29

TRESCIMO: Towards Software-Based Federated Internet of Things Testbeds across Europe and South Africa to Enable FIRE Smart City Experimentation

Nyasha Mukudu[1], Ronald Steinke[2], Giuseppe Carella[3], Joyce Mwangama[1], Andreea Corici[2], Neco Ventura[1], Alexander Willner[3], Thomas Magedanz[2], Maria Barros[4] and Anastasius Gavras[4]

[1]University of Cape Town, South Africa
[2]Fraunhofer FOKUS, Germany
[3]Technical University of Berlin, Germany
[4]Eurescom GmbH, Germany

Abstract

Smart Cities are able to offer efficient services for creating more sustainable environments. However, this will require stakeholders to have access to suitable testbed infrastructure for prototyping new services. This chapter describes the underlying federated testbed infrastructure, designed and implemented within the EU TRESCIMO project, to support Smart City experimentation. The focal point of this chapter is to share the technical challenges on the design and implementation decisions made to provide a sustainable state of the art federated testbed infrastructure between the Technical University of Berlin (TUB) and the University of Cape Town (UCT). The starting point for this testbed implementation is the layered TRESCIMO Smart City reference architecture, onto which existing standard compliant and SDN/NFV based testbed toolkits have been mapped. The chapter describes these toolkits and their integration within the context of TRESCIMO. Furthermore, it will outline how this SDN-based infrastructure setup can be utilised in future FIRE SDN projects, such as SoftFIRE.

Keywords: Experimentation, Internet of Things, Network Function Virtualisation, Smart City, Software Defined Networks.

29.1 Introduction

Many cities around the world are presently confronted with the challenge of providing services that address economic, social and environmental requirements faced by residents. A drastic increase in urbanisation is expected to result in two-thirds of the world population residing in cities [5]. This will result in an unprecedented demand for services such as energy management, clean water, healthcare and transportation just to name a few. This has driven many cities to investigate the adoption of Smart Cities services that utilise Internet of Things (IoT) generated data to make informed decisions which result in enhanced city services. Smart Cities have the potential to intelligently manage available resources in order to create integrated, habitable and sustainable urban environments [1]. The realisation of the Smart City vision will require the integration of various domains within a common framework.

Machine-to-Machine (M2M) communication technology will enable various physical objects within cities to be connected. Smart City services should have the capability to analyse the data and provide instant real-time solutions for many challenges faced by cities. Additionally, many different technical and non-technical service requirements must be considered. Consequently, large-scale experimentation is required to provide the necessary critical mass of experimental data required by businesses and end-users to evaluate the readiness of M2M and other IoT technologies for market adoption.

In this chapter, we describe a prototyping environment implemented to address the need for testing facilities for Smart Cities. This testing environment, developed as part of the Testbeds for Reliable Smart City Machine-to-Machine Communications (TRESCIMO) project, includes participants from Europe and South Africa, and allows for the ability to test Smart City use cases from both a developed and developing world context. Many partners were involved in the testbed deployment and experimentation aspects of the project: Technical University of Berlin (TUB) in Germany; University of Cape Town (UCT) in South Africa; Fraunhofer FOKUS Institute (FOKUS) in Germany; Council for Scientific and Industrial Research (CSIR) in South Africa; and Fundaci i2CAT (i2CAT) in Spain.

29.2 Problem Statement

The need for large-scale testbeds for Smart Cities has been recognised by industry and academia in order to develop a reference implementation model for Smart Cities [9]. However, creating an experimental facility capable of coping with the diverse nature of Smart City services and the number of connected devices remains a challenge. In addition, the expected growth in demand for services makes it imperative for testbeds to allow service providers to create strategies to cope with these increases.

In the TRESCIMO project, we created a federation of testbeds that allows for experimentation which makes use of enabling technologies, standardised platforms and applications for Smart Cities. These Smart Cities can have different needs and requirements thus allowing for flexible configurations depending on contexts and use cases. Experimental tools are incorporated as an effective option to study the behaviour of integrated software and hardware before implementing real deployments.

The TRESCIMO facility is based on a virtualised standardised M2M platform Open Machine Type Communications Platform (OpenMTC) and an open-source, Slice-based Federation Architecture (SFA)-compatible framework for managing and federating testbeds, Future Internet Teagle (FITeagle). The facility consists of three interconnected sites located in Berlin – Germany, Cape Town – South Africa and Pretoria – South Africa. Additionally, the ability to federate with other testbeds will allow the sharing of resources (i.e. sensors, actuators and data) among different services and users regardless of their location.

29.3 Background and State of the Art

In order to evaluate new protocols and architectures that will enhance Smart City deployments, testbeds that incorporate a wide range of heterogeneous resources are needed. Additionally, considerations should be made on the variability of resources, the size of traffic or data generated, and the operational complexity they introduce. Although there are several existing wireless and wired testbeds that offer researchers the ability to perform Smart City related experimentation, many of these testbed do not cater from some important user specific requirements. Some of these requirements include ease of user management, experiment control and connectivity.

Different approaches have been developed to overcome these issues. The focus of the SmartSantander project was to create a European experimental test facility for architectures, key enabling technologies, services and applications for smart cities [15]. The facility had to support both experimenters and real world end users using sensors deployed within the city of Santander in Spain. This was achieved using separate planes for IoT data and testbed management data. In addition, the suitability of IoT technology for supporting real world smart cities was demonstrated.

In the GEneralized architecture for dYnamic infrastructure SERvice (GEYSERS) project, cloud-based infrastructure is provided to experimenters as an on demand service [2]. This approach results in a more seamless and coordinated provisioning of virtual infrastructures composed of network and IT resources. Adopting a similar approach, TRESCIMO was able to cope with the diverse range of deployment scenarios found in smart cities. The Federated E-Infrastructure Dedicated to European Researchers Innovating in Computing Network Architectures (FEDERICA) project is used to create virtual versions of physical testbed resources [3].

29.4 Smart City Testbed Design

The TRESCIMO research facility was designed to allow for experimentation making use of enabling technologies, standardised platforms and applications for Smart Cities with different configurations. This section provides details on key requirements for Smart City experimentation facilities identified using use cases from various Smart City application domains. Based on these requirements, the architecture of the federated TRESCIMO testbed was developed.

