on Agents in Principle, Agents in Practice, pp. 159-179, 2011.
[149] Lang, F. and A. Fink, " Learning From the Metaheuristics: Protocols for Automated Negotiations", Group Decision and Negotiation, Vol. 24, no:2, pp. 299-332, 2015.
[150] Zheng, R., N. Chakraborty, T. Dai, and K. Sycara, "Multiagent negotiation on multiple issues with incomplete information: extended abstract", In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, pp. 1279-1280, St. Paul, USA, 2013.
[151] Aydoğan, R., D. Festen, K. V. Hindriks, and C. M. Jonker. "Alternating Offers Protocol for Multilateral Negotiation". In Fujita et. al (editors). Modern Approaches to Agent-based Complex Automated Negotiation, Springer, To be appeared.
[152] De Jonge, D. and C. Sierra, "NB3: a Multilateral Negotiation Algorithm for Large, Non-linear Agreement Spaces with Limited Time", Autonomous Agents and Multi-Agent Systems, Vol. 29, pp. 896-942, 2015.
[153] Dynamic M2M Event Processing, Oleg Kostukovsky Walt Bowers, EclipseCon 2014 https://www.eclipsecon.org/na2014/sites/eclipsecon.org.na2014/files/slides/EclipseCon2014-DynamicEventProcessing_draftV2.pdf
[154] A Survey of the State-of-the-art in Event Processing, Otávio M. de Carvalho, Eduardo Roloff, Philippe O. A. Navaux 11th Workshop on Parallel and Distributed Processing (WSPPD), 2013
[155] Intelligent M2M: Complex event processing for machine-to-machine communication, Ralf Bruns, Jürgen Dunkel , Henrik Masbruch , Sebastian Stipkovic, Expert Systems with Applications 02/2015; 42(3):1235–1246
[156] IBM CPLEX Optimizer, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html. Visited July 2015
[157] Xpress-MP Documentation, http://www.ic.unicamp.br/~lee/mc548/trabalho/xpress/docs/auxiliary/content/cd-manuals.html. Visited July 2015

[158] M. Barcelo, J. Llorca, A. Tulino, R. Narayan, "The Cloud Service Distribution Problem in Distributed Networks", IEEE International Conference on Communications (ICC), June 2015. DOI: 10.1109/ICC.2015.7248345
[159] D. Bertsekas and R. Gallager. Data Networks, 1987. Prentice-Hall.
[160] M. Allalouf and Y. Shavitt. Centralized and distributed algorithms for routing and weighted max-min fair bandwidth allocation. IEEE/ACM Transaction on Networks, 2008.
[161] Bard, J.F., Practical Bilevel Optimization: Applications and Algorithms, Kluwer Academic Press, 1998.

5 State-of-the-art analysis on M2M Communication Technologies

5.1 General

Communication technologies provide means to exchange information between two physically separated devices, which may be distributed sensors attached in the edges of the network and which may have connection via cellular networks to backbone networks. The referred communication includes bi-directional traffic for sensor data collection, control message delivery and device management. A few common traits can be derived for the application:

- Huge amount of distributed sensors need to be connected.
- Information exchange is to be implemented among heterogeneous nodes and systems.
- Variable data rates are required by different application needs.
- High reliability of the communication should be guaranteed.
- Latency should be minimized in order to support real time operation.

Different air interfaces can be adopted to provide such communication media in the edges of the networks. There are at least two different levels for the wireless communication: cellular and local radio connectivity. A cellular network, also called as mobile network, is a wireless Wide Area Network (WAN) network distributed over land areas called cells, each served by at least one fixed-location transceiver, known as a base station. The history of cellular networks can be traced back to early 80's, when the first generation analog cellular network was widely deployed in North America. Over almost half of a century, nowadays the coverage of cellular networks is nearly everywhere, which provides reliable connection over relatively long distance. For local radio connectivity, Wi-Fi is the most well-known standard, which belongs to Wireless Local Area Network (WLAN). Wi-Fi provides very high data rate connection over small area. Recently, more and more new standards are emerged to provide the last mile radio connectivity, such as Bluetooth and ZigBee.

5.2 An Overview on Communication technologies

A set of M2M communication technologies is reviewed in Table 1 [9]. The first part of the table gives a view to the technologies that are independent of the physical networking and radio access technologies, and represent examples of communication overlay technologies. Basically, the communication overlay may be able to logically connect the M2M asset devices, M2M gateways, infrastructure servers and user with each other to hide the heterogeneity and dynamicity of physical networks and solve such problems as e.g. mobility, dynamic presence, distribution, device power saving features, security including firewalls and NAT restricted networks. The communication overlay is a kind of logical network operating on top of physical networks like, e.g., the Internet. Overlay networking can be based on the virtual communication layer, which is built on top of other transport media and/or physical network [25]. Overlay networking has been previously applied to improve the robustness and availability of Internet paths between hosts (e.g., MIT RON), enable smooth transition to the improved technology (e.g., 6Bone) or to reduce network load by peer assisted data delivery (e.g., BitTorrent). Overlays have been used to route messages and connect different entities (e.g., SIP and XMPP), and/or also to implement data forwarding and dissemination (e.g., Chord, Tapestry, and Pastry) [26, 27]. Another motivation for overlay networking arises from security challenges. For example, Virtual Private Ad Hoc Networking (VPAN) has been developed to create virtual overlay networks between trusted IP capable devices [28]. There are also specific types technologies of communication overlays developed for opportunistic [29], and dynamic mobile ad hoc networking [30].

The second part of the Table 1 list technologies that are dependent on the physical communication channels i.e. radio access or wireline access technologies. Several standardized short range radio technologies such as Bluetooth, ZigBee, RFID, *etc.*, can be utilized locally, but also as vendor specific and optimized radio technologies for WSNs. The wide area and Internet connectivity can be provided by, e.g., Telecom operator and/or ISP using, e.g., Ethernet, DSL, GSM, 3GPP, WiFi, WiMax, *etc.* In this deliverable, the physical communication technologies are further categorized to cellular network operating in licenced bands and radio access technologies operating in unlicenced bands. It is expected that Internet (i.e. IPv4/IPv6 based networks) establishes the main networking infrastructure between the communication overlays and physical communication channels.

Reliable and delay aware transfer of messages over the network is very important for M2M applications. The hypertext transfer protocol (HTTP) is usually applied to transfer messages in client-server manner in World Wide Web, e.g. HTTP/REST. In addition, there are protocols such as SMTP, POP and IMAP for electronic mail systems. In addition, protocols like XMPP and SIP/SIMPLE have enabled capabilities such as instant messaging and presence to be applied in more real-time communication. XMPP uses decentralized client-server architecture to keep clients simple, and pushes most of the complexity into the servers [10]. The architecture is different than WWW in the sense that it supports inter-domain connections called federations. In addition, email network uses multiple hops between servers to deliver messages but the XMPP architecture uses direct connections. When taking the communication requirements generally set by M2M applications into concern, it is seen that XMPP type of architecture has succeeded to enable real-time messaging with support for simple clients.

The websocket protocol (a part of HTML5 initiative) which relies on HTTP for handshake and negotiation, is message-based and has been designed to allow bidirectional communications. The related message queues protocols can be broker-based (e.g., DDS, AMQP, and STOMP) or broker less (e.g., ZeroMQ) and allow asynchronous communications and operate at the same level as HTTP. MQTT is a message queue designed with M2M applications in mind to enable lightweight publish/subscribe messaging transport.

Table 10 Review of the M2M Communication technologies.

Technology	Forum(s), References	Main Contribution
Sensor-Over-XMPP	XMPP Extension (protoXEP) [11]	a payload format for communicating sensor and actuation information
Data Distribution Service for Real-Time Systems (DDS)	Object Management Group (OMG)	Scalable, real-time
Advanced Message Queuing Protocol (AMQP)	OASIS AMQP standard	Broker-based messaging, publish-subscribe

STOMP	STOMP	Simple broker-based text-based protocol
MQ Telemetry Transport (MQTT)	MQTT Eclipse M2M Industry Working Group	Lightweight publish/subscribe binary messaging protocol
ZeroMQ	ZeroMQ protocol [12]	An openly published simple and lightweight publish-subscribe type of messaging protocol designed for constrained devices and low-bandwidth, high-latency or unreliable networks.
COAP usage for REsource LOcation And Discovery (RELOAD)	Proposed as IETF Internet-draft	peer-to-peer federation of geographically distributed WSN islands
Content-Centric Network	CCNx	Content-based networking. Security by design.
Websocket protocol	IETF RFC 6455	two-way communications with Low-overhead transport (single TCP connection)
CoAP	IETF's Constrained RESTful environments (CoRE) working group	An application layer protocol designed for constrained devices allowing them to communicate over the Internet [13-17].

