Abstract
A key concern in the Internet of Things (IoT) has been the integration of mundane objects in the Internet. Although increasingly interconnected, the IoT ecosystem is largely industry-centered. This leads to the creation of limited and incompatible services disempowering users by hampering their participation. In this thesis, we address this issue by empowering users to create, personalize, and distribute services in the IoT ecosystem. We define a general framework for user empowerment relying on the concepts of People, Places, Things and Applications and study their interactions. These concepts are used as design elements to implement an IoT Service Platform. The general framework, the reference implementation and the evaluation provide the design guidelines for building IoT service platforms enabling user empowerment in the IoT.
User Empowerment in the Internet of Things

THÈSE
présentée à la Faculté des sciences économiques et sociales
de l'Université de Genève
par
Dejan Munjin
sous la direction de
prof. Jean-Henry Morin
pour l'obtention du grade de
Docteur ès sciences économiques et sociales
mention systèmes d'information

Membres du jury de thèse:

M. Michel LÉONARD, Professeur, président du jury
M. Dimitri KONSTANTAS, Professeur
M. Jean-Henry MORIN, Professeur, Directeur de thèse
M. Nicolas NOVA, Professeur, Juré externe
User Empowerment in the Internet of Things

Thèse no 810
Genève, 05 Juillet 2013
La Faculté des sciences économiques et sociales, sur préavis du jury, a autorisé l'impression de la présente thèse, sans entendre, par là, émettre aucune opinion sur les propositions qui s'y trouvent énoncées et qui n'engagent que la responsabilité de leur auteur.

Genève, le 05 Juillet 2013

Le doyen
Bernard MORARD

Impression d'après le manuscrit de l'auteur
# Table of Contents

Table of Contents ........................................................................................................ iv
Tables and Figures ........................................................................................................ vi
List of Acronyms .......................................................................................................... vii
Résumé ........................................................................................................................ ix
Abstract ......................................................................................................................... xii
Acknowledgement ....................................................................................................... xiv

Chapter 1.  Introduction .................................................................................. 15
  1.1  Industry-Centered Internet of Things .................................................. 15
  1.2  Motivation and Problem Statement .................................................... 18
      1.2.1  Motivation .............................................................................. 18
      1.2.2  Application Scenario .............................................................. 19
      1.2.3  Problem Statement ................................................................. 20
      1.2.4  Research question .................................................................. 20
  1.3  Methodology and Approach ................................................................. 20
  1.4  Objectives .............................................................................................. 22
  1.5  Thesis Outline ....................................................................................... 22

Chapter 2.  State of the Art ........................................................................... 24
  2.1  Internet of Things ............................................................................... 24
  2.2  User Empowerment in the Internet of Things ...................................... 31
  2.3  Summary .............................................................................................. 40

Chapter 3.  Requirements for User Empowerment in Internet of Things .... 41
  3.1  Internet of Things Hardware Providers ............................................. 42
  3.2  Internet of Things Users ..................................................................... 44
  3.3  Internet of Things Developers ............................................................ 45
  3.4  Internet of Things Marketplaces ......................................................... 45
  3.5  Summary .............................................................................................. 46

Chapter 4.  Framework for Empowerment in the Internet of Things ........ 47
  4.1  Defining the Framework ..................................................................... 47
  4.2  People in the Internet of Things .......................................................... 47
      4.2.1  Virtual Identities ..................................................................... 47
      4.2.2  Online Social Networks ......................................................... 49
  4.3  Places in the Internet of Things .......................................................... 50
      4.3.1  Physical Places ...................................................................... 50
      4.3.2  Virtual Places ........................................................................ 53
      4.3.3  Hybrid Virtual Places ............................................................ 55
  4.4  Things in the Internet of Things ........................................................... 56
  4.5  Marketplaces and Applications in the Internet of Things ................. 59
  4.6  Designing the Framework ................................................................... 63
      4.6.1  Defining the model of *People, Places, Things and Applications* 68
      4.6.2  Design .................................................................................... 69
Chapter 5. Platform Prototype .................................................................. 81
  5.1 Scenarios .................................................................................. 81
    5.1.1 The Model of People, Places, Things, and Applications in the working Scenarios ....................................................................................... 81
  5.2 System Requirements .................................................................. 85
  5.3 Implementation .......................................................................... 88
    5.3.1 Architectural Strategies ...................................................... 88
    5.3.2 System Architecture ......................................................... 90
  5.4 Prototype Evaluation and Results ................................................. 94
    5.4.1 Experimental ...................................................................... 96
    5.4.2 Demonstration .................................................................... 96
    5.4.3 Description ......................................................................... 97
    5.4.4 Analysis ............................................................................. 98
  5.5 Summary ................................................................................... 99

Chapter 6. Revisiting the Framework and Generalization ...................... 100
  6.1 Design Invariants ....................................................................... 100
  6.2 Elements of User Empowerment .................................................. 104
    6.2.1 End Users Empowerment .................................................. 104
    6.2.2 Empowerment of Developers .......................................... 106
  6.3 Summary ................................................................................... 107

Chapter 7. Conclusions and Outlook .................................................. 108
  7.1 Contribution ............................................................................. 108
  7.2 Future Work ............................................................................. 110
    Bootstrapping Application Markets ........................................... 110
    Deploying “Intelligent” Services ................................................. 110
    Towards a Social Networks of Things ........................................ 111

References .............................................................................................. 114

Appendix ................................................................................................. 120
  1. Informational Schema of Implemented Permanent Storage .......... 120
  2. Use Cases – Illustrative Example .................................................. 121
Tables and Figures

Tables
Table 1: Elements of a request to IoT service platform ........................................ 92
Table 2: Elements of general framework and IoT service platform that empower end users .............................................................. 106
Table 3: Elements of general framework and IoT service platform that empower developers .............................................................. 107

Figures
Figure 1: Vertical Silos created by IoT Service Providers .................................. 16
Figure 2: User centered approach for building the Internet of Things ............ 18
Figure 3: The research cycle followed in this thesis ....................................... 21
Figure 3: Auto-ID labs research program ......................................................... 25
Figure 4: Open and Closed loops for Auto-ID Systems .................................. 26
Figure 5: Architecture of EBBITS platform .................................................... 30
Figure 6: Arduino wireless sensors network ................................................... 32
Figure 7: An exhibition hall in Cool Town project ......................................... 33
Figure 8: Layers of the Web of Things application architecture ..................... 35
Figure 9: Architecture of Data Brokerage Platforms ...................................... 36
Figure 10: Connected Marketplaces Supporting User Innovation in the IoT Space ............................................................................. 38
Figure 11: Stakeholders and empowerment of communities for innovation in IoT ............................................................................. 42
Figure 12: Virtual proxies of connected objects ............................................. 44
Figure 13: Online social relationships defined and managed by People .......... 50
Figure 14: The role of Physical Places and online social networks in the IoT ... 52
Figure 15: Virtual Places in IoT ..................................................................... 54
Figure 16: Mapping physical to virtual places and creating Hybrid Virtual Place 56
Figure 17: People, Places, and Things ......................................................... 59
Figure 18: Application development in IoT vs. smart phone ......................... 60
Figure 19: Two-sided network for the IoT ...................................................... 61
Figure 20: Applications registered in Virtual Places ...................................... 62
Figure 21: Framework for User Empowerment and its Elements .................. 63
Figure 22: General Framework: Instances of People, Places, Things, and Applications ................................................................. 67
Figure 23: Illustrative Scenario for General Framework ............................... 77
Figure 24: Schema of System Requirements ............................................... 85
Figure 25: System Architecture of IoT service platform .............................. 91
Figure 26: Scenario instance on distributed IoT platforms ......................... 101
Figure 27: End user empowerment to choose services in IoT ...................... 104
Figure 28: Proposal of a “Thing Book” ....................................................... 111
List of Acronyms

ACID - Atomicity, Consistency, Isolation, and Durability
ACL - Access Control List
API - Application programming interface
CoAP - Constrained Application Protocol
CSV - Comma-separated values
DHCP - Dynamic Host Configuration Protocol
DIYS - Do It Your Self
DNS - Domain Name System
EPC - Electronic Product Codes
HTTP - Hypertext Transfer Protocol
IERC - European Research Cluster on the Internet of Things
IoT - Internet of Things
IoT-A - Internet of Things Architecture
IPv6 - Internet Protocol version 6
IT - Information Technology
JSON - JavaScript Object Notation
M2M - Machine-to-machine
MQTT - Message Queue Telemetry Transport Protocol
NFC - Near Field Communication
NoSQL - Not Only Structured Query Language
OAuth - Open authentication protocol
ONS - Object Naming Service
Open-ID - Open authentication standard
PHP - PHP Hypertext Preprocessor
PML - Product Markup Language
RDBMS - Relational Database Management System
REST - Representational State Transfer
RFID - Radio Frequency Identification
SaaS - Software as a Service
SoA - Service oriented Architecture
SSL - Secure Sockets Layer
URI - Uniform resource identifier
Web 2.0 - Cumulative changes in the ways software developers and end-users use the World Wide Web
WWW - World Wide Web
XML - Extensible Markup Language
Résumé

Une préoccupation centrale dans le domaine de l'Internet des Objets (IdO) a été l'intégration des objets courants dans l'Internet pour permettre aux systèmes informatiques de percevoir et mesurer le monde à travers eux. L'idée est de permettre aux ordinateurs et appareils de recueillir des données sans intervention humaine, d'interagir avec leur environnement et entre eux. L'IdO est une vision où tout Objet devient « connecté », ceci pouvant aller des nouveaux appareils tels que les téléviseurs intelligents, les smartphones et les vêtements en passant par des objets plus traditionnels comme les réveils, les réfrigérateurs ou encore des monuments. Dès lors, ceux-ci peuvent être utilisés pour recueillir des données du monde physique. Les technologies d'informatique embarquées, la connectivité réseau, les étiquettes-radio et les codes-barres permettent à tous nos objets environnants de devenir des Objets fournissant des données aux systèmes informatiques. La promesse de l'IdO est que sur la base de ces données, nous pourrons mieux comprendre le monde physique qui nous entoure et ainsi de créer de nouveaux services. Les domaines d'application sont presque infinis. Ils peuvent recouvrir des domaines tels que la planification urbaine intelligente, l'environnement durable, la gestion de l'énergie intelligente et bien d'autres. Par conséquent, avec une meilleure compréhension de notre environnement, nous pouvons créer de nouveaux services permettant d'améliorer la façon dont nous vivons et finalement contribuer à une société plus durable.