29.4.1 Design Considerations

Key requirements for testbeds to adequately support IoT experimentation have been identified [9]. However, Smart City services impose additional requirements on testbeds due to the diverse range of possible usage scenarios. As a result, use cases for Smart City experimentation were selected from the energy management, environmental monitoring, healthcare, safety and education domains. This wide range of use cases were analysed, and common experimentation requirements were identified.

In the TRESCIMO facility, a collection of services and infrastructures are used to create an environment capable of meeting the needs of Smart City experimenters. The following subsections highlights key requirements and considers how they are addressed within the TRESCIMO testbeds.

29.4.1.1 Federation

Federation with other testbeds is required to allow experimenters access to a greater pool of available resources [9]. Resources that can be shared include data produced by sensors, control of actuators and integrated M2M/IoT components deployed in the individual testbeds. The TRESCIMO facility combines the CSIR, TUB and UCT testbeds to provide Future Internet Research and Experimentation (FIRE) users with access to resources from the individual testbeds. Furthermore, this creates a facility capable of conducting experiments in both a developing African and a developed European context [4].

29.4.1.2 Heterogeneity

Smart Cities consist of various domains (e.g. healthcare and energy), that utilise a wide range of applications, platforms and devices. Consequently, it is necessary to adopt integrated cloud-oriented architectures of networks, services, interfaces and data analysis tools to meet the needs of future smart cites [11]. TRESCIMO uses a standardised M2M platform to provide support for a wide range of IoT devices and technologies. Furthermore, experimenters are able to provision various virtual smart city infrastructures, to enable the deployment of various experimental scenarios.

29.4.1.3 Scale

Smart City services will have to cope with a large amount of data from sources such as IoT nodes, other services and people. In order to facilitate large-scale experimentation, current IoT testbeds such as SmartSantander offers access to thousands of nodes [15]. The TRESCIMO facility uses a combination of physical and virtual IoT devices to allow experimenters access to a larger number of devices.

29.4.1.4 Reliability

The testbed facility is intended to allow users to run their experiments to completion without any unexpected interruptions. Consequently, it is necessary for testbed operators to utilise monitoring tools to ensure that the facility and its infrastructure are operational. In the case of TRESCIMO, experimenters are able to receive monitoring data for infrastructure used in a particular experiment.

29.4.1.5 Resource management

In order to manage the connected resources in the testbed, it has to support resource discovery, resource reservation and resource provisioning. In the

FIRE context this functionality is provided to FIRE experimenters via a standards-based federation interface.

29.4.1.6 Flexibility

The testbed is designed to be extendible to meet future requirements or additional use cases. This allows for the realisation of a wide range of users, use cases and complex infrastructures. In the TRESCIMO facility, a modular approach was adopted to ease the process of adding new platforms and devices. This high level of flexibility will allow for experimenters to develop new services and devices.

29.4.2 Architecture Overview

The TRESCIMO architecture was designed to provide IoT testbed Smart City use cases. Additionally, the architecture can be partly or fully mapped to trials that verify the functionality in real world scenarios. In order to provide flexibility and accelerated setup for the different use cases and experiments, an Software-Defined Networking (SDN)/Network Functions Virtualisation (NFV) approach was selected. This enabled the capability to provide testbed components as virtualised resources that could be instantiated on-demand as required. In addition to virtualised resources, physical real-world resources and devices were also provided and integrated for dynamic and flexible utilisation.

Figure 29.1 illustrates the overview architecture. The left side of the diagram highlights the general reference architecture while the right side shows the extracted elements from the general architecture concentrating on the elements implemented or deployed at the various geographical locations. In the lower layer of the diagram, the different devices used for the different use cases are shown. These include the Smart Energy trial devices developed by Eskom and CSIR, the environmental monitoring devices deployed by i2CAT. Devices were connected to the rest of the architecture by using simple area networks such as LAN or WLAN, and when this was not possible they were connected via a Delay Tolerant Network (DTN).

Common to all locations is the use of an M2M middleware layer. The middleware exposes gateways that allow for interconnection of devices to the backend of the middleware. The M2M middleware, OpenMTC, developed by FOKUS was adopted. Optionally, the Smart City Platform (SCP) developed by CSIR, could be deployed to provide an enriched depiction or analysis of the

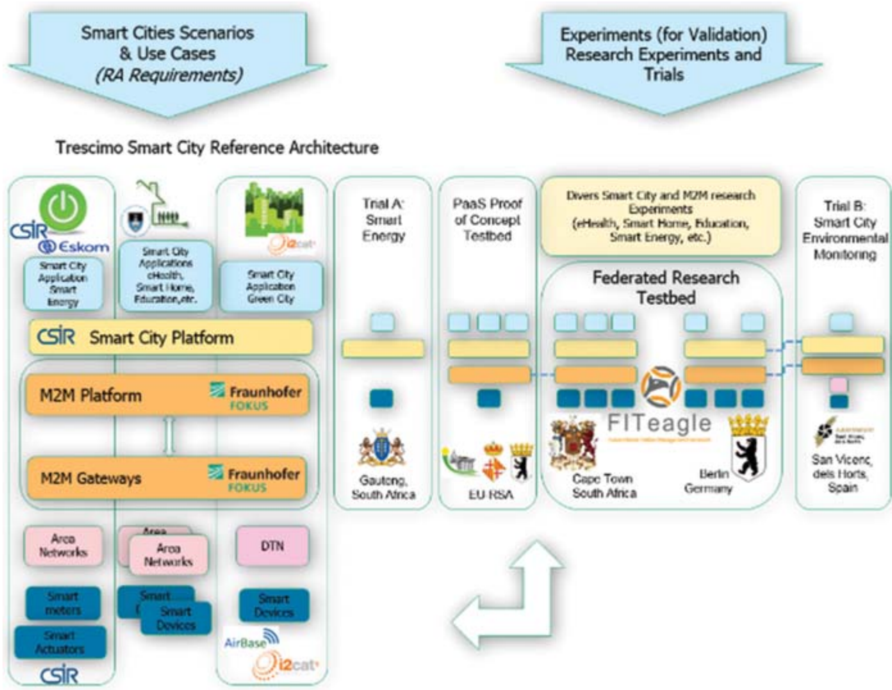

Figure 29.1 Reference architecture of TRESCIMO.

aggregated data exposed by the M2M middleware platform. This SCP provides interfaces for the applications that were used and tested in the experiments and trials. The complete stack of the architecture was federated and managed by FITeagle from TUB, which also provides the standardised SFA interface for experimenters.