Table 11 Review of communication technologies

		Wired	Radio	Full IP
Cellular	3GPP	No	Yes	Yes
Bluetooth	Bluetooth Special Interest Group	No	Yes	No
Bluetooth LE	Bluetooth Special Interest Group	No	Yes	No
Ant/Ant+				
ZigBee	ZigBee Alliance (and IEEE)	No	Yes	No
ZigBee IP	ZigBee Alliance (and IEEE and IETF)	No	Yes	Yes
6LowPan	IETF	No	Yes	Yes
Z-Wave	Z-Wave alliance	No	Yes	No
Wi-Fi low power	IEEE	No	Yes	Yes
Modbus	Modicon (society)	Yes	No	No
BacNet	ASHRAE	Yes	No	No
LonWorks	ANSI	Yes	No	No
KNX	ISO/IEC	Yes	Yes	No

One challenge is related to application of HTTP within constrained local M2M asset network. To solve this problem, IETF CoRE (Constrained RESTful Environments) working group has specified Constrained Application Protocol (CoAP) standard with the goal of supporting REST-like applications in constrained environments. The second challenge is related to the sizes of IP packets and headers. To solve this problem, IETF has created the 6lowpan (IPv6 Low Power wireless Area Networks), which describes an adaptation layer between IPv6 and a layer 2 protocol, such as (but not limited to) IEEE 802.15.4, to handle MTU sizes and compress IPv6 headers from 60 bytes to 7 bytes. There are also other challenges arising from heterogeneity of M2M devices and local M2M asset networks, and coding and integration of M2M application content.

5.3 Communication Overlay

This section introduces the challenge related to the communication with constrained embedded devices. In addition, a couple communication overlay technologies are shortly overviewed. However, it should be taken into concern by the reader, that there are also many other potential communication overlay related technologies available, and the aim is not to describe all of them here, for more information see [25-30].

5.3.1 Communication with constrained embedded devices

A Low Power and Lossy Network (LLN) is a network between constrained embedded devices, in which the communication links may change frequently, or even disappear. In order to face the challenges in those networks, the IETF has set working groups to elaborate standards which guarantee that such an infrastructure is scalable, secure and reliable in terms of communications between each device. One of the most important aspects of these standardization activities is the use of the IP layer as the reference layer. Using this reference, new protocols have been developed for routing, transport, and applications in LLNs. IPv6 brings some outstanding benefits such as an addressing scheme which allows identifying billions of devices and supporting point-to-point communications between a device and a PC connected to Internet. However, the IPv6 protocol is inadequate for LLNs in terms of network overhead. As a result, the IETF 6LowPAN WG [13] proposes adaptations of the IPv6 protocol when the underlying network is constrained. For example, standards have been proposed for the transmission of compressed IPv6 packets over IEEE 802.15.4 networks [14]. IPv6 and 6LowPAN network stacks are natively available on common operating systems for embedded devices (e.g., Contiki and TinyOS), therefore making them able to communicate with both Internet and LLNs devices.

Another aspect or LLNs is the strong constraints on routing protocols, which must be different from those used in traditional IP networks. First of all, link conditions may change frequently during time; therefore a routing protocol must react quickly to these changes. Second, the nodes have really strong storage constraints; therefore, a routing protocol should work even if a node has not stored all the routes to each of the other nodes in the network. Third, since the nodes have severe energy constraints, the exchange of control messages should be kept as low as possible.

One solution to the above-mentioned LLNs limitations is provided by the RPL routing protocol. It has been developed to have really limited control traffic, to fit harsh and constrained environments, with limited data rate and potentially elevated error rate. RPL is a distance-vector protocol based on the creation of a routing tree, referred to as Destination Oriented Acyclic Directed Graph (DODAG), where the cost of each path is evaluated according to the metrics defined in an objective function. The goal of this protocol is the creation of a collection tree protocol, as well as a point-to-multipoint network from the root of the network to the devices inside the LLN.

In order to keep the status of the network updated, the root of the RPL tree sends periodical messages, referred to as DODAG Information Object (DIO). The receiving nodes may relay this message or just consume it, if configured as leaves of the tree. The RPL protocol also introduces a trickle mechanism which allows reducing the transmission frequency of DIO messages according to the stability of the network. In addition, RPL offers several advanced functionalities, the detection of loops in the routes, and the management of local faults (via local or global repair).

5.3.2 Constrained Application Protocol (CoAP)

The IETF CoRE (Constrained RESTful Environments) working group [15] has defined the Constrained Application Protocol (CoAP) standard with the goal of supporting REST-like applications inside constrained environments like those identified by the RoLL [16] and 6lowPAN working groups. Application domains include LLNs and more generally M2M communications, and span over a large range of business use cases such as smart energy or building automation. The specification defines a binary message structure between CoAP endpoints as well as the interaction protocol. By following REST architectural principles [71], CoAP exposes a representation of the information available on a constrained device as a set of identifiable resources. This way, any CoAP endpoint may interact with it remotely using the interaction methods used by the HTTP protocol: GET, POST, PUT, and DELETE.

In order to make the resources discoverable, the CoAP protocol standard advises to expose CoAP endpoint's resource metadata using the CoRE Link Format [17] at a specific URI. CoAP messages rely on the UDP transport protocol between endpoints. This is to accommodate the potentially unreliable and lossy wireless environments that render the TCP protocol inefficient in terms of network resource usage. In order to meet eventual QoS requirements, since UDP is natively unreliable, CoAP has introduced the use of confirmation messages, which correspond to an acknowledgement that a CoAP message has been received.

Collection of data from a CoAP-enabled device is achieved by sending a CoAP request message (GET method) to the CoAP server hosted on the device: as soon as the CoAP server receives such a request, it replies with a CoAP response with data requested by the CoAP client or notifies that the response will be sent in a separate response. Another interaction scheme supported by the CoAP protocol is the publish/subscribe paradigm. Instead of sending periodical requests to a CoAP server to be kept updated on the status of a resource, the CoAP client may subscribe, through specific exposed end-points, to a CoAP server, which will be in charge of periodical updating all the subscribed clients of the status of a given resource.

RESTful architectures make caching of the data possible within the network. Caching is supported by CoAP and makes it possible to optimize the data delivery over potentially constrained wireless links. For each CoAP observed value a lifetime is defined; if two consecutive requests are received by a CoAP server or proxy in a period of time smaller than that defined by the lifetime parameter, the former request will be sent querying the resource, whereas the latter will be served using the cached value. Using caching, some optimizations can be easily foreseeable for M2M communications. By serving fresh information from a cache instead of querying the endpoint itself, one could experience a shorter delay or a better QoS on a particular request. Also, caching may help reduce the overall consumption of an energy-constrained network by reducing the number of wireless transmissions required for collecting data.

A low-power version of CoAP has been implemented for Contiki [18]. The implementation leverages the ContikiMAC low-power duty cycling mechanism to provide power efficiency. Based on the results of the CoAP request/response, cycles are most energy-efficient when each message fits into a single 802.15.4

frame. When messages are bigger than frames, the interoperation of information models, data encoding/decoding, and segmentation/reassembly with constrained M2M capillary networks and M2M asset devices need to be carefully considered. HC (HTTP/CoAP) proxies provide the interworking functionalities for application spanning across LLNs (potentially running CoAP/UDP/IPv6/IEEE 802.15.4 protocol stack) and the Internet (HTTP/TCP/IPv6).

CoAP base specifications identify DTLS [19] and IPsec [20] as mechanisms to offer data origin authentication, integrity and replay protection, and encryption for the CoAP messages. In addition to these, an alternative [21] to IPsec and DTLS has been presented.

5.3.3 Extensible Messaging and Presence Protocol (XMPP)

XMPP has been developed to enable message oriented communication services applicable in the Internet context on top of TCP/IP. The XMPP communication architecture is based on distributed client-server model; however, also client-to-client (peer to peer) communication is enabled. The core services of XMPP includes support for presence information, secure messaging (TLS), overlay communication over IP, near real-time messaging, authentication, contact list management, and service discovery. Each XMPP client has an account hosted by a XMPP server, and the client can be addresses by unique Jabber ID (JID). XMPP JID contains three parts: user, domain and resource as shown in Table 3, RFC 6122/RFC3921 [22]. In XMPP, network domain-part of JID must be a fully qualified domain name or IP address. Each domain presents one logical groups with one user account database. Each domain may present own user account policies. The device owner is usually considered to be also a user of the M2M domain and an owner of at least one XMPP user id. All devices share same user id (local-part and domain-part) with their owner. Separation of devices is done by examining the resource part of JID.

Table 12 An example of Extensible Messaging and Presence Protocol (XMPP) Jabber ID.