La vision de l'IdO a initialement exploré la façon de d'interconnecter les objets. Cependant, alors que l'objectif était de fournir une infrastructure pour la création de services à l'échelle d'Internet, les services créés jusqu'à présent sont nombreux, fragmentés et essentiellement difficiles à intégrer dans la mesure où ils sont enfermés dans des silos centrés fournisseurs. Par conséquent, la principale raison du manque de services disponibles est liée à une approche de construction de l'IdO centrée sur les fournisseurs d'Objets. Cette approche néglige les utilisateurs comme partie prenante à la création de l'écosystème IdO. Ainsi, les fournisseurs créent des situations de consommateurs captifs en ne facilitant pas l'interopérabilité. De telles approches réduisent l'autonomie et le pouvoir des utilisateurs pour personnaliser leurs propres services basés sur des infrastructures IdO hétérogènes.

Les développements récents en matière de matériel et de logiciels libres ont permis une autonomisation des utilisateurs notamment afin de construire leurs propres écosystèmes IdO. Avec les matériels, les logiciels, les librairies et les outils du libre, les utilisateurs peuvent connecter leurs appareils pour créer de nouveaux services. Malheureusement, cette approche n'autonomise que les utilisateurs techniques à créer leurs propres Objets et services IdO. Cette approche adresse la question de l'autonomisation des utilisateurs leurs
User Empowerment in the Internet of Things

permettant de créer des contenus numériques, de connecter des objets, et donc jouer un rôle actif dans l’innovation de services IdO.

Cette thèse propose un cadre général pour l’autonomisation des utilisateurs de l’IdO. Plus précisément, cela consiste à autonomiser les utilisateurs dans leurs capacités à participer à l’écosystème IdO en créant, personnalisant et distribuant des services. Nous identifions trois parties prenantes de l’IdO: les fournisseurs de matériels, les développeurs et les utilisateurs finaux. Leur engagement respectif est nécessaire afin de créer un écosystème favorisant l’autonomisation des utilisateurs. Bien que l’engagement des fournisseurs de matériels soit une condition préalable pour établir une infrastructure IdO, notre travail se concentre sur l’utilisateur et plus particulièrement sur deux groupes d’utilisateurs: les développeurs en tant qu’utilisateurs d’infrastructures IdO et les utilisateurs finaux. Notre objectif est d’identifier une base commune pour l’autonomisation des utilisateurs leurs permettant d’apporter la contribution la plus efficace possible. L’analyse des technologies IdO et les parties prenantes permet de conclure que cette base commune peut être réalisée grâce à une approche de type plateforme de services IdO. Ces plateformes devraient être conçues comme des abstractions permettant de recueillir et stocker des données à partir des objets IdO indépendamment des fournisseurs d’objets. Elles stockent les données recueillies du monde physique et fournissent un support pour la représentation virtuelle des objets (virtualisation d’objets). Toutefois, les plateformes actuelles doivent être repensés (conception) sur la base de modèles centré utilisateur afin de permettre l’autonomisation des utilisateurs.

Pour définir les propriétés nécessaires à la conception d’une plateforme de services IdO, nous proposons un cadre incluant quatre éléments constituant le cœur de l’approche: Les gens (People), les places (Places), les objets (Things) et les applications. Les « gens » sont les utilisateurs finaux et les développeurs. Nous définissons comment une plateforme de service IdO gère la représentation virtuelle de ceux-ci et de leurs relations sociales en-ligne. Les « places » sont les environnements physiques et virtuels qui fournissent le contexte de la personnalisation des services. Les « objets » sont les objets connectés ayant leurs représentations virtuelles sur la plateforme de services IdO. Les « applications » sont des services créés par les développeurs. Notre cadre général analyse ces concepts et les met en œuvre pour la conception de plateformes de services IdO favorisant l’autonomisation des utilisateurs.

Ce cadre conceptuel est ensuite utilisé pour la conception et la mise en œuvre d’une telle plateforme sous la forme d’un prototype. L’évaluation des propriétés et des composants de cette plateforme est utilisée pour revoir et améliorer le cadre conceptuel.
Les principales contributions de cette thèse sont le cadre général d’autonomisation des utilisateurs dans l’IdO et la conception d’une implémentation de référence satisfaisant ce cadre. Finalement, ce travail ouvre de nouvelles perspectives de recherche et des opportunités concrètes dans des domaines tels que « l’autonomisation des objets » et les « réseaux sociaux d’objets ».
Abstract

A central concern in the area of Internet of Things (IoT) has been the integration of mundane objects in the Internet to enable computing systems to sense the world through them. The idea is to allow computers and devices to gather data without human intervention, to interact with their environment and among them. IoT is a vision where any “Internet enabled” object becomes a connected “Thing”. “Things”, ranging from new appliances such as smart TVs, smart phones and clothes to more casual objects such as alarm clocks, fridges and monuments can be used to collect data from the physical world. Embedded computing technologies, network connectivity, radio-tags, and barcodes enable all our surrounding objects to become Things and provide data to networked computer systems. The promise is that based on these data it will help better understand the physical world around us in order to create new services. Application domains are almost infinite. They can be for smart urban planning, sustainable environment, smart energy and many others. Therefore, with better insights on our environment we can create new services allowing to enhance the way we live and eventually contribute to a more sustainable society.

Internet of Things vision initially explores the way to interconnect objects. However, while the aim was to provide an infrastructure for the creation of services on Internet scale, the services created so far are many, fragmented and essentially live in hard to integrate and interoperate silos. Therefore, the main reason for the lack of services is an industry-centered approach to building the Internet of Things. It neglects service users as contributors for the creation of an Internet of Things ecosystem. With an industry-centered approach, service providers create vendor lock-ins by providing Things and services that are not compatible with other service providers. This approach reduces the power of service users to personalize services on top of heterogeneous IoT infrastructures.

Recent developments in open source hardware and software has led to empowering users to build their own IoT ecosystems. With open hardware, software, libraries and tools, users can extend home appliances with connecting hardware and software to create new services. Unfortunately, this approach empowers only technical users willing to spend time to create their own Internet enabled objects and services. This approach addresses the issue of empowering users with the ability to create digital content, build Internet enabled hardware, and therefore play an interesting role in IoT service innovation.

This thesis proposes a general framework for user empowerment in the Internet of Things. Specifically, it aims at empowering users to shape the Internet of Things ecosystem by creating, personalizing and distributing services. We first identify three groups of Internet of Things stakeholders: hardware providers,
developers, and end users. Their respective engagement is required in order to create an ecosystem that will foster user empowerment. Although the engagement of hardware providers is a prerequisite to establish enabling infrastructure for the Internet of Things, our work is user centric and we focus on two groups of users: developers as users of Internet enabled hardware that should be empowered to create and distribute new services and end users that should be empowered to personalize services. Our goal is to identify the common ground for user empowerment in order to make their contribution the most effective. The analysis of enabling technologies and stakeholders leads to the conclusion that this common ground can be achieved through an Internet of Things service platform. These platforms should be designed as networked computer hardware and software that collect and store data from Internet enabled objects i.e. Things. They store data collected by Things from the physical world and they provide support for virtual representation of Things. However, current platforms must be redesigned, following user-centered models, in order to foster user empowerment.

In order to define design properties of an Internet of Things service platform, our framework defines four elements that are mediated by the platform: People, Places, Things, and Applications. People are end users and developers. We define how an IoT service platform should manage their virtual counterparts and their online social relationships. Places are physical and virtual environments that provide context for personalization of services. Things are Internet enabled hardware devices or objects with radio tags that have their virtual counterparts on the Internet of Things service platform. Applications are software applications as services created by developers. Our general framework analyses these concepts and combines them in design elements for Internet of Things service platforms favoring user empowerment.

This conceptual framework is then used to design and implement an Internet of Things service platform empowering the users. The evaluation of the platform properties and its components are used to further revise and enhance the framework.

The key contributions of this thesis are the general framework and the design of Internet of Things service platforms that foster user empowerment. Finally, this research opens new research directions and industrial opportunities in areas such as empowerment of Things and Social Networks of Things.
Acknowledgement

Foremost, I would like to express my sincere gratitude to my family and cousins Arnaud Bresson, Zorica Živković-Bresson, Vera Živković-Mignerey, Radojka Živković, Jovan Živković, Jovan Munjin, and Zora Bačo for their help, encouragement and support during this magnificent adventure of studies at the University of Geneva.

My sincere thanks also go to my advisor prof. Jean-Henry Morin for offering me an excellent guidance and support.

Besides my advisor, I would like to thank the rest of my thesis committee: prof. Michel Léonard, prof. Dimitri Konstantas, and prof. Nicolas Nova for their encouragement and insightful comments.

I thank Dr. Dominique Couturier for his helpful advises in statistical analysis and extraordinary ideas for home automation technologies.

I am grateful to Ramona Westermann for being at my side for most of the time during this thesis.

I thank my colleagues from Institute of Services Science, CUI, and HEC for the stimulating discussions, for all the work and project we have done together, and for all the fun we have had in the last years.

Last but not the least, I would like to thank all the colleagues from IoT communities for enriching discussions during conferences and workshops.

Thank you.
Chapter 1. Introduction

1.1 Industry-Centered Internet of Things

Today, the Internet is a medium for exchange of digital information between people, computers, and smart devices. It hosts billions of exchanges and connections. This momentum will inevitably continue: more smart devices and everyday objects are connected to the Internet, with virtual identities. Market estimates predict an increasing number of everyday objects connected to the Internet that will reach more than 50 billion over the next decade [1]. The networked interconnection of everyday objects and people forms the Internet of Things (IoT) and brings denser exchange of digital information. In the IoT context, the real and the virtual worlds are converging and creating smart environments that have the potential to make energy, transport, cities and many other areas more intelligent and efficient hence the natural connection with the area of sustainable development. However, since the term Internet of Things was coined by Auto-id labs [2], the vision has been mainly focused on the industrial sector and its business development. The enabling technologies were intended to automate the product identification and allow firms to optimize their logistic and supply chains. Every product would embed a unique identifier and could be identified using radio frequency identifiers (RFID), near field communication (NFC), electronic product codes (EPC) or other tags at any stage of a product’s lifecycle: from its production to its disposal. Although the IoT vision evolved, encompassing nowadays smart technologies from the field of Ubiquitous Computing remains focused on industry. The industry trend still focuses on creating proprietary standards for enabling technologies. As a result, it creates and personalizes services by service providers based on proprietary solutions. Consequently, this approach results in vertical silos of business solutions as shown in Figure 1.