The final deployment in Figure 29.2 shows the federation with all involved sites connected via the Internet. The TUB testbed (on the left) implements all the components of the TRESCIMO architecture stack. The TUB testbed is able to integrate physical IoT devices and can also emulate virtual devices. The UCT testbed (on the right) implements all the components except for the SCP. This testbed also integrates physical devices. The CSIR Proof-of-Concept testbed (in the middle) implements a small subset of the architecture in order to demonstrate the use case of the energy management trial within the TRESCIMO stack. In this trial, due to regulations and restrictions, only the SCP and the Active Devices could be deployed. All three testbeds are

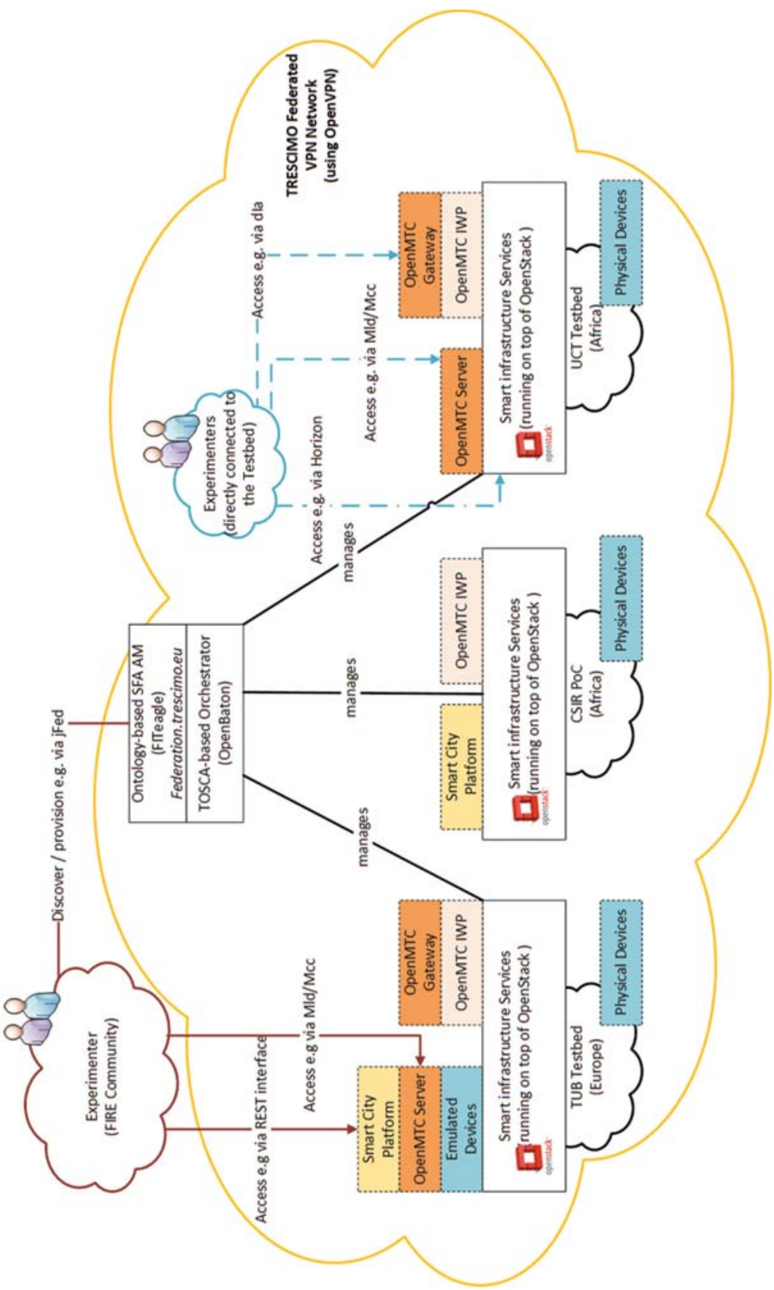

Figure 29.2 Current TRESCIMO federated testbed architecture (based on [4]).

based on an OpenStack setup in order to deploy the needed parts of the architecture. These OpenStack instances are managed by FITeagle with the help of OpenBaton. FITeagle provides the interface to the experimenter in order to setup the experiments. It returns the established endpoints to the instances that are created so that the experimenter is able to connect to the created testbed resource.

29.5 Technical Work/Implementation

Several software tools were utilised to realise the designed reference architecture. To enable access to external experimenters, FITeagle was used as it provides a standardised SFA interface. FITeagle and OpenStack were further interworked with OpenBaton which allowed for enhanced orchestration features of the testbed resources. OpenBaton provides a Topology and Orchestration Specification for Cloud Applications (TOSCA) interface to FITeagle. For the transportation of M2M data a combination of the toolkits OpenMTC and Open5GMTC were used.

29.5.1 Cloud Management – OpenStack

Smart city infrastructure is hosted and managed using OpenStack based cloud servers[1]. An OpenStack server is deployed at each of the individual testbed sites in the TRESCIMO federation as shown in Figure 29.3. Each site hosts a selection of the TRESCIMO Smart City stack and is controlled by a shared NFV Orchestrator (OpenSDNCore). The OpenSDNCore communicates with the OpenStack controllers at each site in order to launch the required components of the TRESCIMO smart city stack.

Each of the individual testbed sites utilises a unique addressing scheme for locally available resources. To meet the requirements of smart city experimenters, it was necessary to enable experimenters to remotely provision and access components in the TRESCIMO smart city infrastructure. To support the provisioning process, the individual OpenStack controllers are interconnected to the OpenSDNCore orchestrator. Public IP addresses are assigned to the relevant endpoints of the provisioned infrastructure. Furthermore, private IP addresses are used to allow sensors to communicate with provisioned M2M gateways and servers. In addition, communication among provisioned virtual machines was achieved using a common IP range (trescimo-net).