JID: Matt@home.com/heating-regulator		
Local-part	Domain-part	Resource
Matt	home.com	heating-regulator

A client connects to the server to send and receive messages, the servers routes messages to the others enabling client-to-client communications over multiple servers, Figure 5.

Discovery can be done directly using a domain part as a server address or discovering server address with SRV lookup from DNS. All clients connect to only their own server specified by JID. Connections are persistent XML streams over TCP and optionally encrypted by Transport Layer Security (TLS) layer. Encryption of TCP stream is strongly recommended, but not required. An administrator of a domain may specify that encryption is mandatory and it is up to administrator or designer to choose whether TLS certificates should be checked. Most client libraries accepts self-signed certificates, this should be taken into account when considering security aspects of client-to-server connections. The availability of each client can be detected with the aid of presence messages. Presence information is shared only with XMPP users that are in the roster/address book of the client sending the presence information.

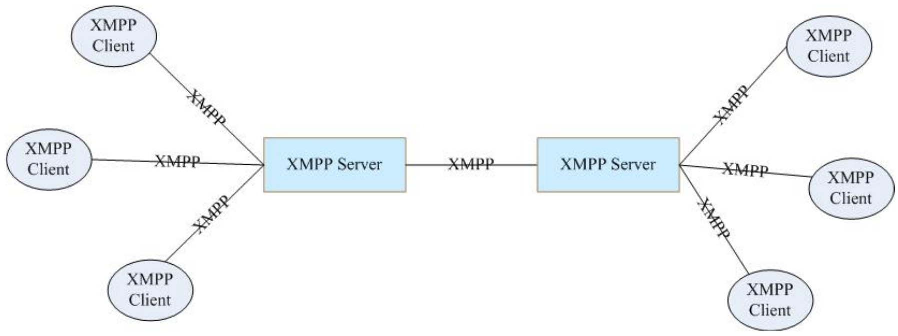


Figure 40 XMPP architecture

Server to server connection is an XML stream over a TCP connection, similar to client to server connections. The most important difference is that server to server connections are not authenticated because they happen in between different domains that do not share a common user database. XMPP servers may use XMPP Dial back (XEP-0220), to verify the domain of the connecting server.

The domain administrator may require stronger identification verification by using TLS certificates and Simple Authentication and Security Layer (SASL). When M2M clients located in different domains would like to exchange messages, routing of messages will be done by the domain specific servers. Then, a communication link between servers of the domains is negotiated to enable messaging between the referred M2M clients.

The capability for handing presence information has been developed for XMPP. The availability of each client can be detected with the aid of presence messages. XMPP offers mechanisms to select how, when and to whom presence information is shared to, as described in requirements listed in [23]. For example, presence can be shared only with contacts that are in roster/address book of client sending the presence information. The related presence information (JIDs) is critical information in the system and it is stored in the roster database. Another useful feature is the publish-subscribe model (XEP0060), which can be used by XMPP entities to subscribe information of the presence of other entities, and receive notifications accordingly.

5.4 Cellular network

The 2G/3G networks such as the GSM and WCDMA were originally designed for circuit switched voice (and data) applications with limited emphasis on scalable data services. However, new applications such as SMS text messaging were rapidly developed which employed low data rate techniques. Moreover, growing data communications needs resulted in 3G packet data technology HSPA, in which the focus was in high data rates. The target of the 4G LTE system was to create all-IP based packet data network with high data throughput. The emphasis was in spectrum efficiency and high data rates. In 2010's the need for M2M communications has become obvious which has set new data network requirements, for example, what comes to data scalability, sensor network connectivity and energy efficiency [see e.g. ETSI 3GPP specification TS 33.812]. Specific use cases demand for low-power wide area solutions for extensive number user devices.

The natural role of cellular networks is to provide a global wireless backbone for diverse M2M systems (see Figure 6). This is facilitated by existing worldwide coverage and the fact that large sub 1 GHz frequency bands have been recently allocated to cellular services, e.g. the 700- 900 MHz bands which suit well for large operation range.

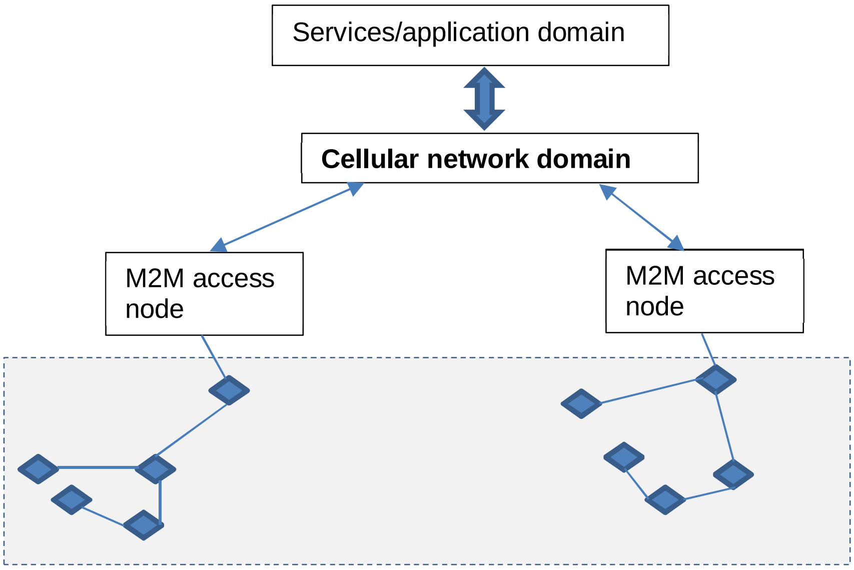


Figure 41 Example: M2M device domain with cellular backbone

5.4.1 2G GSM/GPRS/EDGE

In energy network control applications such as meter reading and SCADA (Supervisory Control and Data Acquisition) functionalities, 2G (GPRS) is currently used for surveillance, remote protection and maintenance. GPRS or similar wireless connection services are offered by commercial tele operators but 2G wireless systems are suffering from end-to-end latency and limited capacity. Moreover, the 2G systems cannot offer sufficient operational reliability during emergency situations such as big storms. Currently, connections are based on packet switched IP-technology which deploy e.g. VPN connectivity. An exemplary energy network application is depicted in Figure 7.

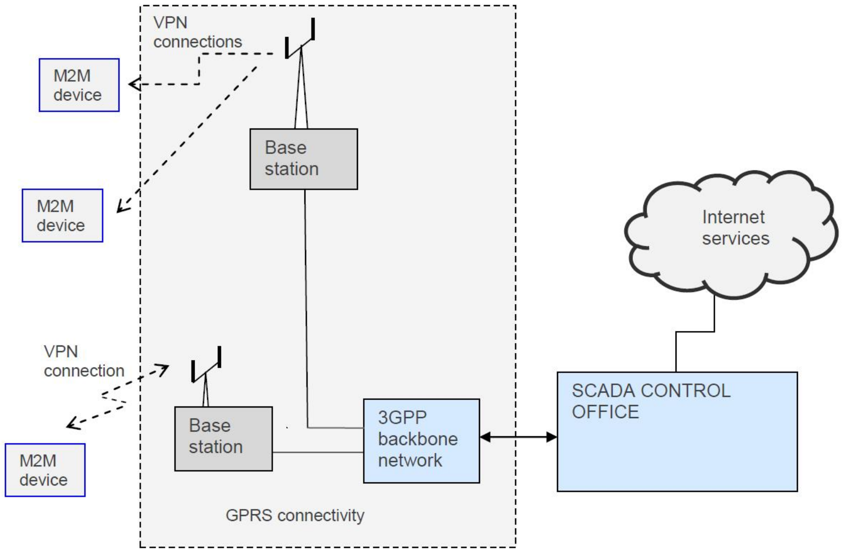


Figure 42 Principle of GPRS based energy network for M2M SCADA application

5.4.2 3G WCDMA/HSPA

Packet switched 3G HSPA with similar network architecture as in **Error! Reference source not found.** applies well to M2M applications. It offers much larger data capacity than GPRS and suits e.g. for video surveillance. However, it suffers from relatively large latency and inefficiency for very small data packet connectivity in sensor networks. This challenge is an in-built feature of 3G network which is due to the complicated control plane architecture.

5.4.3 4G LTE/ LTE Advanced

ETSI 3GPP has designed the 4G LTE mobile communications standard as an evolution of 3G WCDMA/HSPA. In comparison to previous 3GPP standards, LTE has been designed solely for packet-switched IP data transfer in a way that the system control plane is as simple as possible (see Figure 43 Principle of 4G LTE network architecture

6). This enables all-IP network structure with end-to-end delay of less than 50 ms. Time-critical system control is performed by the eNodeB. Even though the LTE supports advanced network topologies including pico- and femtocells and has advanced resource allocation and packet scheduling functionalities it is not flexible enough for versatile M2M applications. The design target has been as high spectrum efficiency and system capacity in a mobile data communication environment. One additional reasoning to the M2M challenge is the fact that LTE user equipments (UE) are rather complex and costly for many M2M applications. Therefore, in on-going LTE evolution (standardization of LTE-Advanced, see ETSI 3GPP technical report TR36.913) there is a specific standardization effort on M2M challenges [TS 33.812].