The industry approach for creating services was developed in order to optimize supply chains during the manufacturing and sale of industrial products. For example companies use RFID tags and sensors to provide their customers with real time tracking of purchased products [3]. Currently IoT technologies are improved enough to overtake the threshold of industry and supply chains and enter into our everyday lives. However, research on the IoT often follows an industry-centered vision. For example, current projects participating in European Research Cluster on the Internet of Things (IERC) [4] are developing network architecture and business-based services. The role of IERC is to coordinate European research activities in the IoT.
It gathers projects, organizations, and companies under a common IoT vision and common research agendas. We can observe from the list of projects [5] participating in IERC that their focus is on building network architecture and protocols (e.g. IoT-A Internet of Things Architecture [6]) or integrating the IoT with the business services (e.g. EBBITS Enabling the Business-Based Internet of Things and Services [7]).

Although such industrial approaches can create new valuable services for users, they also impose several constraints: the user is considered only as a service consumer, possibilities to personalize services depend on each service provider, and the control of connected objects is limited in terms of interoperability between service providers’ systems and business models. Moreover, this approach also imposes a model of management of user’s personal information focused on the interests of service providers. In fact, in order to personalize services for its users, each service provider is supposed to store and manage users’ personal information thus resulting in loss of the user’s privacy. Hoffman et al. in “Privacy-Enhanced Personalization in Ambient Environments” [8] discuss these issues and suggest a framework for services personalization. Their work shows that there is a tradeoff between privacy and services
personalization. The more user personal information is available to service providers, the more user privacy is at risk. Conversely, service can be more personalized by service provider. The problem becomes even tougher in IoT, when the service needs to be interoperable between different service providers: they need to exchange or trade personal information about users resulting in even more decreased users’ control over their personal information and connected objects.

The industry-centered approach with vertical silos is not the best suitable model for the IoT ecosystem. With its vertical silos, it imposes vendor lock-ins on IoT services and neglects the importance of user participation. In order to break the industry silos and shift this trend towards a more open IoT, several research works suggest users’ empowerment to actively participate by building the IoT ecosystem. F. Michahelles [9] puts forward a series of arguments to support the hypothesis that user empowerment in IoT is a key element for technology adoption and uptake. He observed the history and the uptake of the Internet and the Web, and states that the IoT should follow the same path. Accordingly, users in IoT should be empowered to participate in innovation of IoT technologies, to develop software applications, and to generate digital content. However, user empowerment in IoT has received very little attention from the research community. The term “empowerment” and the elements of user empowerment in IoT have not been studied under a unified framework.

Several challenges have been identified in order to empower users in the IoT ecosystem for the promotion of open innovation and technology adoption (e.g. Michahelles F. [9], G. Kortuem and F. Kawsar [10]). For the purpose of this work, we identified the following challenges: first, to empower software developers to cope with diversity of devices and objects in the IoT. Diversity of hardware requires developers with diverse backgrounds. The challenge here is the simplification of the task of developing software applications by defining a common platform for heterogeneous hardware. The second challenge is to empower users to personalize devices and services in the IoT. The number and diversity of ubiquitous devices in IoT calls on the users to dedicate time, effort, and sometimes technical knowledge in order to personalize IoT devices and services. The challenge is to simplify the task for IoT services personalization while preserving diversity in suggested services. The third challenge is to empower users’ communities and software developers to innovate in service creation. The challenge here is to recommend the design of a unified platform for the distribution of user created services.

This thesis attempts to contribute to the progress of user empowerment in IoT by filling the gap between the industry-centered and user-centered approaches. Our aim is to identify which elements can empower users in order to personalize services and create new services in the IoT context.
1.2 Motivation and Problem Statement

1.2.1 Motivation
The Internet, Web 2.0, and the smartphone ecosystem demonstrate a different approach from that followed by industry-centered IoT. Namely, in terms of open standards and open platforms that can be beneficial for all involved stakeholders. It is widely accepted that the difference between service users and service developers is blending with the increased access to online information and education. Nowadays, user-generated content, open innovation, and crowdsourcing are the new models of society and economy demonstrating the importance and value of co-creation with users. Users are not only participating as service consumers but as active service developers as well. The term user *empowerment* emerges more frequently in research works arguing that users should be more involved in order to solve the above-mentioned shortcomings of the industry-centered IoT. It has been argued that the more users are involved in innovation, the more technology and services will be adapted to their needs [9].

Open platforms and protocols, cloud services, development tools, and open hardware are at the heart of user participation in IoT. An open ecosystem, accessible for users led innovation empowers them with the ability to shape their own IoT ecosystem. However, open technologies, standards, and communication protocols are not the only factors that empower users to participate. The approach of open IoT focuses mainly on technical users that are building IoT services as personal solutions. Figure 2 shows user-centered approach for the building and personalization of the IoT ecosystem.

![User centered approach for building the Internet of Things](image)

In this approach, users are supposed to have enough knowledge, time, and motivation in order to build their own IoT solutions. The user’s task is to assemble IoT enabled hardware, connect it to IoT computer platforms, and program the IoT applications. However, this approach does not allow users to easily reuse and distribute the services that they create. Our overview of IoT
platforms shows that there is a plethora of available software libraries for technical users but a lack of software applications [11].

In order to empower users and communities to effectively shape IoT technology and services we need to ensure that they can effectively communicate and distribute their innovation artifacts. In analogy to the application stores as distribution channels for user created applications (i.e. user led innovation) in the smart phone ecosystem, a model of connected marketplaces has been suggested in order to empower users for the dissemination of innovation artifacts in the IoT [10]. Therefore, users are not only the service consumers; they can cluster in communities that are empowered to shape the IoT technology and to build the services influencing the upcoming IoT ecosystem.

This thesis focuses on user empowerment in the IoT. User empowerment in IoT has been mentioned as user generated digital content, user participation, and user led innovation [9]. We focus here on the aspects of power given to developers and users to create, share, and personalize the services in IoT.

1.2.2 Application Scenario

Ideally, developers should be empowered to create and distribute new IoT applications and services. They should have tools simplifying development of services relying on heterogeneous IoT hardware. End users in IoT should be empowered to control the provided services fully. They should have the power to control interactions between their connected objects, to share digital information, and personalize their services.

For example, let us define a working scenario as follows: you buy an Internet enabled home appliance and bring it home. You connect this appliance to the Internet and it uploads the usage information to a computer infrastructure on the network. You want to share this information with some of your friends on your online social network. Sharing of usage information allows you to get the opinions and recommendations related to your home appliance from these friends. Your home appliance also becomes more “social” and helps you give recommendations to your friends. In addition, you would like to use a software application that would analyze the usage of your home appliance and similar appliances of your friends and recommend new social activities personalized for you. However, although you use similar devices at work, you do not want your colleagues or your entire online social network to know what you do at home.

This scenario defines the basic tasks of an IoT user: connect the hardware in a plug and play way, define online social context for sharing of digital information, and personalize services with a software application. We will reuse this scenario for defining system requirements for IoT platforms.
1.2.3 Problem Statement

IoT research has not addressed user empowerment considering developers and service users. The aim of this thesis is to suggest a general framework for user empowerment in order to identify the aspects of power granted to the users, and help them create and personalize IoT services.

IoT technologies imply disruptive changes in business and society [12], consequently user empowerment is an interdisciplinary problem. In this thesis we apply a design science approach [13] to address both the social issues of empowerment and the technological issues that act as barriers for large scale innovation of new services on top of IoT technologies. More specifically, the problem is to define elements of user empowerment in IoT and to suggest new design guidelines for IoT platforms enabling user empowerment for the creation and use of IoT services.

1.2.4 Research question

In this research, we intend to study and design a general framework based on existing literature and current industry trends. The framework will address the most important aspects of user empowerment in IoT. The concepts defined in the framework will afterwards be used in a model of IoT software platform favoring user empowerment. The software platform model will provide design guidelines for IoT platform providers and validate our framework. Therefore, the general research question at the heart of this thesis is:

*How to empower users in the IoT platforms design to enable them create applications, services, and realize personalization.*

1.3 Methodology and Approach

This research uses design research methodology to address the research question. According to Vaishnavi et al. [13] design science research activity emphasizes knowledge generation by building and identifying the properties of design artifacts. Our objective is to give an answer to the research question by building an IoT platform (i.e. design artifact) for user empowerment. Our research process will be embedded in a platform in order to encourage and promote user empowerment in the IoT within the IoT research community and industrial projects. The platform will demonstrate how user empowerment enables the creation, personalization, and distribution of services in the IoT ecosystem.

Our research follows the general design cycle that Vaishnavi et al. [14] describes as a process of five steps: awareness of a problem, suggestion, development, evaluation, and conclusion. From the suggested general design cycle, we base our research on the steps shown in Figure 3.
Awareness of the problem of user empowerment comes from our observation of current industry trends, and comparison with successful models in the IoT like the smart phone ecosystem [11]. Our initial experimentation with IoT hardware and software platforms confirmed this. Our experimentation with open source hardware platforms shows an abundance of software libraries for prototyping but also identifies the lack of platforms for distribution and reuse of finished applications and services. Our experimentation with commercial IoT hardware (e.g. smart meters [15]) shows the constraints imposed by hardware providers that are building closed systems with one application per hardware thus disempowering user led innovation. The awareness of this problem is also confirmed by a review of the existing literature on user empowerment in the IoT (F. Michahelles [9]). Having identified the problem, we have built the general framework in the suggestion phase. In this step, we reviewed the relevant literature and the current trends in the IoT. The exploratory study towards the general framework identifies the core elements for building IoT platforms favoring user empowerment. In the development phase, we identified requirements for IoT service platforms and we implemented our suggestion in a software prototype. After the development of IoT service platform, we carried out an evaluation of the implementation using design science evaluation methods in
order to assess our design. As a result, we generate new knowledge on the issue of user empowerment in IoT.

1.4 Objectives
The first objective of this thesis is to study IoT literature and trends from user empowerment’s perspective and to categorize them towards a unified framework. The framework will provide high-level guidelines for the IoT research community to better understand and define user empowerment in IoT. The second objective is to suggest a model of IoT platform enabling user empowerment. We will confirm the validity of the framework by implementing an IoT platform prototype empowering users to create and personalize IoT services. The platform prototype will help define design patterns for engineering of IoT service platforms enabling user empowerment. The purpose of the platform prototype is twofold. First, validate the general framework and second, demonstrate how users can be more involved in the creation of IoT services. As an overall objective, this thesis shows how user empowerment for the creation and personalization of services can help improve the IoT ecosystem.