[1]https://www.openstack.org/

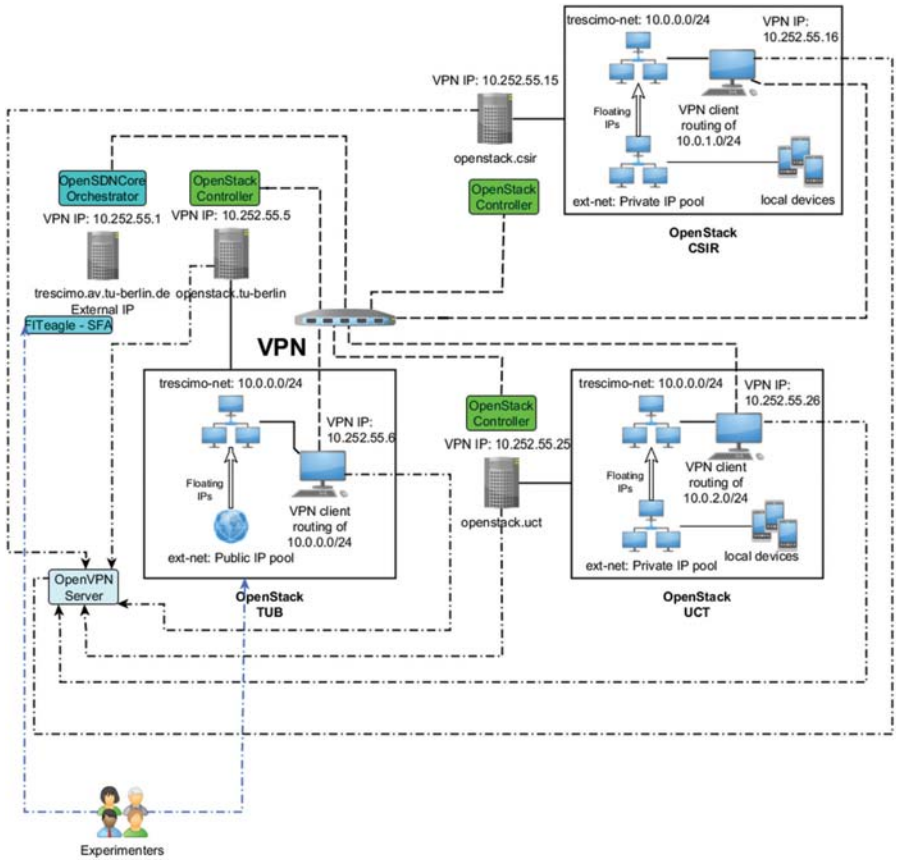

Figure 29.3 Testbed interconnections.

29.5.2 Experimentation Management – FITeagle

TRESCIMO uses FITeagle [16], the first implementation of a semantic-based Slice-based Federation Architecture (SFA) Aggregate Manager (AM), to dynamically provision Software Defined Infrastructures (SDI) for the Smart City context. These SDIs are made available to the Global Environment for Network Innovations (GENI) [14] and FIRE community. For the TRESCIMO testbeds, the FITeagle framework offers interfaces to the FIRE community and handles all aspects of the experiment life cycle including Authentication (AuthN) and Authorization (AuthZ). This is based on X.509 certificates signed by trusted Certificate Authorities (CAs), such as from Federation for FIRE[2] or PlanetLab Europe (PLE)[3].

While physical devices are managed by dedicated resource adapters deployed within FITeagle, requests for virtualised services are forwarded to the OpenSDNCore Orchestrator framework via its TOSCA interface. Depending on the selected location, the relevant OpenStack sites are then contacted to instantiate the requested services. The underlying workflow with references to the related SFA AM method calls is depicted in Figure 29.4.

Due to the fact that the TRESCIMO facility uses the OpenSDNCore Orchestrator to provisionVirtualised Network Function (VNF)s internally, compatibility with the offered TOSCA [13] interface was improved. For this, the Open-Multinet (OMN)[4] ontology and translator, developed within the World Wide Web Consortium (W3C) Federated Infrastructures Community Group[5], were extended to provide the required properties and mappings to GENI v3 Resource Specifications (RSpecs) [12].

Figure 29.5 highlights the resource adapters implemented to enable FITeagle to interact with virtual and physical infrastructure for realising the required API functionality. These resource adapters are:

- **TOSCA Resource Adapter (TOSCA RA)** is responsible for communicating with the OpenSDNCore for orchestration of virtual infrastructure. OpenBaton receives a Resource Description File (RDF)-style topology file and converts it into a TOSCA-compatible topology for use by the OpenStack controller. This will result in the provisioning of the required Smart City infrastructure and feedback on the created endpoints will be provided to the experimenter in the form of an RDF response.
- **Interworking Proxy Resource Adapter (IWP RA)** handles interaction with the interworking proxy and the devices it manages. This resource adapter provides experimenters with details on the connected devices and options for setting the URIs of the M2M endpoints used in a particular experiment. As a result, experimenters can instruct the interworking proxy to create resources inside the desired M2M endpoints. Connected devices can be addressed by using the resource URIs or can be found via discovery on a specified M2M gateway. Devices can be connected to the federated testbed using a wired connection or a wireless area network as shown in Figure 29.5.

[2]www.fed4fire.eu
[3]https://www.planet-lab.eu
[4]http://open-multinet.info
[5]https://www.w3.org/community/omn