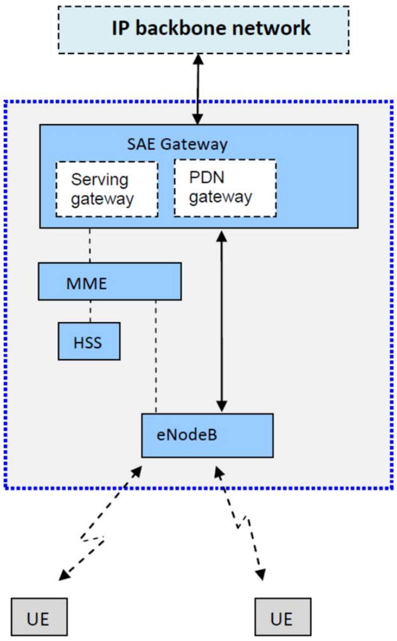


Figure 43 Principle of 4G LTE network architecture

As a result of these activities, 3GPP announced in July 2015 that its study report TR36.888 has set the minimum requirements for low-cost M2M LTE user devices. These requirements include e.g. the following:

- 2G data rates.
- Improved spectrum efficiency for low data rate M2M (vs. 2G)
- Equal or better service coverage than 2G
- Radio frequency coexistence of low-cost M2M UEs with legacy LTE UEs
- Re-use the existing LTE/SAE network architecture and hardware
- Support limited mobility and low power consumption modules

Transferring M2M traffic from 2G to LTE networks requires a new UE category that is specified for low data rates and delay tolerant applications. The design goals for M2M specific, long range terminals are defines as follows [24]:

- Very low device cost
- Support large number of M2M devices
- 10+ years battery life with two AA batteries
- 15-20 dB link budget (coverage) enhancements.

5.4.4 5G Prospects

One important topic in European Community 5G project METIS was M2M wireless communications [https://www.metis2020.com/]. One of the five main scenarios was named “Ubiquitous things communicating” in which the main challenges were listed as very low energy, very low cost, and a massive number of devices. Accordingly, the assumption is that a majority of the connected M2M devices will most likely be simple, such as sensors and actuators. In addition, advanced and service critical devices will become connected such those of smart electrical grids, industrial and medical equipments. It is foreseen that widely varying requirements in several domains e.g. in terms of energy consumption, cost (complexity), transmission power, and latency, cannot always be met by today's cellular networks. For example, ubiquitous devices will sometimes communicate in a local context.

Real challenge is to design a unified communications network for applications combining information from different types of data sources at the same time managing the large control overhead by the high number of devices.

Second M2M application scenario was identified as "Super real-time and reliable communications". This covers M2M communication with real-time constraints, enabling new functionalities for

- traffic safety and efficiency
- smart energy grid
- e-health
- efficient industrial communications.

These new applications are expected to require much higher reliability and lower latency than today's communication systems. As an example for certain automotive use cases, a maximum end-to-end latency must be guaranteed with very high reliability, e.g. 99.999%. Challenging M2M use cases are listed as specific test cases which are analyzed in more detail. These test cases include:

- Teleprotection in smart grid energy networks
- Emergency communications
- Real-time remote computing for mobile terminals
- Traffic efficiency and safety

Prospected air interface technologies are reviewed in document ICT-317669-METIS/D2.4 [https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D2.4_v1.pdf]. In 5G air interface design three main topics are studied: 1) Flexible air interface, 2) Waveforms, coding & modulation and transceiver design and 3) Multiple accesses, Medium Access Control (MAC) and Radio Resource Management (RRM). Key 5G technology enablers are identified as an air interface for ultra-dense networks (UDN) and for moving networks; multi-carrier transmission schemes with filtering; and novel access schemes for massive M2M access as well as for non-orthogonal access.

5.5 Access technologies

5.5.1 IEEE802.3 (Ethernet)

Ethernet is the most popular LAN technology in the world. It is an easy, relatively inexpensive way to provide high-performance networking to all different types of computer equipment.

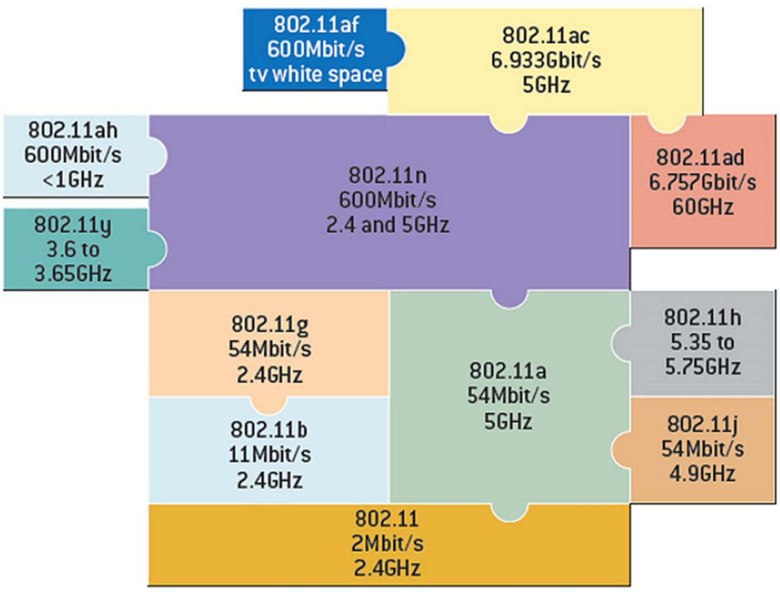
Ethernet was invented at Xerox PARC and developed jointly by Digital Equipment Corporation, Intel and Xerox. Introduced in 1980, Ethernet was distinguished by its high speed (10 Mbps), its unusual signaling methodology (the latest version of which is now referred to as Carrier Sense Multiple Access with Collision Detection or CSMA/CD), and by the physical medium on which it ran: a thick, high-quality coaxial cable with a bright yellow braided sheath.

Today, the term Ethernet refers to a whole family of closely related protocols characterized by their raw data rates (10 Mbps, 100 Mbps, 1 Gbps or 10 Gbps) and the physical medium on which they operate. Ethernet now runs on a wide variety of physical media. Among the most common are: coaxial cable (thick or thin), many types of copper cable called twisted pair, and several types of fiber-optic cables using a variety of signalling methods and light wavelengths.

The CSMA/CD approach is used by any form of Ethernet operating in half-duplex mode-that is, the mode in which transmit (Tx) and receive (Rx) signals can be sent on the same wire or data path. In full-duplex mode, transmit and receive signals are separated onto dedicated, one-way channels. This eliminates the need for CSMA/CD, as all the transmissions on a single data path will be coming from a single device. Half-duplex mode is seldom used in versions of Ethernet running on fiber, and is not supported at all in the 10 Gbps standards.

5.5.2 IEEE802.11 (Wi-Fi)

IEEE802.11 is a group of standards that define media access control (MAC) and physical layer (PHY) specifications for wireless local area network (WLAN). Figure 44 gives an overview of the operating bands and typical data rates of individual standard in the IEEE802.11 family. Nowadays IEEE802.11 based network becomes the dominant technology for indoor wireless communication, including access points (APs) and stations (STAs) such as laptops, tablets, PDAs, printers using a WLAN interface. Operating at free-licensed 2.4GHz/5GHz ISM frequency bands, it becomes universal available for users to set up convenient Internet access.



Year	Standards
1999	802.11a
	802.11b
2003	802.11g
2009	802.11n
2012	802.11ad
2014	802.11ac
	802.11af
2016	802.11ah

Figure 44 Overview of IEEE802.11 standard family

Wi-Fi provides very high speed connection, but it falls with short communication distance. In order to support even larger coverage, sub-GHz frequency bands becomes a nice candidate, thanks to the better propagation characteristics as compared to 2.4GHz frequency bands. A new standardization activity started in the IEEE802.11 family, the IEEE802.11ah **Error! Reference source not found.**, or sub-GHz Wi-Fi. The aim of this new standard is to support Internet-of-Things (IoT) or M2M communication with its mandatory low to medium data rate modes, and to support extended Wi-Fi range for cellular traffic off-loading with its optional high data rate modes. It emphasizes on ultra-low power radio with multiple years of battery usage. The expected transmission range is up to 1 km with at least 150 kbps. The timeline of IEEE802.11ah standardization is shown in Figure 38. The standard will be finalized in the March of 2016.