1.5 Thesis Outline
This thesis is structured as follows: Chapter 2 presents the state of the art in IoT and user empowerment. We first present the concept of Internet of Things, major technologies, and the current state of the art in IoT. This part presents the current progress for the establishment of IoT infrastructure, standards, protocols, and platforms. The research projects selected in this part represent the leading progress in business and industry-centered IoT. Secondly, we present the research on user empowerment and open innovation in the IoT. Through these contributions, we observe the current shift of industry-centered IoT toward more open and user centered approaches. Moreover, we identify how this works address the issue of user empowerment in the IoT.

Chapter 3 draws general requirements for user empowerment. We identify the stakeholders in the IoT and define requirements that will empower users to participate in the building of IoT applications and services. Our observations from the state of the art lead us to define three required elements for user empowerment: empowerment for user-led innovation, empowerment to deal with the diversity of IoT technologies and services, and empowerment for service creation and diffusion.

Chapter 4 defines a general framework for user empowerment in the IoT, based on identified general requirements. We define a model of interactions between users through the IoT platforms. The model relies on the concepts of People,
Places, and Things in IoT and their virtual counterparts. We argue that this model provides the design elements for the IoT platforms enabling user empowerment.

Chapter 5 presents a reference implementation of the model for user empowerment. The aim of the implementation is to validate our theoretical framework. We focus on building an IoT service platform that empowers users for service creation, personalization, and deployment. We validate our general framework by demonstrating an implemented platform prototype. We also evaluate platform prototype by analyzing its design elements.

Chapter 6 revisits the general framework after implementation. We first define design invariants for IoT platforms based on the results from implementation. After revisiting the framework and the model of user empowerment, we generalize the model and discuss application domains.

Chapter 7 concludes this research by summarizing the contribution and discussing on future work.
Chapter 2. **State of the Art**

### 2.1 Internet of Things

The term Internet of Things has been often used to describe uniquely identifiable objects and devices connected to the Internet network [16]. The basic idea is that any object like food, clothes, home appliances, vehicles, buildings, etc. has a unique identifier, can sense its environment and exchange digital information either through the Internet, through embedded connectivity or other proxy devices. Therefore, all mundane objects can provide better insights about themselves and the environment. Ultimately, the promise is that with a better insight into our environment, we can create new services that will enhance our lives. The application domain of the Internet of Things is almost limitless, ranging from better urban planning, sustainable urban environment, medical treatment and continuous care, emergency responses, mobile payments, intelligent shopping, to waste management and recycling [17].

The idea of Internet of Things roots back in Mark Weiser's vision of Ubiquitous Computing [18]. The vision, originating in 1991, forecasts that computers will be embedded in light switches, thermostats, stereos, ovens, and other appliances suited to a particular task. All these devices are supposed to be interconnected in a ubiquitous network. His work takes into account human psychology and assumes that technology will, over time, disappear from our sight. This vision is not about hiding computers from humans, it is about our ability to cease to be aware of technology when we master it sufficiently well. Thus, mature ubiquitous technology will be indistinguishable from the environment in our everyday life. Currently, the research stream that follows this vision of Ubiquitous Computing is progressing in the development of a range of “smart technologies” (e.g. smart meters, smart grid, and smart environment). Ubiquitous computers are not only interconnected, they can also connect to the Internet and provide the technology for creation of global scale services. Although some aspects of Ubiquitous Computing cover connecting embedded computers to the Internet, the term Internet of Things has not been used. Instead, the vision of Internet of Things is much broader and includes Ubiquitous Computing vision and technologies.

The term “Internet of Things” has been coined by Kevin Ashton in 1999 [2]. His vision of Internet connected objects presents the works of Auto-ID labs [19]. Auto-ID labs are leading network of academic research laboratories in the field of networked Radio Frequency Identifiers (RFID). The vision forecasts that the computers will overtake humans in the creation of digital information. It postulates that the Internet is almost wholly dependent on human beings for creating digital information. The problem is that humans have limited time, attention, and accuracy and are not good at capturing data about things in the real world. In order to cope with this problem, it has been suggested that
computers should be empowered to gather digital information about the real world. RFID and sensor technologies should enable computers to observe, identify and understand the world without the limitations of human intervention. Currently, “Auto-ID labs” is an independent network of seven academic research labs that follow this vision. This research stream is focused on enabling computers to read sensors and unique identifiers from objects in order to create business value for the enterprise. Approximately 100 research projects are participating in their research program composed of three building blocks: hardware, software and network, and business processes and applications. From Figure 3 that shows their research program, we observe that this research stream adopts industry-centered approach for building the IoT. However, this research stream brings major progress to the field of enabling technologies for IoT.

![Auto-ID labs research program](http://www.autoidlabs.org/research/page.html)

Figure 3: Auto-ID labs research program

Auto-ID research is focused on developing the essential systems for dynamically extracting, networking and storing product (i.e. object) data. These works are relevant for our research as they define the infrastructure for connecting tagged products to the Internet. Although business aspects of these works are included, we essentially focus on infrastructure for connecting objects. This infrastructure enables enterprises to automatically extract data from tagged products and use it to control manufacture and supply. McFarlane D. provides an overview of Auto-ID based systems for control of products [16]. As shown on Figure 4, he describes two Auto-ID based systems: information-oriented (open loop) and
decision-oriented (closed loop). Open loop systems provide increased accuracy, quality and timelines of data within existing enterprise information systems, while closed loops imply changes in the decision making process.

![Figure 4: Open and Closed loops for Auto-ID Systems](image)

Although the Closed-loop suggests an extension by embedding decision and execution processes within enterprise environment, we only focus on the common elements that define how data is collected from tagged products. Thus, we focus on the system elements for data collection that can be valuable for users. The system elements for automatic data collection are:

1. Operation,
2. Product sensing, and
3. Analysis

These elements imply tagging objects (Operation), reading and storing data about objects (Product Sensing), and data analysis (Analysis). The enabling infrastructure for Auto-ID systems consists of:

1. **Electronic Product Codes (EPC)** is embedded in a memory chip (smart tag) on each product. These codes are used to uniquely identify each physical object. EPC is used at the Operation level.
2. **Product Markup Language (PML)** is a standard language for describing physical in the same way that Hypertext Markup Language (HTML) describes content on the Web. PML is used during Product Sensing and storage of product information.
3. **Object Naming Service (ONS)** is a service that indicates to computer systems where to find information about the physical objects carrying an EPC code. ONS is partially based on the Internet’s existing Domain Name System (DNS). ONS is used during Product Sensing in order to find product information on the Internet.

The enabling infrastructure provides an input for Analysis. The analysis of product data is performed using enterprise resource planning systems. This infrastructure empowers enterprises to identify individual objects by opposition to identifying category of objects. Consequently, the more information about objects is accurate the more manufacture and supply can be optimized. This work accurately describes the industrial approach of building IoT. It also provides the building blocks of the systems for an automatic unique identification of objects.

From an economic perspective, following the Auto-ID labs research program, Fleisch E. defines the IoT concepts [20]. He states that every physical thing in this world can be connected to the Internet. However, there is a boundary between smart things, which can change their states and communicate seamlessly with their surroundings with none smart things, which only have a single status and are not active in communicating. He defines six main points that distinguish the IoT from the Internet.

1. **Invisible versus flashy hardware.** Whereas the network nodes on the Internet end in powerful multipurpose computers, the network nodes in the IoT end in small, in many cases even invisible, low-end computers.
2. **Trillions versus billions of network nodes.** Fleisch estimates that the number of connected devices is likely to increase from actual ~5 billion to at least 84 billion microcomputers. In addition, for uniquely identifiable objects without computing power the estimation is up to 555 billion. These estimates already suggest that people will not be willing to communicate directly with embedded computers and that a new network infrastructure (IoT) is required.
3. **Last mile bottleneck versus highway.** Driven by user demand (multimedia content, streaming etc.), the last mile bandwidth has been significantly increased over recent years. The IoT devices only require a small portion of this bandwidth.
4. **Babylon versus global identification and addressing.** So far, most network addresses for smart things are based on local and vendor-specific closed-loop schemes. However, IoT should follow the successful example of the Internet and use globally available addressing scheme in order to enable tagged object to be accessed by any computer.
5. **Machine-centric versus user-centric.** Internet services are targeted towards human beings as users. The IoT almost completely excludes
humans from direct intervention. When humans need to be involved, e.g., for decision-making, they contribute via personal computers and mobile phones.

6. **Focus on sensing versus communication.** The success of the Internet started with the World Wide Web (WWW), which allowed humans to communicate at a low cost. The second boost of the Internet was driven by enabling users to generate digital content, i.e., data is not only consumed by users but also provided by users. IoT gives another dimension to data. It enables us to automatically sense and measure the real world.

These differences summarized in six points describe IoT without distinction between the infrastructure and application levels. According to Fleisch, on the infrastructure level, IoT can be viewed as an extension of the Internet whereas the application level can be observed as a special set of Internet applications enabled by the IoT infrastructure. This work allows us to understand what distinguishes IoT from the Internet and to identify that the application layer can be observed as an extension of current Internet applications.

The most common technologies for building the Internet of Things ecosystem are Radio Frequency Identifiers (RFID), Wireless sensors, Smart Technologies, Electronic Product Codes (EPC), Near Field Communication (NFC), and embedded systems. Some of these technologies have already matured out of the research stage and have been improved by the industry [21]. Nowadays, it is expected that IoT technology will also cross the boundaries of the industry to become pervasive. Ericsson estimates 50 billion of connected devices in 2020 [1], Fleisch E. estimates at least 84 billion [20] and many enterprises were planning to adopt IoT technologies from 2012 [22]. Although the progress in enabling technologies and its lower production cost is the foundation for building IoT, the disruptive impact of these technologies can be foreseen already across the society and the economy. The nature of technology that “knows” everything about people and “senses” everything around people requires new public policies, business models, and ethical guidelines. IoT is therefore a cross domain vision of disruptive technologies and practices with an unpredictable impact on our society and economy [12]. In order to develop an IoT ecosystem and to cope with the problems of the disruptive nature of IoT technologies research is covering all the fields that IoT technology can affect. Among the major fields are communication protocols, standards, applications, and governance [23]. The research projects members of European Research Cluster on the Internet of Things (IERC) address technology issues as well as societal and economic trends [24]. Consequently, there are several definitions of the Internet of Things depending on context. For example:

Definition focused on technology [25]:

...
"A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability."