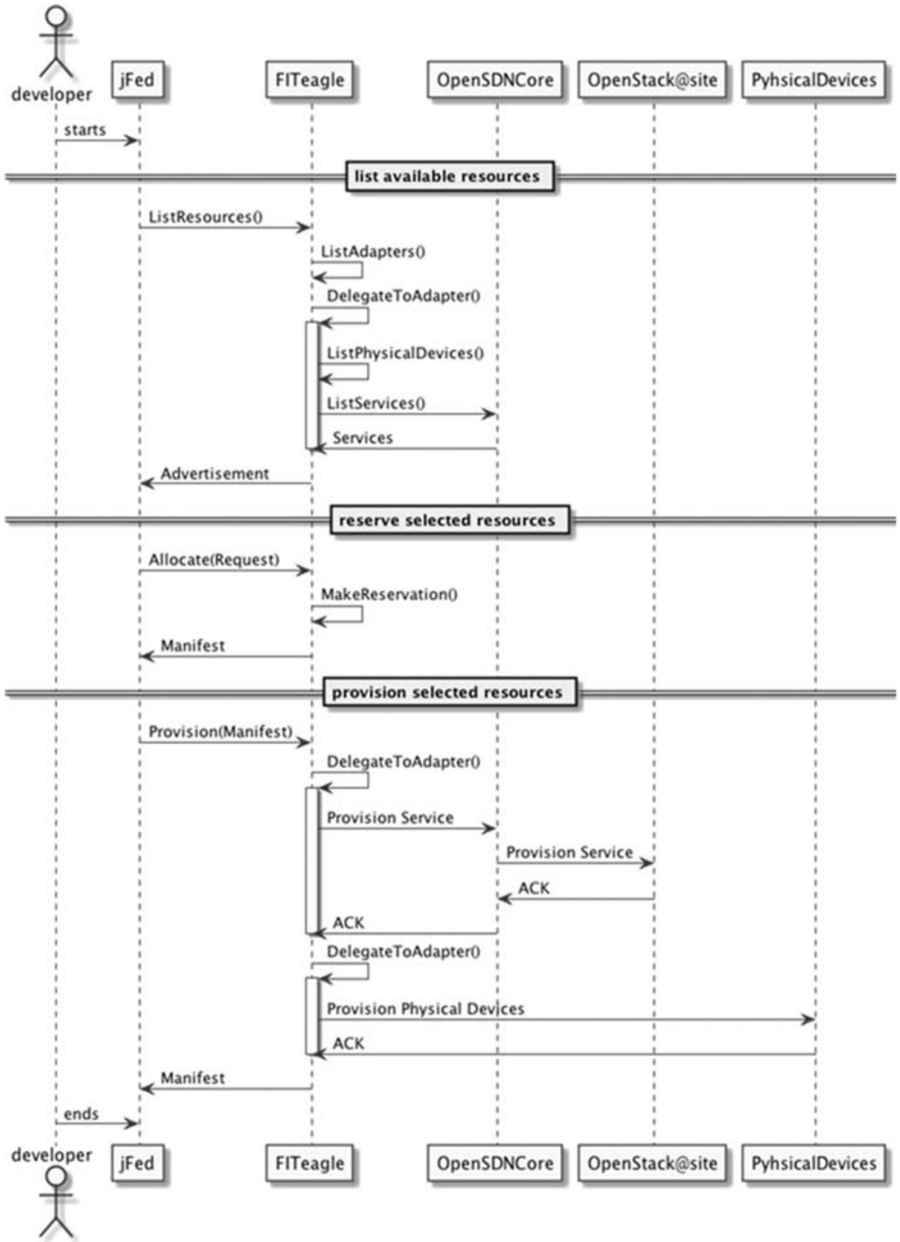

Figure 29.4 Message flow between FITeagle, OpenSDNCore and devices.

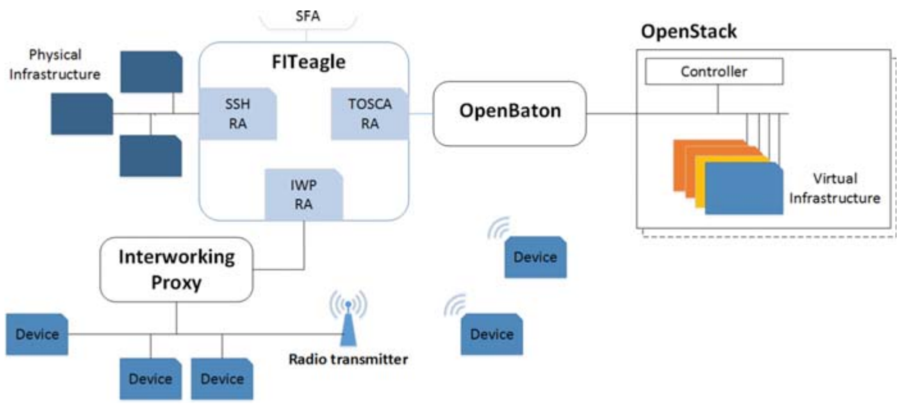

Figure 29.5 FITeagle resource adapters.

- **Secure Shell Resource Adapter (SSH RA)** focuses on creating specific user profiles for physical hosts connected to the federated testbed. It allows experimenters secure access to specific hardware resources using a SSH server and provided SSH keys.

29.5.3 NFV Management and Orchestration (MANO) – OpenBaton

With the increasing acceptance of NFV and SDN technologies, Telco Operators are modifying their traditional network infrastructures in order to provide more flexibility enabling new business opportunities. One major change introduced by those technologies is the capability of deploying on demand different network topologies on top of the same physical infrastructure.

The aim of the European Telecommunications Standards Institute (ETSI) NFV Industry Specification Group (ISG) was to provide a set of guidelines that can be used by different software-based network functions providers for providing their solutions as a service. In particular, the ETSI NFV Management and Orchestration (MANO) domain defines an architecture for managing those VNFs on top of a common NFV Infrastructure. In TRESCIMO, the NFV MANO capabilities have been required specifically for the management of the M2M VNFs on top of the federated testbed. OpenSDNCore, a Fraunhofer FOKUS toolkit, has been selected as NFV Orchestrator for the TRESCIMO testbed. Fraunhofer FOKUS decided to open source its MANO platform (part of the OpenSDNCore project) under the name of OpenBaton. Figure 29.6 shows the OpenBaton architecture.

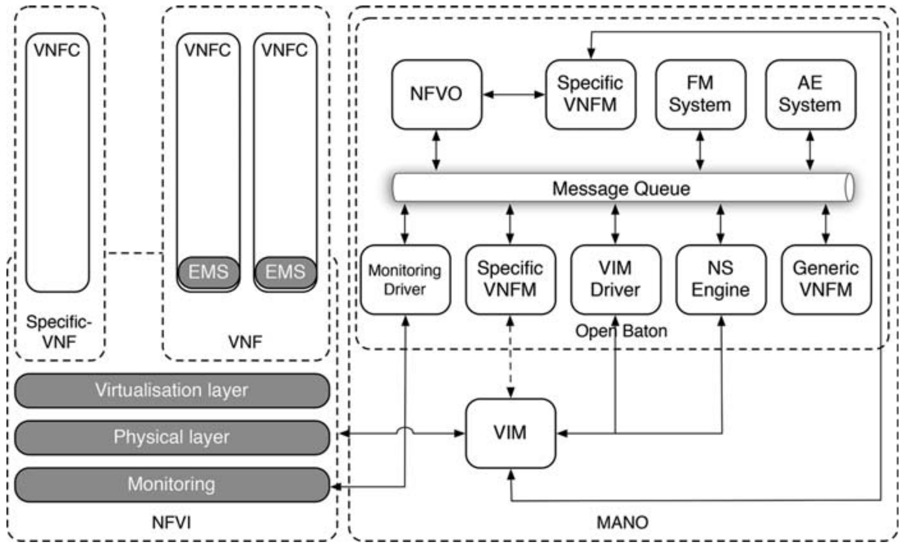

Figure 29.6 OpenBaton functional architecture.