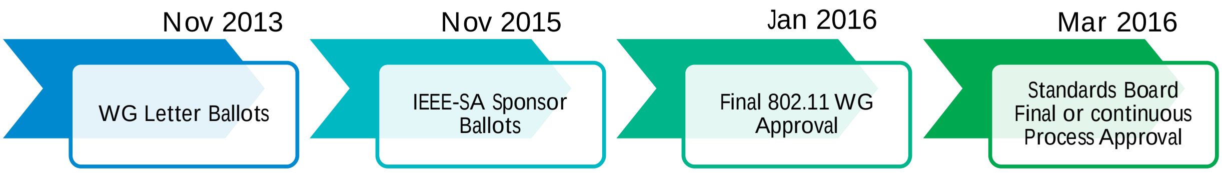


Figure 38 Timeline of IEEE802.11ah standardization

The PHY layer parameters of IEEE802.11ah is almost directly 10 times scaling down from the high performance IEEE802.11ac. On the other word, the focus of this sub-GHz Wi-Fi shifts from traditional design spec of a Wi-Fi system, such as high data rate, to new features needed by a M2M communication network, low power consumption and large coverage.

Given the technical features as aforementioned, the potential use cases of IEEE802.11ah can be summarized as:

- **Sensor Networks** – used as the communication medium for the transmission of short-burst data messages from sensors, which include smart metering;
- **Backhaul networks for sensors** – used to create the backhaul of mesh networks created by IEEE 802.15.4 networks;
- **Extended Wi-Fi range for cellular traffic off-loading** – used to off-load traffic from a cellular network. The caveat is that the performance should be at least comparable with the one from the cellular network;
- **M2M communications** – Whereas current systems are optimized more for human-to-human (H2H) communications, IEEE 802.11ah standard will mainly consider sensing applications.
- **Rural communication** – Wireless communication in rural areas has led to some effort that is also titled as bridging the digital divide. Large potential is given by sub 1 GHz due to the wider supported range.

5.5.3 Bluetooth low energy (BLE)

In June 2010, Bluetooth® SIG announced the formal adoption of Bluetooth Core Specification version 4.0 **Error! Reference source not found.**, and unveiled an entirely new protocol stack for rapid build-up of simple links, the Bluetooth low energy (BLE) standard. Since 2011, BLE is also branded as Bluetooth SMART. Catering to the extended usage of wireless communication on smart phones and tablets, quite a few attractive features are incorporated in the BLE standard:

- Ultra-low peak, average and idle mode power consumption

- Ability to run for years on standard coin-cell batteries
- Low cost
- Multi-vendor interoperability
- Extended communication range

As compared to Bluetooth 2.1 + EDR and Bluetooth 3.0 + HS, these new features are reflected on simplified link control layer and packet structure. In a BLE system, star topology is supported with limited number of nodes, which is mainly catered for body and peripheral networks. The PHY data rate supported in a BLE system is 1Mbps at 2.4GHz frequency bands.

In December of 2013, an updated version Bluetooth 4.1 was published. The new features in Bluetooth 4.1 include:

- Coexistence with cellular
  Automatic coordination with cellular network, such as LTE.
- Allow master and slave modes simultaneously
- Better connections
  Keep reconnection time interval flexible and variable, leaving manufacturers with more control over Bluetooth connections.
- Bulk data transfer
  Sensors can collect data during a run, bike ride or swim, and at a later moment transfer bulk data more efficiently.
- IP connectivity
- Add dedicated channel for IPv6 based communication.

In the upcoming BLE5.0, mesh network capability is to be included, which evolves from CSRmesh. CSRmesh is first announced in Feb, 2014. Initiated with lighting control, CSR mesh is intended for home automation and IoT. An illustration of the CSRmesh is shown in Figure 46. The key features of CSRmesh is to make use of three advertising channels in BLE to set up flood mesh network, which is easy for setup. Power consumption is controlled by duty cycling the radio. As compared to other non-flood routing in setting up a multihop network, such as a ZigBee network, the signaling overhead is much lower. Two layers of security are offered by CRSmesh, encryption (AES-128) and additional network passcode.

Figure 46 Illustration of CSR mesh

In addition to the support to mesh network, other new features to be supported in the upcoming BLE5.0 include:

- Extended range
  - Expect receiver sensitivity to be improved by more than 10dB with spreading and channel coding.
- Direction finding
  - Include Angle-of-Arrival (AoA) and Angle-of-Departure (AoD) capabilities for indoor positioning and more accurate asset tracking.
- Higher data rate
  - 2Mbps mode is to be supported.
- Isochronous channels
  - Hearing aids and broadcasting audio use cases
- Extension of advertising packets size

Besides the prospects of technology enhancement in the next generation of BLE system, the biggest advantage of BLE over other connectivity standards is that Bluetooth (including BLE) already becomes a common peripheral interface of a smart phone. Therefore, the BLE network can always be used seamlessly with the cellular network. In that sense, it greatly extended the accessibility of the system.

5.5.4 IEEE802.15.4 (ZigBee)

IEEE 802.15.4 defines PHY and MAC layer for low-rate wireless personal area networks (LR-WPANs). Mandatory modes of narrow band PHY are operating at 868/915MHz and 2.45 GHz bands. The original IEEE802.15.4a UWB PHY becomes an optional PHY mode in IEEE802.15.4 2011 version. Low to medium data rates are supported, which are 250 kbps @2.45GHz, 40 Kbps @ 915MHz and 20 kbps @868MHz. Multiple topologies are defined by the standard, star, peer-to-peer, and cluster tree.

Adaptive superframe structure provides great flexibility for network setup. CSMA-CA based channel access yields high throughput and low latency for low duty cycle devices. Optional contention-free guaranteed time slot allows for periodic traffic or applications requiring low latency. Addressing space up to: 2 bytes for PAN ID + 8 bytes for address. In that sense, IEEE802.15.4 has the capability to set up large scale sensor network, and provides freedom to customize the network according to the application scenarios.

To fit the IEEE802.15.4 or LR-WPAN to the large scale M2M grid, IP networking is the bridge. 6LoWPAN, an acronym of IPv6 over Low power Wireless Personal Area Networks, defines encapsulation and header compression mechanisms that allow IPv6 packets to be sent to and received from over IEEE 802.15.4 based networks. 6LoWPAN is a concluded group in IETF. It enables IP-capability for the 15.4 nodes, and provides interoperability between 15.4 nodes and existing IP devices.

Sometimes people try to interchange the term of IEEE802.15.4 with ZigBee. ZigBee is based on IEEE802.15.4 (PHY and MAC), but defines a suite of high-level (from network layer and above) communication protocols used to create personal area networks. Figure 47 shows the difference between IEEE802.15.4 and ZigBee. ZigBee Pro, or ZigBee 2007, is the 3rd generation of ZigBee standards, preceded by ZigBee 2004 and ZigBee 2006. As compared to IEEE802.15.4, generic mesh networking is supported in ZigBee standard. New features in ZigBee Pro include:

- Stochastic addressing: 16-bit network address is randomly selected. If nodes choose the same address, ambiguity is solved using the 64-bit IEEE 802.15.4 MAC address.
- Many-to-one source routing
- Fragmentation
- Dynamic channel selection
- Asymmetric connections
- Security: in addition to AES-128b and a global network key, ZigBee Pro allows each couple of nodes to have their own key, and thus p2p encryption.

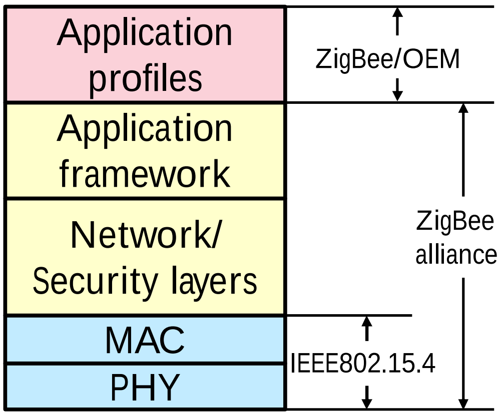


Figure 47 IEEE802.15.4 vs. ZigBee

Aiming to connect to the IP network, ZigBee IP was specifically designed for ZigBee Smart Energy protocol version 2 (SEP2), which is elected by the United States National Institute of Standards and Technology (NIST) as a standard profile for smart energy management in home devices.