Definition focused on economy [26],

"A world where physical objects are seamlessly integrated into the information network and where the physical objects can become active participants in business processes. Services are available to interact with these 'smart objects' over the Internet, query and change their state and any information associated with them, taking into account security and privacy issues."

Definition focused on social, environmental, and user contexts [27]:

“Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts”

In this thesis, we focus on user empowerment and therefore the last definition of IoT is the most appropriate.

The research following industry-centered approach is mainly focused on developing platforms for business integration. EBBITS (Enabling business-based Internet of Things and Services) is one such research project. The aim of this project is to establish a platform for interoperability between heterogeneous structures for the business environment. It is intended to create ubiquitous infrastructure that dynamically connects to sensors and devices in the physical world (Figure 5).

EBBITS project [7] performs research and development in the following areas:

- Internet of Things Architecture Technology using Service oriented Architecture (SoA)
- Communication Technologies with distributed discovery architecture
- Integration of Scalable Network Technologies with sensors in a transparent and seamless way
- Software and Services with their orchestration and support for semantic interoperability
- Security and Privacy Technologies for cloud computing
- Enterprise Framework with sustainable business models
Although this project aims at creating an important infrastructure for the IoT it does not address the potential of users and third party firms. It is centered on creating an infrastructure without addressing the issue of user empowerment.

ELLIOIT (Experiential Living Labs for the Internet of Things) [28] is a project that is more user centered. The aim of the project is to study and develop a set of Knowledge-Social-Business (KSB) Experience Models to represent human behavior in IoT [29]. The project is developing a user experience based platform implementing the ELLIOIT approach. The aim of the project is to enable significant increase in the innovation capacities in enterprise environment through the Open Innovation paradigm and co-creation with users and designers. Although the project claims to be user-centered, it is focused on the empowerment of enterprises to benefit from user knowledge while designing IoT services. Their user experience based platform gathers user knowledge and experience in the context of IoT. The research experiments within three Living Labs, each composed of physical space, information space, and societal space. The project explores the potential of user co-creation techniques and tools, such as serious gaming, and participative requirements engineering. The project is
expected to dramatically increase the adoption of IoT and to enhance the potential of innovative IoT application and services opportunities for enterprises.

2.2 User Empowerment in the Internet of Things

Despite research attempts to cover all concerned domains, IoT development remained mostly industry-centered for more than a decade. More recently, a research stream started to address the challenge of disruptive IoT technologies by involving users in building the IoT ecosystem. It is based on the observation that Web 2.0 empowers users to participate in the creation of digital content, applications, and services in IoT [9]. The underlying idea is to empower users to build the IoT ecosystem “by the people and for the people” in order to solve the people concerns and in particular problems related to privacy and service personalization. The aspect of empowerment here is to lower the barrier to entry for user participation to generate digital content and innovate in IoT.

Generating digital content by users assumes their interaction with Internet enabled physical objects. This interaction with physical objects by controlling sensors and actuators attached to microcontrollers is called physical computing. P. Papadimatos in his thesis [30] defines physical computing as the “distribution of computation across tangible physical artifacts that are spread through the physical environment”. As the physical artifact using sensors and microcontrollers can be connected to the Internet, the physical computing is becoming an important element of the IoT. Arduino is a widely known and accepted hardware platform for physical computing and IoT [31]. It provides a platform for easy prototyping of Internet connected objects. It consists of a board with a microcontroller to which can be attached various sensors, actuators, wireless or wired extensions, or even home appliances (Figure 6).

Arduino hardware and similar approaches (e.g. Netduino [32]) provide an “Open Source Hardware” empowering technical users and enthusiasts to build smart connected devices or to hack their home appliances in “Do it yourself” ways. Sensors, actuators, and hardware components called “extension shields” are easily attachable to Arduino boards providing wide hardware modularity. The hardware specification together with embedded software and development tools are open to all users. The open source hardware introduces a new excitement into hobby, art, and education. Therefore, users who program Arduino include designers, artists, and amateur programmers. First-time users are able to learn open source hardware programming by referencing a plethora of examples distributed by online communities. User created programs for micro controllers, commonly called “sketches”, are easily uploaded to the board so there is no need to program the microcontroller operating system.
However, writing a program is a complex process, and in the case of open source hardware, it involves knowledge on physical computing, electronics and several programming languages in order to create IoT applications and services. Although users can be considered fully empowered to create the IoT services from open source hardware and software, the complexity of the task to create services only empowers enthusiasts willing to devote their time and energy to complex programming projects. Visual programming language such as “Splish” [33] or other variants of end-user programming [34] have been suggested in order to simplify the task of hardware programming for non-technical users. A graphical interface is provided within a development environment where users can drag and drop icons representing conditions and interactions between physical objects, sensors, and actuators. However, the integration of physical computing with the IoT is a new challenge and end-user programming of complex applications or services available through the Internet has not been addressed in these works.

The Cool Town project [35] was one of the first to use Web protocols to create ambient environments. The aim of this project was to tie Web resources to
physical objects and places allowing users to interact with physical objects using laptops, personal digital assistants, and smart watches [36]. Thus, nomadic users could use mobile devices to read the barcodes, or objects with electronic tags (RFID) and retrieve associated web resources. In addition to the simple reading of web resources associated with the objects, users can also post new digital content associated with their physical location. The Web architecture has been chosen in order to avoid creating just another middleware. In order to ensure interoperability among heterogeneous devices, Web standards such as HTTP have been used as they support adaptation to wide range of reading devices. The project introduces the notion of “place managers”. Place managers associate objects with tags and web resources. Every place manager is therefore in charge of creating a repository of associations between the physical environment and the web resources. Figure 7 shows an example of an architecture with an exhibition hall.

Moreover, the project introduces the notion of personalization and delivery of web content according to the user’s context. Physical locations can be active and push web resources to user device without the user’s explicit request. That is, the place managers can define web resources associated to physical locations to be delivered according to the time of the day or user identity. However, the work also identifies that this model of computing environments
presents potential threats to user privacy. An environment that can process the user’s detailed location information can become a threat to user privacy.

Based on pioneering works on integration of tagged objects into the Web the research community continues to integrate other IoT technologies. The Web of Things community became very active in the integration of tagged objects, smart technologies, and wireless sensors with the Web. Using the Representational State Transfer (REST) protocols [37] instead of enterprise middleware, their aim is to lower the barrier of entry to end users and to empower them to create simple, ad-hoc applications that combine real-time data provided by sensor nodes.

D. Guinard et al. uses well-known web standards to represent the physical objects and sensor nodes as web resources allowing end users to create applications using simple web mashup techniques [38]. Additional benefits of this approach have been demonstrated in their projects with the integration of user created web mashups with enterprise business solutions used in the manufacturing industry [38]. The integration of physical objects and sensor nodes with the existing web infrastructure also empowers users to easily share digital content from tagged objects or wireless sensors with online social networks. Several examples of smart objects or home appliances that “Tweet” [39] demonstrate the potential of integrating the IoT with online social networks. Additionally, authentication services of online social networks can be used to grant access to user’s social relationships on potentially sensitive personal information [40]. D. Guinard in his thesis [41] provides a detailed look into this integration of physical objects with the web, defining the concept of “Web of Things” by analogy to “Internet of Things”. A tremendous effort has been made on technical aspects and building blocks of the web of things application architecture. Thus, four main architecture layers have been suggested in order to integrate smart devices, sensors and tagged objects into the web: accessibility layer, findability layer, sharing layer, and composition layer. Figure 8 shows the layers of the Web of Things architecture. Although the figure shows five layers, the application layer has received very little attention. Applications could be built on top of each of the four layers.

The main contribution of this work is the exploration of web protocols in order to use the web for an IoT application layer. By leveraging IoT technologies on the Web, this work attempts to integrate the Internet of Things with the existing Internet of People. However, the channels for distribution of user created applications or web mashups and the details of user participation have not been explored. The user empowerment is supposed to come “naturally” with better ways to program web mashups.
Our review [11] of IoT enabled Web platforms shows that the Web of Things concept is gaining a positive momentum. We observed an emergence of web based “Data Brokerage” platforms. The most prominent examples of these platforms are Cosm (formerly Pachube) [42], Evrythng [43], Thingspeak [44], and Nimbits [45]. We defined these platforms as “Data Brokerage” because their focus is to collect data from Things and to accommodate this data for other platforms hosting web applications. These web-based data brokerage platforms are designed to empower users to share digital information from the IoT on online social networks and other web platforms. Their relatively simple RESTful API allows users to connect things and web programmers to export digital information in the form of web streams. Figure 9 shows a simplified architecture of these platforms.

Data Brokerage platforms are striving to break the vertical silos of industry-centered IoT. Instead of having one platform per hardware provider, they are attempting to become hubs for connecting the heterogeneous IoT hardware.
As shown in Figure 9, the design of Data Brokerage platforms can be represented as a stack with the following elements:

1. RESTful API at the bottom allows technical users, enthusiasts, and hardware providers to connect *things* using HTTP verbs (i.e. GET, PUT, POST, DELETE).
2. Data storage records digital information from connected *things* on the platform.
3. Event triggers and/or query languages module. Event triggers allow web developers to define the actions that are activated according to the changes in the Data storage module (e.g. notification if the temperature goes over 35 degrees). Query languages allow web developers to pull the digital information from Data storage. It allows queries similar to SQL to be executed through the REST API. The digital information generated, either with event triggers or by executing queries, is exported in the form of web streams like RSS, XML, JSON or other. These streams are used in the same way as the RESTful web services and can be consumed by any third party application through the Internet.
The model of web based data brokerage platforms follows the user-centered approach for building the IoT (Figure 2). The openness of the model for connecting and sharing of digital information empowers technical users to build and connect their hardware and developers to create the web applications for IoT. Several examples demonstrate how this model of platform empowers individuals to build the monitoring of the real world and create web reports using IoT technologies. A prominent example of user empowerment was shown with radiation measurement taken by users after the Japanese nuclear meltdown in 2011 that was also described as “IoT bottom-up revolution” [46]. Japanese individuals were monitoring radiation level by Geiger counters and streaming real time data to Pachube [42] where it was publicly displayed. The true scale radiation readings in form of data streams were published immediately, contradicting the intentionally lower measurement results published by the authorities. Although these platforms empower users to build an IoT ecosystem, their focus is to deal with “Big Data” collected from connected devices. Consequently, other aspects of user empowerment like service creation and distribution are neglected.