OpenBaton is based on a very modular and extensible architecture. Internally each functional entity is implemented as an independent component, and the communication is based on micro-services principles using a messaging system. The Advanced Message Queuing Protocol (AMQP) is used as messaging protocol between the different components. The main entry point is represented by the Network Function Virtualisation Orchestrator (NFVO) providing a OASIS TOSCA Northbound API which can be consumed by FITeagle for requesting the instantiation of Network Services (NSs) as composition of different M2M VNFs. The NFVO interacts with the Generic Virtual Network Function Manager (VNFM) for managing the lifecycle of those VNFs. The instantiation of the virtual compute and network resources is done via the Virtualised Infrastructure Manager (VIM) Driver. Considering that OpenStack represents the standard de-facto implementation of the VIM, for TRESCIMO has been employed and integrated the OpenStack Plugin[6]. OpenBaton provides the capability of interacting with a multi-site NFV Infrastructure. Basically, each testbed has been registered as an independent Point of Presence (PoP) using the OpenBaton dashboard. Each different VNF can be configured in order to be deployed on a specific PoP accordingly.

[6]https://github.com/openbaton/openstack-plugin

29.5.4 M2M Platform – OpenMTC/Open5GMTC

The M2M platform was used to transport the data from the devices to the SCP or to the applications directly. In fact two toolkits were used. OpenMTC, a reference implementation of the ETSI M2M and the OneM2M standard, was used to transport the data. Open5GMTC, which utilises the OMA Lightweight M2M Standard and CoAP as the transportation protocol, was used to configure the gateway and the applications so that they can register with the correct endpoint and deliver the data to the specified destination.

The OpenMTC framework is separated in a gateway and a backend component. The gateways are registered with the backend and the applications are registered with either the gateway or the backend. The devices can act like an application or a so called Interworking Proxy (IWP) that mediates between the specific devices and the framework. In doing so, the devices working as actuators or sensors will receive a digital representation in the system. The IWP is then an application from the perspective of the gateway or the backend. The applications can access historic data and also subscribe to receive new incoming data. By using access rights, other applications within different operating domains or use cases can be granted permission to view the same data if required.

The Open5GMTC, which is an enhancement of the OpenMTC, is also divided into a server instance and corresponding clients. In the TRESCIMO project the device management (DM) capabilities were used in order to configure some components of the architecture. From Figure 29.7 one can see the used example in the Proof of Concept implementation. Here the M2M DM Server which is a component of Open5GMTC was deployed statically, as well the Active Gate IWP and the DTN Gateway with their connected devices. The emulated devices, the M2M Platform and the SCP were deployed dynamically and configured to establish connections between them. The M2M Platform was connected with the M2M DM Server so that the latter can configure the static instances of the Active Gate IWP and the DTN Gateway to connect to the M2M Platform when it is available.

29.6 Results and/or Achievements

In TRESCIMO the following objectives were achieved: the architecture to fulfil the use cases was defined and successfully setup; and the toolkits were integrated and the various TRESCIMO stack components were deployed within the testbeds and the trials. It was shown that with the incorporation of

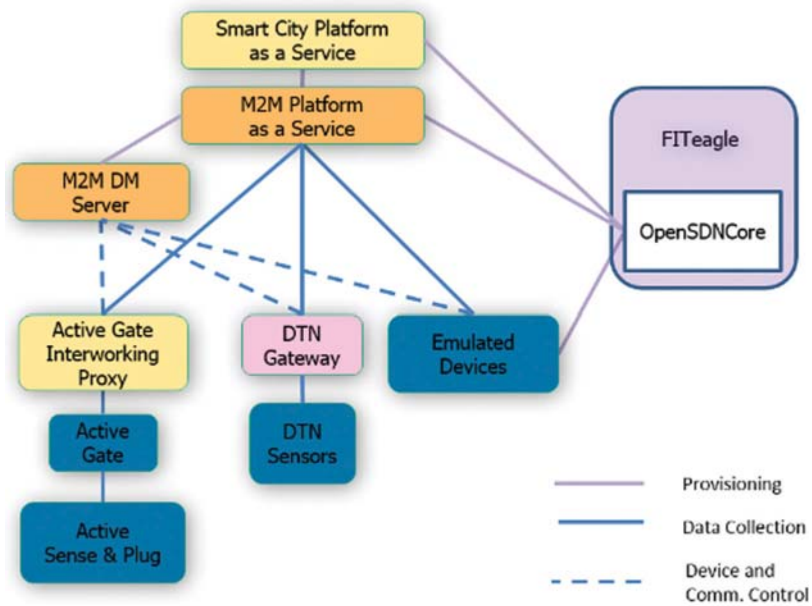

Figure 29.7 Architecture used in the PoC to illustrate the cooperation of the M2M frameworks.

SDN and NFV, the TRESCIMO testbeds were able to provide the necessary environment for IoT experimentation to be carried out.

29.6.1 Integration of the Toolkits

The integration of the various toolkits was an important part of the project as it allowed for the validation of the designed architectural framework. In the second prototype version of the TRESCIMO stack, illustrated in Figure 29.8, one can see all of the integrated components and their interconnection. The M2M Platform is connected on the Southbound interface to various devices either via different Interworking Proxies (IWP) or directly through M2M Gateways. The M2M Platform is linked to the Device Management (DM) Server. On the Northbound interface the M2M Platform interfaces with the SCP which supports heterogeneous applications. These applications can also be directly connected to the M2M Platform bypassing the SCP. On the right seats FITeagle which uses the orchestrator of the OpenSDNCore OpenBaton. The combination is used to deploy and orchestrate the components of the heterogeneous (emulated) devices, the M2M Gateway and Platform, the SCP and the heterogeneous applications.