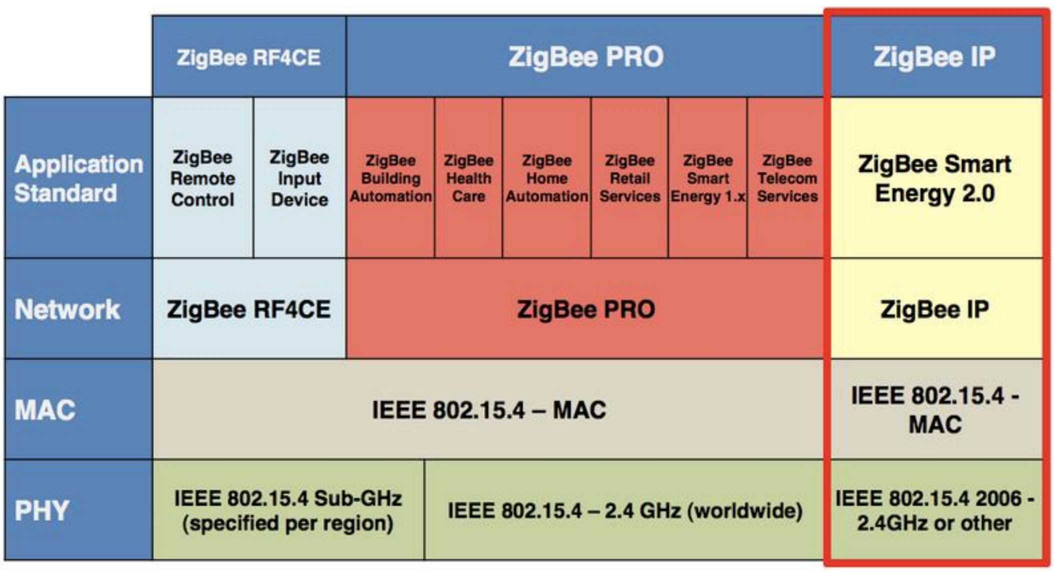


Figure 48 ZigBee IP Error! Reference source not found.

5.5.5 IEEE802.15.4g (SUN)

Smart metering utility network (SUN) enables multiple applications to operate over shared network resources, providing monitoring and control of a utility system. The application scenarios include:

- Gas Metering
- Demand/Response
- Distribution Automation
- Machine to Machine (M2M)
- Wireless Sensor Networks applications

SUNs are required to cover geographically widespread areas containing a large number of outdoor devices. IEEE 802.15.4g is PHY amendment to IEEE 802.15.4 to cater for such smart metering utility network. The update in PHY, as compared to IEEE 802.15.4, includes:

- Three PHY modes: Multi-Rate FSK (MR-FSK), MR-OFDM, MR-O-QPSK
- Common signaling mode to facilitate multi-PHY management
- Operating at sub-GHz, different physical layer interfaces are expected over larger coverage. The common signaling mode allows the interoperability of different PHY interfaces.
- Variable data rates and bandwidths
- Configurable preamble length

Better co-existence support is provided in MAC layer, as compared to IEEE 802.15.4.

References

[1] https://en.wikipedia.org/wiki/IEEE_802.11ah

[2] http://www.ieee802.org/15/pub/TG4.html

[3] https://en.wikipedia.org/wiki/6LoWPAN

[4] http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeeip/

[5] http://greentechadvocates.com/2013/04/04/zigbee-ip-smart-grid-meet-the-internet-of-things/

[6] Bluetooth core specifications V4.0 2010, https://www.bluetooth.org/en-us/specification/adopted-specifications

[7] http://www.bluetooth.com/Pages/low-energy-tech-info.aspx

[8] https://wiki.csr.com/wiki/CSRmesh

[9] Latvakoski, J.; Iivari, A.; Vitic, P.; Jubeh, B.; Alaya, M.B.; Monteil, T.; Lopez, Y.; Talavera, G.; Gonzalez, J.; Granqvist, N.; Kellil, M.; Ganem, H.; Väisänen, T. A Survey on M2M Service Networks. *Computers* 2014, *3*, 130-173

[10] Saint-Andre, P.; Smith, K.; Tronçon, R. *XMPP: The Definitive Guide*; O' Reilly Media Inc.: Sebastopol, CA, USA, 2009; p. 306.

[11] Sensor over XMPP. Available online: http://xmpp.org/extensions/inbox/sensors.html (accessed on 30 October 2014).

[12] MQTT protocol. Available online: http://mqtt.org (accessed on 30 October 2014).

[13] IETF IPv6 over Low Power WPAN (6LowPAN) WG. Available online: http://tools.ietf.org/wg/6lowpan (accessed on 30 October 2014).

[14] Hui, J. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. IETF RFC 6282, September 2011. Available online: http://tools.ietf.org/html/rfc6282 (accessed on 30 October 2014)

[15] Constrained RESTful Environments (CoRE) WG. Available online: http://tools.ietf.org/wg/core/ (accessed on 30 October 2014).

[16] IETF Routing Over Low power and Lossy networks (ROLL) WG. Available online: http://tools.ietf.org/wg/roll/ (accessed on 30 October 2014).

[17] Constrained RESTful Environments (CoRE) Link Format, IETF RFC 6690. Available online: http://tools.ietf.org/html/rfc6690 (accessed on 30 October 2014).

[18] Kovatsch, M.; Duquennoy, S.; Dunkels, A. A Low-Power CoAP for Contiki. In Proceedings of the Workshop on Internet of Things Technology and Architectures (IEEE IoTech 2011), Valencia, Spain, 17 October 2011.

[19] Rescorla, E.; Modadugu, N. “Datagram Transport Layer Security” IETF RFC 4347, April 2006. Available online: http://tools.ietf.org/html/rfc4347 (accessed on 30 October 2014).

[20] Kent, S. “IP Encapsulating Security Payload (ESP)”, RFC 4303, December 2005. Available online: http://tools.ietf.org/html/rfc4303 (accessed on 30 October 2014).

[21] Yegin, A.; Shelby, Z. CoAP Security Options. Internet-Draft, October 14, 2011, Expired April 16, 2012. Available oneline: http://www.ietf.org/archive/id/draft-yegin-coap-security-options-00.txt (accessed on 30 October 2014).

[22] Saint-Andre, P. Extensible Messaging and presence protocol (XMPP): IETF RFC 3921. Available online: https://tools.ietf.org/html/rfc3921 (accessed on 30 October 2014).

[23] Instant Messaging/Presence Protocol Requirements. RFC 2779, February 2000. Available online: https://tools.ietf.org/html/rfc2779 (accessed on 30 October 2014).

[24] http://www.cambridgewireless.co.uk/docs/LTE%20Evolution%20for%20Cellular%20IoT%2010.04.14-1.pdf

6 A view to the State-of-the-Art on Security of M2M Systems

6.1 Overview

Security solutions developed for traditional IT networks are not effective in grid networks because of the major differences between them. Their security objectives are different in the sense that security in IT networks aims to enforce the three security principles (confidentiality, integrity and availability), while the security in automation (grid) networks aims to provide human safety, equipment and power lines protection, and system operation. Moreover, the security architecture of IT networks is different than that of the Grid network since security in IT networks is achieved by providing more protection at the center of the network (where the data resides), while the protection in automation networks is done at the network center and edge. Their underlying topology is also different where IT networks use a well-defined set of operating systems (OSs) and protocols, while automation networks use multiple propriety OSs and protocols specific to vendors. Finally, their Quality of Service (QoS) metrics are different in the sense that it is acceptable in IT networks to reboot devices in case of failure or upgrade, while this is not acceptable in automation networks since services must be available at all times.

The emergence of the M2M ecosystem has led to the revision of the conventional network security paradigm [5]. The M2M communication systems, by essence, interconnect heterogeneous network segments, where the heterogeneity is expressed in terms of functional capabilities and capacities. This encompasses, for instance, the interconnection of a WAN with low-power network segments like WSNs in multihop communication schemes. The security of communications involving several hops is usually addressed using hop by hop security where each segment of the communication path from source to destination is secured using distinct credentials possibly managed by distinct parties involving rekeying operations at each transmission node. Credential management with this model may be complex. Furthermore, in order to provide reasonable security, the various parties involved in credentials distribution and management needs to trust each other. This requirement is often difficult to achieve in the case of M2M communications and it is generally accepted that end to end security involving the use of a single set of credential from source to destination is a better model. However the deployment of end to end security is a challenging task as it requires solving the problem of credential distribution at a global level. [4]

6.2 Security of communication

Table 1 lists the major security technologies used in the LAN (device) domains to offer, e.g., authentication, integrity and confidentiality. Table 2 lists the major network technologies, or security technologies used to secure the network access. These technologies provide, e.g., interoperability for secure communication in WANs but in some cases also secure and interoperable communication between devices in WANs and devices in LANs.

Table 1. Review of security technologies used in the LAN device domains to offer, e.g., authentication, integrity and confidentiality.