Our review [11] uncovers that they don’t focus enough on plug and play hardware and that the number of third party innovation artifact like software applications is limited. However, we also observe that these platforms can be the common ground for distribution of third party software applications and therefore empower users to create and distribute IoT services.

G. Kortuem et al. suggest a model for user innovation in IoT through the connected marketplaces [10]. They observe user empowerment from the perspective of online channels i.e. marketplaces that enable users to distribute their innovation artifacts. In that work, the authors drew inspiration from the Smart Phone application marketplaces as channels for distribution of user innovation (i.e. software applications). They performed a survey of tools for users and developers in the IoT space and uncovered a rich set of tools for the creation of hardware, software, and data but also a poor support for the distribution and share of these artifacts. In order to cope with this problem, they put forward a model of connected marketplaces (Figure 10).

They define a marketplace as “a technical platform for efficient and effective distribution of artifacts” and suggest five connected marketplaces:

- **Smart Object Marketplace**: intended to enable users who may want to build their own Internet enabled objects to share these objects or trade them with others.

- **Application Marketplace**: intended to enable users who will program the software applications for Internet enabled objects to share or trade these applications. This marketplace is linked to the Smart Object Marketplace.
• **Configuration Marketplace**: intended to enable users to share and trade the configuration artifacts that are defined as “software artifacts for coordinating smart objects”. This marketplace uses and operates the artifacts of the two preceding marketplaces.

• **Data Marketplace**: intended to enable users to share and trade the data from Internet enabled smart objects. This marketplace is linked to the preceding three marketplaces in that it contains data emitted by objects, applications, and configurations.

• **Manipulator Marketplace**: intended to enable software firms and users to write, share, and trade “manipulators”, parameterized aggregators that detect patterns across a set of data. For example, manipulators can be created for a better understanding of energy consumption.

![Figure 10: Connected Marketplaces Supporting User Innovation in the IoT Space](source: Kortuem G. and F. Kawsar, Market-based user innovation in the Internet of Things.)
The model of five connected marketplaces has been analyzed using an example of smart homes. The authors argue that the connected nature of these marketplaces creates an exponential innovation effect. We observe in this work an intention to empower IoT users to shape the IoT technologies by sharing and trading their innovation artifacts: hardware, software, and data. However, this promising model of connected marketplaces provides solid arguments to create IoT marketplaces, but the model has not yet been implemented.

Although the IoT vision is not new, user empowerment received very little attention from the research community. The term “empowerment” is very wide and should be positioned in the context of IoT. For example, the *Oxford dictionary* [47] defines the term *empower* as “give (someone) the authority or power to do something”. The *Merriam-Webster dictionary* [48] as “to give official authority of legal power to” or “to enable”. However, depending on the power given to someone the term “empower” has different meanings across disciplines. For example in politics, it refers to giving someone rights to work, to vote, or to enable e-participation (e.g. K. Soonhee 2012 [49]). In information systems, it refers to education and training that give power to employees to work with enterprise information systems (e.g. R. Sehgal 2007 [50]). User empowerment in the IoT is often defined in the context of Web 2.0 (e.g. F. Michahelles 2009 [9]). Web 2.0 represents cumulative changes in the ways software developers and end-users use the World Wide Web. The success of Web 2.0 demonstrates the importance of platforms that empower users to participate. The Web 2.0 platforms feature the interfaces enabling developers and users to extend and personalize the services provided by these platforms. The most prominent examples of Web 2.0 platforms for user empowerment are online social networks. The core service of online social networks empowers users to create the graph of their social relationships. Users create links between their social relationships and use it to share digital content and to communicate. Moreover, these platforms expose interfaces that allow their services to be used on other platforms. For example, social plugins [51] for authentication or rating of digital content that can be used on other web platforms and empower users to easily create and share digital information among their online social networks. These interfaces empower users to access several services by providing a consistent user experience. In addition, these platforms expose an API that allows external developers to create applications and extend the core functionalities of the platforms. The social graph created by users can be used by software applications in order to create or personalize services with online social context. At present, the most prominent example of online social networks reaches 1.11 billion of monthly active users and 10 million third party applications [52]. As a result, the interfaces of Web 2.0 platforms empower users to extend the core services and allow communities to grow.
In the context of IoT, data brokerage platforms and the Web of Things community follow the successful example of Web 2.0. However, using the web protocols and building IoT enabled web platforms are only the first steps towards user empowerment to create and personalize services. These platforms empower users to connect Things using web technologies but they still lack interfaces for the creation and distribution of IoT services. Web 2.0 platforms empower developers to extend core services and IoT platforms should follow this approach. The interfaces like application registries and application markets empower communities of developers to create new applications and services and distribute them to users. Larger number of applications and services also empower users to benefit from new services on these platforms thus enabling to obtain new users. The growing number of users attracts communities of developers by opening new opportunities to monetize the applications and services. Consequently, the IoT platforms that follow Web 2.0 approaches need to expose the interfaces not only for connecting Things but also for creating and personalizing new applications and services. As a result, user empowerment in the IoT enables the creation and personalization of services based on IoT platforms designed to allow this two-sided network between users and developers. Furthermore, in order to enable user empowerment, the design of IoT platforms should take into account the ubiquitous nature of IoT technologies and the requirements of a multi-stakeholder’s ecosystem.

2.3 Summary

In this chapter, we summarized the works that attempt to develop an IoT ecosystem. Since early works on Ubiquitous Computing and auto-id technologies, the focus of research remained mainly industry-centered. The research progressed in building the enabling infrastructure for an IoT. This approach does not lead to user empowerment to create and personalize services provided by IoT infrastructure. Currently, open source hardware and software platforms attempt to empower IoT users to build their own services. However, this model does not empower users to distribute their innovative services. In order to tackle the challenge of service distribution, the model of connected marketplaces has been suggested. However, this model has been suggested at a very high abstraction level and has not been implemented.

Our approach should take into account the fact that the IoT infrastructure will be provided by hardware provider firms who might lower the price of the IoT hardware and supply the market with Internet connected objects. However, users should be empowered to create, distribute, and personalize IoT services using this IoT hardware. In the next chapter, we will define the requirements for a general framework that would empower users to build and distribute services at the application level i.e. on top of IoT infrastructures.
Chapter 3. **Requirements for User Empowerment in Internet of Things**

Currently, the importance and value of user led innovation is widely accepted. Open Innovation is becoming a common model for collaboration between firms and users in order to create successful products and services [53]. The Internet, Web 2.0, high speed Internet and the general drop in hardware cost enable communication between firms and users and provide the platform for collaboration and innovation. Web 2.0 demonstrates the valuable impact of online social networks and collaborative platforms as communication channels for companies and users. In this fast moving and competitive environment, companies are increasingly using these online channels for connecting and interacting with users [54]. The IoT introduces the concept of connected objects but needs the critical mass of firms and users in order to shape this new ecosystem. In order to reach this critical mass of users the IoT hardware and service providers need to empower users to participate in innovation. It requires hardware providers to connect critical mass of things. It also requires users to participate in the creation of new services. As Web 2.0 grew with the number and the contribution of its users, the IoT is bound to follow the same path. In this case, the value of users’ contribution will be highly beneficial for IoT and therefore technologies and services will be better adapted to their needs. When users get familiar with new technology, they naturally create online communities in order to reach various objectives. At a community level, users provide various useful inputs for technology providers in order to improve products and services. Although there can be plethora of community types and user profiles involved in innovation we distinguish two user profiles of IoT platforms that, if involved, can each influence the IoT ecosystem in respective ways:

- End users
- External developers or software firms

This distinction allows to simplify our observation of stakeholders in IoT ecosystem and to focus on the most important elements of user empowerment for innovation. Big hardware providers as well as network providers are at the beginning of the innovation process. They are establishing the infrastructure that allows everyday objects to become connected things. The hardware and the software platforms should be open enough in order to involve both above-mentioned communities of hardware users. Hardware or software platforms should also empower both community profiles to participate in innovation in order to create new IoT services. Figure 11 shows the interaction of these three stakeholder profiles and their contribution to innovation in IoT through empowerment.
In order to build the empowerment framework introduced in Figure 11, we need to identify general requirements for IoT stakeholders.

### 3.1 Internet of Things Hardware Providers

The IoT hardware providers create Internet connected objects. They are the first group of IoT stakeholders and they should ensure that the hardware allows users to build their own services from it. Consequently, we defined requirements for IoT hardware providers that are user-centered.

**Communication standards**: In order to create the Internet enabled hardware, common communication protocols and standards are required. The standardization of communication protocols should create the IoT hardware that users can connect to the Internet in plug and play way e.g. [55].

**Open innovation**: Hardware providers should involve users at all stages of the innovation process, from hardware design to service use. A valuable input should be collected from the community of end users through crowdsourcing in order to design user-friendly hardware. However, hardware providers should not
hinder the innovation process when the hardware reaches the market. The IoT hardware should be designed as a platform that allows users to build IoT applications.

**Open IoT service platforms:** The IoT platforms can be “open” from different points of view (e.g. open source or open for development of software applications). However, the first requirement is that hardware providers should empower users to decide to which service available on the Internet they will connect the IoT hardware. The second requirement is that the hardware providers create platforms open to service innovation. It does not necessarily imply open source hardware or software but a platform that allows for the creation of new services. An example of open hardware platform is the smart phone. User acquires a smart phone that only has an operating system, and a few software applications. The smart phone is opened as a platform in the way that it features an API that allows developers to access device functionalities (e.g. accelerometer, positioning system, storage, etc.). These design features allow users to develop and use their own or third party services (e.g. software applications).

**Empowerment for the creation of IoT services:** Not all IoT hardware can be compared to the smart phone devices. Some objects in the IoT can have some processing power and Internet connectivity (e.g. smart meters) while some other only have unique identifiers (e.g. clothes). In order to bring all IoT objects to the application level, hardware providers should provide service platforms for Internet enabled hardware. The IoT hardware connected by users should have virtual proxies on these platforms and the platform should be opened for creation of applications (Figure 12).

As a result, virtual representation of connected objects should allow the creation of services by using virtual proxies of objects whether the objects are smart devices or tagged objects. An open API should allow interaction with proxies. The model of the platform shown in Figure 12 is quite similar to “Data Brokerage” platforms. However, this model should be extended to meet requirements for user empowerment and become a model of IoT service platform.
3.2 Internet of Things Users

**Service personalization:** Users of Internet enabled objects should be empowered to personalize services provided by these objects. Personalization of these services should be performed on IoT service platforms. Users should be allowed to use the virtual proxies of objects and chose the services. The concept of “service” is wide e.g. monitoring of virtual proxies, remote controlling through virtual proxies, graphs, charts, and tools for data analysis, software applications and other. Although all these examples of services are useful for user empowerment, we focus on services as software applications.