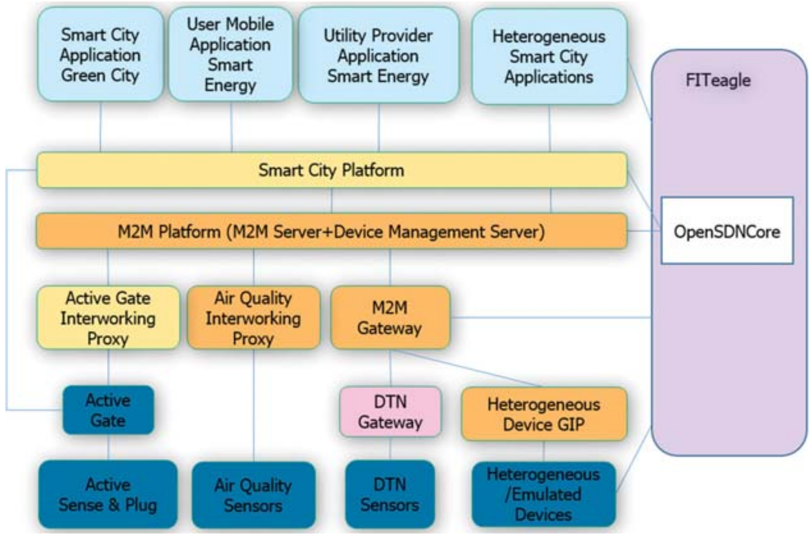

Figure 29.8 Detailed prototype architecture version 2.

The TRESCIMO stack of the components were deployed inside an OpenStack cloud computing environment. Some of these components are statically deployed and thus always available. This was the case for FITeagle, OpenBaton and the M2M DM Server. The integration of the components can be divided into two types: IoT stack components integration and infrastructure or management components integration.

The central element of the IoT architecture was the M2M framework. In order to enable the different devices to connect and communicate with the framework IWPs were developed specific to each device. This allowed for the data generated by the devices to be accessible on the M2M platform by the relevant high-level applications. In order for these applications to access this data, the SCP enabled the discovery of new devices and functionality to subscribe to the data generated by the devices. Although the communication from the devices to the SCP was done by the M2M framework a data model was necessary to link the data to the real world.

The integration of the management part of the architecture was performed by linking the different management tools to each other. The FITeagle framework, which provided the SFA interface to the experimenter, was connected to the OpenBaton VNF orchestrator via a TOSCA-interface. To achieve this a TOSCA-adapter in FITeagle was added and TOSCA was added to the API of OpenBaton. Additionally some RSpecs were added or modified to be able to define the to be deployed topology via the SFA interface. OpenBaton can start VMs inside of OpenStack. These VMs were started using modified images that have a web service installed to listen for the configuration parameters received by OpenBaton. Additionally, service startup and configuration scripts are placed inside the images. These scripts, using the received configuration parameters, allowed for the service to be installed on an instantiated VM at its launch time. To support the flexibility of the IoT architecture stack these scripts were created for all different components like the M2M Platform and the SCP.

With the complete integration of all the components and services, it is possible for an experimenter to deploy a setup of the TRESCIMO architecture stack with several devices configured and also spread over different locations. Due to the endpoints provided by FITeagle, the experimenter could connect to these endpoints and run the experiment.

29.6.2 Smart City Experimentation

The main aim of the TRESCIMO facility is to allow for research activities in the areas of M2M, IoT and Smart Cities. This will require experimenters to be able to provisioning connected M2M nodes and services in order to meet the requirements of a particular experiment. A few of the smart city experimentation use cases, to validate the functionality of the testbed, are presented in the following sections. In addition, the key observations for each use case are discussed.

29.6.2.1 Smart buildings

Connected homes are equipped with various IoT devices and applications that are capable of communicating with remote users [8]. This enables services such as remote control of appliances to be deployed. In the case of TRESCIMO, the possibility of reusing existing smart home devices to provide new services for residents is investigated. This involved using low powered devices to monitor energy consumption and control model appliances. To cope with the limitations of these devices, an M2M application (Interworking Proxy) that abstracts the complexities of the devices and provides data to external applications via an OpenMTC gateway was created. This application was deployed in conjunction with OpenMTC gateways for a number of experimental scenarios. We observed that this approach enabled smart energy applications to utilise data from home automation application without having to know any of the procedures necessary to collect the data. However, these smart energy applications still needed to have knowledge of the data structure used by the Interworking Proxy to store data. Extending the IWPs to utilise a semantic description generator may solve this problem [10].

29.6.2.2 Energy management

The adoption of smart metering technology will enable energy providers to use demand side management (DSM) techniques to monitor and manage energy consumption in homes. In particular, energy providers are interested in implementing demand response (DR) and making consumers more energy aware [6]. This will enable energy providers to send residents requests to reduce current energy consumption based on the state of then national power grid. This technique is used to ensure the stability of the electricity grid. To facilitate DR actions, it is necessary for energy providers and consumers to agree on the appliances available for control and incentives for the consumer [7]. For demand response applications, it is imperative that the energy provider be able to receive real time data on the energy consumption of individual appliances from IoT devices deployed in the home. This data will be used to generate DR requests for residents. We conducted DR experiments within the TRESCIMO facility in order to test the ability of the smart city infrastructure to support time sensitive applications. An energy manager application was created to generate DR messages based on the current state of electricity grid. These messages were delivered to consumers prompting them to reduce their current energy consumption. The developed energy applications were able to send messages and carry out required actions. An example of the messages sent is shown in Figure 29.9. In this example, the residents allows an energy application to switch off devices on their behalf. We discovered that it was possible to implement a privacy aware DR solution by storing consumer information about energy used by specific appliances on a residential M2M gateway.

29.6.2.3 Education

This experiment scenario focuses on enabling university lecturers to provide a laboratory for their classes, thus enabling students to experiment with state of the art technology. The teacher utilises the jFed experimenter client[7] developed in the scope of the Fed4Fire project[8]. This client utilises the SFA interface of the TRESCIMO testbed provided by FITeagle. After successfully

[7]http://jfed.iminds.be/
[8]http://www.fed4fire.eu

Figure 29.9 Example of DR related messages sent to a resident.

logging in with a valid X509 certificate the teacher is able to create an M2M topology using the graphical interface and by providing an RSpec detailing the required infrastructure. Figure 29.10 shows a sample topology created using the jFed client. After successfully creating the topology, the lecturer then provides the students details of the accessible resources and their endpoints. In TRESCIMO, it was possible for the lecturer to create topologies that use resources from multiple testbed sites. For the topology shown in Figure 29.10, energy measuring devices and an M2M gateway was deployed at UCT in South Africa, while the M2M server was deployed at TUB in Germany. Students were able to access both the gateway and server within an acceptable level of latency.