Technology, What to Solve?	Forum(s), References	Main Contribution
Security technologies of Bluetooth To enable authentication and pairing of Bluetooth devices, to encrypt the transmitted data, and to check integrity of transmitted data packets in Bluetooth network.	Bluetooth Special Interest Group (SIG)	Frequency-hopping spread spectrum Master clock is shared to slaves Binding of devices is done through selected pairing mechanisms Different security modes Link keys to establish authenticated and/or encrypted ACL links SAFER+ block cipher (not for encryption!) E0 stream cipher Secure Simple Pairing (SSP) with Elliptic Curve Diffie-Hellman (ECDH).
Security technologies of Bluetooth low energy	Bluetooth Special Interest Group (SIG)	AES-CCM block cipher

Similar problems to solve as in BT.		Data can be signed with Connection Signature Resolving Key (CSRK), MAC, counter Privacy feature to change private address on a frequent basis

Security technologies of ZigBee Authentication, encryption, integrity checking, countermeasures against replay-attacks.	ZigBee Alliance	16 channel hopping Key establishment, key transport, frame protection, ad device management implemented mostly at the network (NWK) and application support sub-layer (APS). AES-CCM to offer confidentiality and authentication, AES-CBC-MAC to offer authentication or AES-CTR to offer confidentiality. 32-, 64-, or 128-bit MAC and 128-bit key. Key sequence counter
Security technologies of ZigBee RF4CE (Even lower power and simplified version of ZigBee)	ZigBee Alliance	3 channel hopping Simpler pairing mechanism than in ZigBee AES-CCM with 128-bit key.
Security technologies of Wi-Fi Authentication, encryption, integrity, replay protection		WPA2 that uses Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) based on 128-bit AES-CCM block cipher Cryptographic hash function Key management Replay protection WPA2 can be used in two modes: Shared key: relying upon the use of a key shared by all Wi-Fi clients and access point. Enterprise, relying upon the use of an external AAA server
Default security technologies of EPCglobal UHF Class 1 Generation 2(v1.2.0) RFID tags To prevent unauthorized writing and disabling and counterfeiting of RFID tags.	EPCglobal	32-bit KILL and ACCESS passwords (perma)locking of memory Unique unprogrammable TID numbers

Table 2. Network related security technologies used to enable secure network access.

Technology, What to Solve?	Forum(s), References	Main Contribution
IPsec	IETF's IP Security Protocol (IPsec) concluded working group	IPsec WG developed a security protocol in the network layer to provide cryptographic security services to support combinations of authentication, integrity, access control, and confidentiality. IPsec has been implemented, e.g., over 6LoWPAN [6].
EAP based authentication EAP/TLS EAP/TTLS EAP/SIM EAP/AKA EAP/PEAP	IETF network working group	EAP is an authentication protocol commonly used to secure access to wireless networks and point to point connections.
TLS/DTLS To choose cipher suites for providing (mutual) authentication of end-points,	IETF's Transport Layer Security (TLS) working group	Specifying new TLS and DTLS protocols and extensions to them. They run on top of transport layer protocols and provide, e.g., confidentiality and data integrity between two communicating

encryption and integrity of transmitted data.		applications. Both has been implemented, e.g., over 6LoWPAN.
WTLS To offer data integrity, confidentiality and authentication of end-points.	Open Mobile Alliance (OMA)	Similar to TLS but originally meant to be used devices without TCP/IP. Works over UDP or WAP Datagram Protocol (WDP)
3GPP	3rd Generation Partnership Project's (3GPP) Service and System Aspects (SA) working group 3	IMS security Security of multimedia broadcast and multicast service (MBMS) Generic Bootstrapping Architecture (GBA) Key establishment mechanisms Ongoing: security for system improvement for machine-type communications Lawful interception

M2M communications are commonly classified into three main communication domains: M2M device, network and application domains. Those three domains are associated to three distinct security domains involving generally distinct business actors to perform credential management.
Thus, the device domain corresponds to M2M capillary communications occurring in a LAN or PAN proximity network. Generally the owner of this network has the responsibility to manage the credentials used to secure those communications occurring between the devices and the gateway.
In some cases there is a need to secure the connection between the Gateway/device and the point of entry to the Internet (WAN). This is the case when using 3GPP communications to protect the radio transmission. In this case, the SIM card holding credentials used to secure the access to the wide area network is managed by the mobile network operator to achieve "network access security".

Table 3 lists security technologies that have been or can be used at application layer, e.g., in M2M services, and also in M2M architectures.

Table 3. Security technologies applicable for applications layer.

Technology, What to Solve?	Forum(s), References	Main Contribution
JSON security To provide authentication, integrity and confidentiality, access control and resource control	IETF's JavaScript Object Signing and Encryption (JOSE) working group Used, e.g., in/by SAML 2.0, WS-Federation, OpenID, OAuth 2.0, XMPP, ALTO and to provide integrity of exigent (alarms)	Standardizing integrity protection and encryption security services, in order to increase interoperability of security features between protocols that use JSON.
XML security To provide authentication, integrity and confidentiality.	W3C's XML security working group	XML Signature provides integrity, message authentication, and/or signer authentication services for data. XML Encryption specifies a process for encrypting data and representing the result in XML. XKMS specifies protocols for distributing and registering public keys.
HTTP Strict Transport Security (HSTS)	IETF's Web Security (websec) working group	HSTS is designed to allow web sites or http servers to tell users' browsers or http clients that they want to communicate only over an encrypted connection.
DNSSEC To provide secure DNS.	IETF's Domain Name System Security (dnssec) working group	Enhancements to the secure DNS protocol.

ETSI M2M security architecture	ETSI	Pre-provisioned device/gateway credential types, e.g., SIM/AKA or X.509v3 certificates. Defining M2M bootstrap procedures based on GBA, TLS or EAP/PANA. Securing M2M service connection by GBA, TLS or EAP/PANA. Securing mId by TLS or DTLS, XML security or relying access network security.
SASL To provide authentication.	IETF's Simple Authentication and Security Layer (SASL) concluded working group	A framework for authentication and data security in Internet protocols. Application protocol that uses SASL can in theory use any authentication mechanisms supported by SASL.
HTTPS Authentication of end-points, protection against MitM attacks, encryption of communication	IETF	Usage of TLS to secure HTTP connections
AAA protocols To authenticate entity's identification To authorize the entity To account aka track a network resource consumption	IETF's Authentication, Authorization and Accounting (AAA) concluded working group, RADIUS Extensions (radext) working group, Diameter maintenance and Extensions (dime) working group 3GPP	Diameter Radius 3GPP EPS AAA
Trusted Key Distribution Centers (KDC) Authentication and key exchange, sometimes also access control	MIT (Kerberos) OpenID Foundation	Kerberos 5 OpenID Single Sign-On (SSO)

6.3 Access control

A typical network security service (e.g., authentication, access control, data confidentiality, etc.) is made up of two phases. The protection of sensitive information is typically achieved through access control mechanisms. Several access models such as DAC [17, 18], MAC [19] and RBAC [20], has been proposed to regulate access to information. One major drawback of these approaches is the limited expressiveness of the policies that can be specified. In the last year attribute-based access control (ABAC) paradigm is emerging to for the specification of access control policies. Such a paradigm has moved away from the “classical” view of access control, which was based on (authenticated) users and their respective identities. In the classical view, an access request was modeled as a triple (s,o,a), where s denoted a subject (corresponding to a user identity), o denoted an object (corresponding to the identity of some protected resource) and a denoted an access mode (such as read or write). In ABAC, an access request is modeled as a collection of attribute name-value pairs. ABAC is particularly suitable in “open” computing environments [21] where the user population is not known in advance and access is allowed or denied on the basis of user characteristics, rather than identities.

The ABAC paradigm has promoted several initiatives both in academia and industry. The most prominent initiative is XACML [22], which has become the de facto standard for policy specification and enforcement. XACML is an OASIS standard that provides an attribute-base language implemented in XML for the specification of access control policies and a reference architecture for their enforcement. Several open-source XACML implementations are currently available, for instance SUN-XACML, HERAS-AF [23], XEngine [24], enterprise-java-xacml and WSO2 Balana. A common characteristic of existing XACML implementations is that they are implemented as a monolithic component. Although the choice of implementing the XACML reference architecture as a monolithic component may be suitable for enterprise environment, it cannot fully address the key challenges for the enforcement of access control policies in open and distributed systems and cloud environments. In particular they are not flexible and it is difficult to adapt them to accommodate different deployment configurations according the current needs. Moreover, although the XACML standard provides extensibility points to extend the functionality of the authorization mechanisms (e.g. user define functions), the monolithic nature of current XACM implementations limit the exploitability of these extensibility points.

The M2Mgrid project will address the limitations of XACML by defining a novel architectural framework for the development of authorizations services tailored to enforcing access control policies in power management systems. The authorization services will provide great extensibility to support the retrieval of information necessary for policy evaluation from external sources as well as the relocation of the computation of complex functions thus relieving the burden on the authorization service. Moreover, the authorization service will provide great flexibility by supporting a variety of deployment configurations as demanded by the ADM use cases.