**Social networks:** Users of Internet enabled objects should be empowered to use services provided by software applications within their online social networks. This is backed by the observation that the online social networks allow communities to share their personal experiences by writing reviews, commenting and rating [56]. This user input should be enabled by digital information generated by connected objects. However, the sharing of digital information should remain within the user defined social context. Thus, the online social network should empower users to personalize the IoT ecosystem and services by controlling the sharing of digital information within their social context.

**Control of digital information share:** Users of Internet enabled objects should be empowered to control the objects and to share digital information on IoT service platforms according to physical environments. As the IoT implies tremendous increase in the number of connected objects, users should be
empowered by mechanisms to control sets of connected objects and services. This feature should empower users to personalize services by personalizing virtual environments instead of personalizing each Internet connected object.

3.3 Internet of Things Developers

**Service creation:** The cloud-based platforms should be designed in order to empower developers to create new services. From the point of view of hardware providers, developers and third party software firms are also hardware users. They use hardware in order to create new services thus continuing the innovation process after hardware providers. The IoT service platforms should feature programming interface in order to allow developers to develop services. Therefore, developers should be empowered by software tools and libraries in order to create applications.

3.4 Internet of Things Marketplaces

**Service diffusion:** Internet of Things provides an infrastructure for new application and services that are supposed to improve our quality of life but most of them are yet to come e.g. [17, 57]. Current trend of industry-centered approach followed by hardware providers create one Thing – one application model. Smart meters for individual appliances [15] are an example: hardware providers offer one smart meter and one computer software with one software application, for monitoring the electricity consumption of appliances. This model does not lead to the creation of the Internet – scale applications and services composed from plethora of connected Things, which would really be useful. Consequently, the IoT applications and services need better integration of things that should follow a “one Thing – many applications” model. In order to apply this model, hardware providers should empower external firms and developers in the process of service innovation but also empower them to distribute new IoT services. Successful examples of such approaches can be observed in the smart phone ecosystem. Engaging third-party firms and developers in the innovation process in the smart phone ecosystem is achieved with application marketplaces. Designed as cloud based services these application marketplaces are becoming artifacts for distribution of third-party applications and services. With large numbers of end users, smart phone marketplaces are an attractive hub for monetization of software applications for third-party firms and developers. Moreover, an increasing number of available software applications is attracting new end users thus increasing new market opportunities for application developers. This “network effect” between end users and third party developers explains the extraordinary impact of application markets: “When successful, these platforms catalyze a virtuous cycle: More demand from one user group spurs more from the other.” [58].
3.5 Summary

In this chapter, we defined the requirements for a general framework for user empowerment in IoT. These requirements involve three groups of stakeholders: hardware providers, service users and service developers. IoT hardware providers should empower two other groups of stakeholders by creating prerequisite infrastructure. We assume that the IoT infrastructure, open for service creation will be established by hardware providers. That allows us to focus on empowerment of end users and developers. Requirements for empowerment of end users and developers provide the elements for service personalization, online social networks, and control of sharing of digital information. Requirements for service developers provide the elements for service creation and service dissemination.

In order to design a general framework for user empowerment in the IoT we will analyze the elements of empowerment that emerge from the requirements related to the end users and developers.
Chapter 4. **Framework for Empowerment in the Internet of Things**

### 4.1 Defining the Framework

Hardware providers are the first stakeholders that should establish prerequisite infrastructure for the creation of IoT services. However, The IoT infrastructure needs to empower people to create services on top of it and adapt it to their needs. This empowerment requires leveraging the IoT hardware within IoT service platforms. IoT service platforms should expose hardware abstractions and be designed as open platforms for creation of new services. These platforms will empower users to create services from the level of virtual hardware abstractions (i.e. proxies) instead of rebuilding their own services, as it is the case of horizontal approach with open source and Do It Yourself hardware (Figure 2). Assuming that the hardware providers will fulfill this requirement, we focus on the design elements of IoT service platforms that will empower users to create and personalize services.

In order to define the framework, we first identify the core elements (physical and their virtual proxies) that the cloud platforms will create and mediate. The first element describes service users and service developers that we group under the term *People*. The term *People* does not attempt to create a strong distinction between services’ users and services’ developers. Web 2.0 creates a communication and knowledge platform for communities of both user profiles. It empowers service users to get the knowledge and competencies to become service developers. Consequently, this blurring distinction between service users and service developers allows abstracting both profiles as one concept: *People*. The second element is the environment where *People* interact with Internet enabled objects. We define environments as *Places*. Third element is the Internet enabled object that we define as *Thing*. The fourth concept is the *Application*. We focus here on services developed as software applications (e.g. Software as a Service – SaaS). These four concepts are the building blocks for our framework. We furthermore analyze each of these elements.

### 4.2 People in the Internet of Things

#### 4.2.1 Virtual Identities

Internet has been designed to enable communication between humans. Over time, it allowed development of services intended to interconnect computers used by humans. Although some services created on top of the Internet infrastructure are used anonymously, most of them require identification. Identification gives the capability to the service provider to uniquely identify the user. It also grants service users the ability to access information associated
with their identity provided by the service provider. In the Internet used by humans, service users are intentionally creating their digital identities, requesting an online service, and providing their identity information. Therefore, digital identity of *People* on IoT service platforms allows service providers to identify the service. There are many identification protocols for service users. None of these protocols establishes itself as a unique identification protocol. Instead, several identification protocols successfully coexist. Services were created aiming at unifying multiple identification protocols and allowing service users to reuse their identification credentials on different platforms offered by different service providers (e.g. Open ID [59], Open Authentication [60]).

During the use of services, users associate personal information with their digital identities. In addition, it is an input used by service providers in order to personalize services for users. Web 2.0 and ubiquitous connectivity of service users allow them to frequently upload digital information associated with their digital identities. However, this decision to upload an increased amount of digital information to service provider is mainly up to users.

In the IoT settings where all our everyday objects are connected to the internet, the virtual representation of service users will be even more important. The network of connected objects that can sense our environment will have to know our identity and usage history in order to adjust the services according to our preferences. Although the initial idea of IoT was to automate the product identification process [16], an object (e.g. smart card or RFID tag) associated with a digital identity can be used to identify service users [61]. The disruptive nature of IoT attempts to change the way we use identification in order to use services. If service providers can identify everyday objects, the identity of service users can be inferred. For user empowerment we should consider user-centricity here which means that service users are in control of identification process in order to minimize information disclosure [8].

Heckmann D. in his thesis dissertation describes the concept of *Ubiquitous User Modeling* that is based on digital information collected from smart devices [62]. His model defines how digital information collected by ubiquitous devices (e.g. *Things*) during their interactions with user can be exploited to create a "user model". The user model represents an aggregate interaction history that can allow computer systems to predict user preferences and therefore personalize services. In the IoT, a good compromise between privacy and personalization of services should be user empowerment to control and manage digital information that is collected by *Things* and associated with their digital identities. In this way, service providers would be able to identify users, but users would be empowered to present usage history of different *Things*, at different places, to different service providers. Thus, our framework focuses on user empowerment to personalize services by choosing which usage history created by *Things* will
be accessible by which services. As a result, users control the extent of digital information that is used by service providers as input for service personalization.

### 4.2.2 Online Social Networks

The recent trend of online social network sites allows service users to build virtual social relationships. These online services allow *People* to construct virtual identities and articulate lists of *People* with whom they share social connection within the system [63]. A particular characteristic of online social networks is that they allow *People* to create virtual social relationships that might not otherwise be made. Although *People* are primarily communicating with their offline social relationships, they rapidly extend their online social connections by connecting with other *People* based on shared interests, political views, or activities. Since their introduction, online social networks have attracted millions of *People*. In addition, most online social networks offer ubiquitous access from mobile devices such as smart phones.

IoT promises ubiquitous Internet connectivity of *People* with devices and everyday objects. Consequently, not only mobile devices such as computers or smart phones allow *People* to interact on demand with their online social networks, but also with surrounding objects that become a medium for interactions with online social networks. As a result, the real world social relationships and online social networks of *People* are merging. Any object around *People* can share their presence and activities with online social network in a seamless way. Everyday objects are autonomous and interconnected, working together in order to help *People* in their lives. Although this approach of connected objects promises new socially aware services that could be benefic for *People*, it also opens the privacy issues as new challenges for such services. The invisible interconnection of everyday objects with online social network services is emphasizing the problem of *People* awareness of information sharing [64]. As the IoT technology remains largely invisible and pervasive, *People's* virtual presence on online social networks does not match the real world and online social context. For example depending on where *People* are and which objects they are using they do not necessarily want these objects to share the usage information with their entire online social network. The social context for sharing with for example friends, family, work colleagues should be preserved. This specificity of invisible interactions in IoT imposes a new design of online social networks that takes into account detailed *People* vision of their online social relationships to remain in control of sharing. In order to cope with these problems, users should be empowered to personalize their online social context. IoT service platforms should allow users to define groups from their online social network. These groups (e.g. family, friends, work, etc.) present user defined social context to be used as input for socially aware services. Taking into
account analysis of virtual identities and online social networks, we define People as:

*Service users creating virtual social relationships, defining social contexts and controlling digital information shared within that context.*

Figure 13 shows the online social relationships of *People* that should remain within defined social context and under the control of their owners.

![Figure 13: Online social relationships defined and managed by People](image)

### 4.3 Places in the Internet of Things

#### 4.3.1 Physical Places

Many communication and collaborative systems use the terms “space” and “place” in order to model the environment of *People*. Their definition and their specific function will have to be taken into account while designing the IoT service platforms. Although these two terms have been widely studied in collaborative systems e.g. [65], it is important to clarify their differences and to position them within the IoT context.

We live in a three-dimensional world and we are highly skilled at arranging space around us. For example, the objects we use for work are organized according to our activities. Physical spaces are therefore arranged according to their purpose and needs for interaction. In the physical world, we act within the space where we are. The space and its nearby physical objects help us organize
our individual activities. In our everyday activities, we are not only interacting with surrounding objects but also with other people. The awareness of other people’s presence within space enables us to adapt our own activities. In the IoT context, we define space as:

\[\text{Space is part of the physical world that provides the opportunity for interactions among physical People and Things.}\]

Spaces in IoT provide an opportunity for connecting Things and People. The IoT technologies will enable computer systems to sense and represent this spatial organization of the real world. However, in our everyday interactions we adapt our behavior and activities according to a sense of place and not to a sense of space. Physically, a place is a space where every person can have its own representation of reality. The distinction is like between a “house” and a “home”; a house might be any building but a home is where we live. Consequently, place is an element of user empowerment as it allow user to assign meaning to a physical environment. Our framework will therefore focus on Place as an element of user empowerment for personalization of physical environment in IoT.