29.7 Discussions and Conclusions

We have shown that an intercontinental Infrastructure-as-a-service IoT testbed may be setup consisting of multiple sites connected via VPN connection, capable of supporting several different experimental use cases. The testbeds

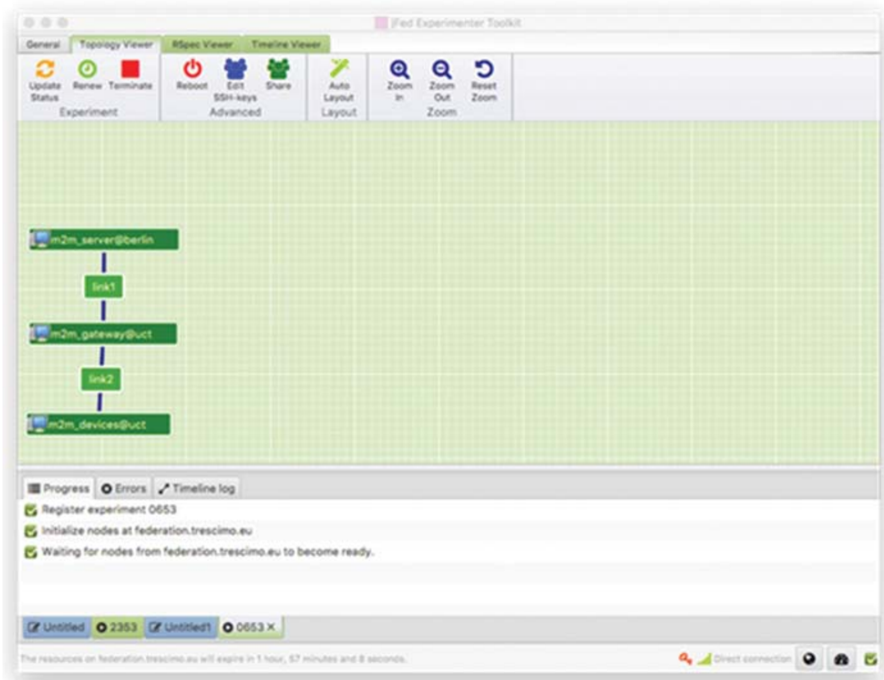

Figure 29.10 Example of an experimental topology created using jFed client.

are federated by providing the FIRE SFA interface. This enables FIRE and GENI experimenters to utilise resources deployed at any of the three testbed sites.

Further short-term work includes the complete replacement of the old OpenSDNCore orchestrator by OpenBaton and the integration of more generic devices to support use cases in the areas of healthcare and security. Based on feedback provided by experimenters, we are investigating the possibility of providing some of the applications developed during the smart city experimentation process, as on demand services for experimenters. This will hopefully ease the process of getting started with smart city experimentation.

According to FIRE's definition of testbed federation, each testbed site should host a FITeagle instance that supports FIRE tools such as the jFed client. In TRESCIMO, the testbeds are interconnected via a Virtual Private Network such that they can be viewed, in FIRE terms, as one distributed multi-site testbed. FIRE experimenters can thus access TRESCIMO testbed resources by first communicating with the FITeagle instance hosted at the

TUB testbed site. The long-term objectives are for each individual testbed site to be independently federated. That is, a FITeagle and OpenBaton instance will be deployed at each location and enable instances to communicate with each other without the need of the established VPN connection.

Acknowledgements

The TRESCIMO project has received funding from the European Union's 7th Programme for research, technological development and demonstration under grant agreement no. 611745, as well as from the South African Department of Science and Technology under financial assistance agreement DST/CON 0247/2013. The authors thank all the TRESCIMO partners for their support in this work.

References

[1] J. M. Barrionuevo, P. Berrone, and J. E. Ricart. Smart Cities, Sustainable Progress. *IESE Insight*, (14):50–57, Third Quarter 2012.

[2] Bartosz Belter, Juan Rodriguez Martinez, Jos Ignacio Aznar, and Jordi Ferrer Riera. The {GEYSERS} optical testbed: A platform for the integration, validation and demonstration of cloud-based infrastructure services. *Computer Networks*, 61:197–216, 2014. Special issue on Future Internet Testbeds Part I.

[3] M. Campanella and F. Farina. The FEDERICA infrastructure and experience. *Computer Networks*, 61:176–183, 2014.

[4] L. Coetzee, A. Smith, A. E. Rubalcava, A. A. Corici, T. Magedanz, R. Steinke, M. Catalan, J. Paradells, H. Madhoo, T. Willemse, J. Mwangama, N. Mukudu, N. Ventura, M. Barros, and A. Gavras. TRESCIMO: European Union and South African Smart City contextual dimensions. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 770–776, Dec 2015.

[5] DESA United Nations. *United Nations, Department of Economic and Social Affairs, Population Division: World Urbanization Prospects, the 2014 Revision: Highlights (ST/ESA/SER.A/352)*. UN publications, 2014.

[6] Karen Ehrhardt-Martinez, Kat A. Donnelly, and John A. Laitner. Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities, Report E105. Technical report, American Council for an Energy – Efficient Economy, 2010.

[7] ETSI. TR 102 691 V1.1.1 (2010-05) Machine-to-Machine communications (M2M); Smart Metering Use Cases. Technical report, ETSI, May 2010.

[8] ETSI. TR 102 857 V1.1.1 (2013-08) Machine-to-Machine communications (M2M); Use Cases of M2M applications for Connected Consumer. Technical report, ETSI, August 2013.

[9] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, 49(11):58–67, November 2011.

[10] A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano. Assisting iot projects and developers in designing interoperable semantic web of things applications. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 659–666, Dec 2015.

[11] R. R. Harmon, E. G. Castro-Leon, and S. Bhide. Smart cities and the Internet of Things. In *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, pages 485–494, August 2015.

[12] G. Klyne, J. J. Carroll, and B. McBride. *Resource description framework (RDF): Concepts and abstract syntax, W3C, W3C Recommendation*. 2004.

[13] D. Palma and T. Spatzier. *Topology and orchestration specification for cloud applications (TOSCA) version 1, Nov*. 2013.

[14] L. Peterson, R. Ricci, A. Falk, and J. Chase. Slice-based federation architecture. Technical report, GENI, 2010.

[15] Luis Sanchez, Luis Muoz, Jose Antonio Galache, Pablo Sotres, Juan R. Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, and Dennis Pfisterer. SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, 61: 217–238, 2014.

[16] Alexander Willner, Daniel Nehls, and Thomas Magedanz. FITeagle: A Semantic Testbed Management Framework. In *Proceedings of the 10th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, pages 1–6, Vancouver, Aug 2015. ACM.