6.4 Challenges for Security Solutions

When dealing with security algorithms, protocols and policies for the IoT-based SG, several challenges need to be taken into consideration [2]:

- Scalability: The SG could span over large areas (several cities or the entire country), and involves a large number of smart devices and objects. This will make it difficult to conceive scalable security solutions, such as key management and authentication.
- Mobility: with mobile devices/objects, such as e-cars and on-the field technical agents, there will be a continuous need for authentication and secure communication with a changing surrounding (smart meters, electric charging stations, etc.).
- Deployment: Since the SG could span to the entire country, objects/devices are deployed at a large scale, work unattended, and could be placed on remote places with no physical perimeter protection, making them easily accessible. Security solutions should be able to detect any attempt to tamper with them.
- Legacy systems: Already deployed systems and devices, could have a little or no support for security, since they were based mostly on proprietary solutions (hardware and software), deployed on isolated islands with no communication, or through private communication infrastructure. Integrating those legacy systems to the IoT based SG is a real challenge, since in most cases there is no way to replace them with new systems, or update them so they can support the desired security solutions.
- Constrained Resources: several devices/objects of the SG, especially those massively deployed are resource constrained. Special care need to be taken when developing security solutions, to be sure that their limited resources could accommodate the solutions. This make applying classical security solutions, especially those based on public-key cryptography or on PKI, a challenge.
- Heterogeneity: Due to the discrepancy on the resources of the devices/objects on the SG (memory, computation, bandwidth, energy autonomy, time-sensitivity, etc.), and their implemented protocols and communication stacks (for non IP-based devices) achieving secure end-to-end communications is a challenging task, requiring the most often adaptation of existing solutions or even using gateways.
- Interoperability: It could be seen as one of the consequences of protocols and communication stacks heterogeneity, between devices/objects in the SG. Legacy system and devices/objects that couldn‘t support TCP/IP stack (ex, Zigbee v1, HART) could not communicate with IP-based systems and devices/objects, unless through gateways, making end-to-end secure communication impossible. Interoperability could also be seen between two devices implementing the same protocols and communication stacks, but different feature capabilities: one with fully support, the other with partial support (ex, DTLS with/without certificate support)
- Bootstrapping: How to efficiently bootstrap the millions of devices/objects of SG with the necessary initial keying materials (cryptographic keys, cryptographic functions/algorithms and parameters, etc.)?

- Trust Management: Objects/devices on the SG could be managed by different entities (end-users for smart appliances, SG's operator for smart meters and sensors, etc.). Objects/devices couldn't communicate if a minimal trust level isn't established. While objects/devices owned/managed by the same entity could easily establish a trust relationship, building trust between objects/devices owned/managed by different entities is a challenge, especially in such large-scale network.
- Latency/Time Constraint: Some parts of the SG need to respond on a real-time basis to events and messages. For instance, electric SCADA (Supervisory Control and Data Acquisition) system8, used on transmission and distribution sub-stations, must respond on a real-time basis to any variation on current, voltage or frequency values of the electricity in addition to other meteorological parameters influencing equipment's functioning all provided by different kind of smart objects (sensors, actuators, etc. etc.), in-order to keep the assets safe and prevent the propagation of anomalies (power overload or outage) to other parts of the power grid. This makes time-consuming operations (i.e. public-key operations) not suitable.

References

[1] Fadi Aloula, A. R. Al-Alia, Rami Al-Dalkya, Mamoun Al-Mardinia, Wassim El-Hajjb, "*Smart Grid Security: Threats, Vulnerabilities and Solutions*", International Journal of Smart Grid and Clean Energy, vol. 1, no. 1, September 2012: pp. 1–6, ISSN: 2315-4462, DOI: 10.12720/sgce.1.1.1-6

[2] Chakib Bekara, "*Security Issues and Challenges for the IoT-based Smart Grid*", Procedia Computer Science, Volume 34, 2014, Pages 532-537, ISSN 1877-0509, DOI: 10.1016/j.procs.2014.07.064

[3] Khurana, H., Hadley, M., Ning Lu, Frincke, D.A., "*Smart-grid security issues*", Security & Privacy, IEEE, Vol. 8, Issue: 1, pags. 81-85, February 2010, ISSN: 1540-7993, DOI: 10.1109/MSP.2010.49

[4] Latvakoski, J.; Iivari, A.; Vitic, P.; Jubeh, B.; Alaya, M.B.; Monteil, T.; Lopez, Y.; Talavera, G.; Gonzalez, J.; Granqvist, N.; Kellil, M.; Ganem, H.; Väisänen, T. A Survey on M2M Service Networks. *Computers* 2014, *3*, 130-173

[5] Inhyok, C.; Shah, Y; Schmidt, A.U.; Leicher, A.; Meyerstein, M. Security and trust for M2M Communications. *IEEE Veh. Tech. Mag.* 2009, 69–75.

[6] Raza, S.; Chung, T.; Duquennoy, S.; Yazar, D.; Voigt, T.; Roedig, U. Securing Internet of Things with Lightweigth IPsec. Available online: http://soda.swedish-ict.se/4052/ (accessed on 30 October 2014).

[7] Judge, P.; Ammar, M. Security issues and solutions in multicast content distribution: A survey. *IEEE Netw*. 2003, *17*, pp 30–36.

[8] Challal, Y.; Bettahar, H.; Bouabdallah, A. A Taxonomy of Multicast Data Origin Authentication: Issues and Solutions. *IEEE Commun. Surv.* 2004, *3*, 34–57.

[9] Kellil, M.; Romdhani, I.; Lach, H-Y. Multicast Receiver and Sender Access Control and its Applicability to Mobile IP Environments: A Survey. *IEEE Commun. Surv.* 2005, *7,* 46–70.

[10] Hardjono, T.; Weiss, B. The Multicast Group Security Architecture. Available online: https://tools.ietf.org/html/rfc3740 (accessed on 30 October 2014)

[11] ETSI. Security of Multimedia Broadcast/Multicast Service (MBMS). Available online: http://www.etsi.org/deliver/etsi_ts/133200_133299/133246/10.01.00_60/ts_133246v100100p.pdf (accessed 30 October 2014)

[12] ETSI. Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM). Available online: http://www.etsi.org/deliver/etsi_ts/102400_102499/102466/01.01.01_60/ ts_102466v010101p.pdf (accessed on 30 October 2014)

[13] Ben Jaballah, W.; Meddeb, A.; Youssef, H. An efficient source authentication scheme in wireless sensor networks. In Proceedings of the Proceedings of IEEE/ACS AICCSA Conference, Hammamet, Tunisia, 16–19 May 2010; pp. 1–7.

[14] Lim, S.Y.; Lim, M.H. Energy-Efficient and Scalable Group Key Management for Hierarchical Sensor Network. *J. Ubiquitous Syst. Pervasive Netw.* 2011, *2*, 39–47.

[15] Alexander, R.; Tsao, T. Adapted Multimedia Internet KEYing (AMIKEY): An extension of Multimedia Internet KEYing (MIKEY) Methods for Generic LLN Environments. Available online: http://tools.ietf.org/html/draft-alexander-roll-mikey-lln-key-mgmt-04 (accessed on 30 October 2014)

[16] Keoh, S.; Kumar, S.; Garcia-Morchon, O.; Dijk, E.; Rahman, A. DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs). Available online: http://tools.ietf.org/html/ draft-keoh-dice-multicast-security-04 (accessed 30 October 2014)

[17] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in Operating Systems. Communications of the ACM 19(8): 461-471. 1976

[18] Butler W. Lampson. Protection. ACM SIGOPS Operating Systems Review 8(1): 18-24. 1974.

[19] Ravi S. Sandhu. Lattice-Based Access Control Models. Computer 26(11):9-19. 1993.

[20] Ravi. S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman. Role-Based Access Control Models, IEEE Computer 29(2): 38-47. 1996

[21] H. Takabi, J. Joshi and G.-J. Ahn, Security and privacy challenges in cloud computing environments, IEEE Security Privacy, 8(6) 24-31. 2010.

[22] OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard. 2013.

[23] S. Dolski, F. Huonder and S. Oberholze. HERAS-AF: XACML 2.0 Implementation, Technical report, University of Applied Sciences Rapperswil. 2007.

[24] A. X. Liu, F. Chen, J. Hwang and T. Xie. Designing fast and scalable XACML policy evaluation engines, IEEE Trans. Computers, 60(12) 1802-1817. 2011.

7 Concluding Remarks

The provided view to the state of the art and practises is just a snapshot of the available technologies related to the scope of M2MGrids project described within the project at the execution time of the related tasks. Therefore, it is important for the reader to take into concern that the majority of the analysis results are from the 1^{st} project year i.e. 2016.

-