IoT service platforms must allow People to define Places that will support interactions between connected Things and other People. We suggest that the IoT service platforms mimic the real world properties when connecting Physical Places and take into account People’s roles. A person in the role of owner should be in full control of the physical places and the interactions between People and Things within its boundaries. The concept of “place masters” has been introduced in early research works on nomadic computing systems [36] and is also a crucial element for the design of user-centric IoT service platforms. The place master arranges a physical space according to one’s own perception and therefore creates Physical Place. The place masters are in charge of physical registration of Things and People in physical places. That involves connecting objects and defining which People are to be aware of interactions in this physical place. To illustrate these concepts we can consider a house and its occupant. A house is a Physical Space and its occupant leaves in it. The house is the occupant’s home place and he or she is the Place Master. He or she can decide which Things to connect to an IoT service platform and with which People to share digital information from the Things in this Physical Place. Figure 14 shows the role of place masters and physical places within online social networks.
This analysis leads us to define Physical Places in IoT. We define Physical Place as:

*Physical Place is the individual perception of space. It is a physical space with some added meaning, convention, and role. Place is arranged by a Place Master and it defines boundaries for interaction among People and Things.*

One particular case could be public spaces that “do not belong to anyone” or actually “that belong to everyone”. In this case, anyone among People can register Things within this space. It is the role of IoT service platforms to allow People to give personal meaning from selected virtual objects and create the virtual public places. The Place Master in this case could be the public authorities.

The field of ubiquitous computing introduced the notion of smart environment and physical spaces that can be modeled into computing systems. IoT leverages connections of smart environments to the global Internet network and emphasizes the need for Place Masters. When connected objects sense our environment and the digital information is shared not only locally but over the worldwide network, it is important to empower users to keep control over connected objects and information share. In the user-centric IoT ecosystem, Place Masters should be empowered to connect physical objects within their physical places to IoT service platforms.
4.3.2 Virtual Places

We have defined how Place Masters connects Things on their physical places and shares digital information from these Things with People from their online social networks. We will now proceed to define how this information can be shared within online social networks.

The IoT ecosystem allows for the integration of everyday objects with People and Places. The aim of IoT technologies is to harvest digital information from the real world objects and to analyze our environment in order to increase our efficiency and productivity [2]. Since connected objects provide digital information about themselves (i.e. tagged objects) and physical places where they are (i.e. sensors), they are creating the basis of contextual digital information. For example if someone is away from home for a longer than expected, the sensors can detect this absence and with the help of other devices and actuators take corresponding actions for energy saving. The field of ubiquitous computing thoroughly studies local interactions of interconnected devices and the use of contextual digital information [66] but the IoT technologies are raising new challenges when sharing such information. In order to enhance our individual productivity, digital information is increasingly being shared globally through the Internet. It is shared across time and space, providing the opportunity for distant places to adjust according to our activities. For example, our presence detected by a sensor in the office could mean that our home has been put in energy saving mode. By connecting Things from our physical places to the Internet, particular care should be given on how the digital context will be shared and used. Moreover, there are other issues to be taken care of in terms of policies of IoT based actions.

The IoT service platforms need to feature not only the virtual representation of People and Things but also of Physical Places alongside Place Masters in charge of managing the creation of their virtual counterparts (i.e. proxies). The virtual representation of physical places has a special role on IoT service platforms as virtual container for People and Things. It embodies the virtual context for virtual identities, and manages digital information share among online social networks. Consequently, an important property of Virtual Places is to limit digital information share defined by Place Masters (Figure 15). The above figure shows the virtual places and their place masters. It also illustrates the role of Virtual Places for managing online social network. In both cases, the virtual place withholds the virtual interactions between People mapping them to their physical entities through their proxies. In order to enhance our efficiency and productivity, the IoT technologies will collect and keep track of considerable digital information needed to recreate our physical and social context.
Our physical and social contexts will help us relate to our environment and to each other but computing systems never capture this context completely [67]. When the social and physical contexts move to a virtual environment, we suddenly have to keep track of where it has gone to and how it is used. Virtual Places allow Place Masters to manage and keep online social context under control. It empowers them to give meaning to their Virtual Places, thus completes this social context and helps them relate to their online social networks.

The second property of Virtual Places is to keep all the digital information generated by People and Things. Keeping this information in Virtual Places allows for better control of historical traces and information lifecycle management. In this way, it helps place masters to have continuous insight of information share over time. The third property of virtual places is to moderate the effect of unawareness of presence, sometimes called invisible audiences [68]. When we move around physical spaces, we are aware of physical objects and the presence of other people. The sense of presence of physical objects and others allows us to structure and adapt our own activity. In the IoT the physical and virtual world are seamlessly blending and bringing geographically distant Things and People virtually close, and the result is the problem of presence unawareness. In addition, the IoT is often associated to the concept of “invisible computer” [18], promising autonomous interactions between connected Things. To illustrate this concept, let us take the example of the connected home: the Place Master of a connected home will not continuously receive
feedbacks and be aware of which People are online and observing his or her interactions with objects at home. That becomes intrusive and a great threat to privacy because it does not permit the place owner to adjust his activities according to the virtual presence of other People. The Virtual Places in IoT service platforms offer the possibility for Place Masters to define and be aware of virtual audience in Physical Places.

Based on these properties and features of Virtual Places we define Virtual Place as:

*Virtual Place is a virtual representation of physical place holding the digital information generated by People and connected objects. In a Virtual Place, registration of People and objects is done by Place Masters who control the Virtual Place. A Virtual Place is a container for managing contextual digital information.*

Although our physical context has much broader and complex meanings [69-72], we consider here that a Virtual Place is only a container for static context separated from activities [67]. Typically, a Virtual Place is an online storage defined with any combination of geographic location coordinates, digital information, and virtual representation of People and Things. It maps a Physical Place and stores the digital information of our physical activities. However, our online activities can be “space less”. People connect and interact with online social networks driven by common interests that cannot be mapped to one Physical Place. In this case, Virtual Place embodies online activities that can include People and Things from different Physical Places (e.g. shopping activity that includes different fashion stores, friends, and clothes). Consequently, Virtual Places are only the first step towards personalization of interactions between People and Things but they need to be extended in order to empower People for personalization of activities.

### 4.3.3 Hybrid Virtual Places

So far, we have defined how Physical Places will be mapped into their virtual counterparts. Although this was the first step towards defining the management of digital information and creation of virtual environments, Virtual Places remain the artifacts for creating static virtual contexts. The next element of user empowerment is to design Virtual Places that can support activities of People and Things that are not performed at one physical place.

Any activity can actually change our physical context. Our interactions with other People and Things also depend on it. In order to support this feature we introduce the concept of Hybrid Virtual Place (Figure 16). Hybrid Virtual Places inherits all the properties of Virtual Places. They can be considered as a
particular combination of proxies and digital information of People and Things in Virtual Places. As they inherit the properties of Virtual Places, they are created and maintained by a Place Master who creates Hybrid Virtual Places and registers People and Things. Therefore, Hybrid Virtual Place is not entirely mapped to one physical location; it is one virtual artifact supporting interactions and activities with distant People and Things. According to these properties, we define Hybrid Virtual Place as:

**Hybrid Virtual Place is a virtual artifact that acts as a container for sets of People and Things registered by Place Masters. Although People and Things registered by a Place Master can have physical locations, the Hybrid Virtual Place is “space less” as it is not a proxy of physical locations.**

In fact, the relationship between Virtual Place and Hybrid Virtual Place is more social than technological. People create social contexts around their physical and online activities and the role of IoT technology is to mimic respectfully social conventions by avoiding information disclosure outside this context.

4.4 Things in the Internet of Things

Humans may soon not be the biggest contributors of online digital information any more. This trend is confirmed with a growing number of IoT devices (the predictions of over 50 billion connected devices by 2020 [1, 22]). Nevertheless,
this number of connected devices brings only a vision of changes to come without a more precise definition of “Things”. The heterogeneity of devices and technologies requires the agreement of a common definition. Different objects (e.g. books, houses, vehicles, etc.) are connected to the Internet using different communication protocols. The question is what are the Things in the Internet of Things? In order to give our definition of Things, let us briefly review the origin and technologies used to connect objects to the Internet. This analysis of design architectures of connecting technologies will allow identifying the properties that are common to physical objects and their virtual counterparts. Based on the common properties we will define the term Things within our framework.

The idea of embedding the computation power and micro computers into mundane objects roots back from the Ubiquitous Computing vision [73] also called Invisible Computing [18]. It introduced a vision of machines like light switches, thermostats, stereos, and ovens having embedded computers interconnected over a ubiquitous network. Following this vision, a research stream developed an entire domain of “Smart Technology”. With embedded computers, our surrounding objects and home appliances are becoming “smarter” and interconnected. From this vision emerged more or less simple or complex devices and systems, from simple sensors to buildings and vehicles (e.g. Smart Sensors, Smart Cards, Smart Grid, Smart Homes, and Smart Cars). Consequently, these devices have embedded computing power and “self” connectivity. Smart devices often embed wireless or wired connectivity hardware that enable their interconnection. Nowadays, it is common to find the Smart Sensors designed as groups of short distance interconnected devices with one base hub or router proxying them to the online services through the Internet. Embedded computers are nowadays also accessible to IoT users and empower them to build smart devices in the “do it yourself” way. Open hardware platforms allow hobbyists to build their own or to hack existing home appliances with embedded Internet connectivity [74, 75]. Using these technologies devices can autonomously send digital information to the Internet, they can be controlled remotely, and they can also use online services and react autonomously.

The term “Internet of Things” has been coined with the emergence and especially with the decreasing cost of automatic identification technologies [2]. Simple identification tags attached to any objects are used to identify objects and automatically fetch the associated digital information from the Internet. The automatic identification process can be summarized as follow:

- An active or passive tag (e.g. RFID or NFC) is attached to an industrial product in the factory. Tags most often contain only a unique identifier for the product.
- Unique identifiers on the tags can afterwards be read anywhere but only in short range by tag reader devices.
• Tag readers send product unique identifiers to online repositories.
• Online repositories contain databases associating URIs to unique identifiers.
• When unique identifiers are provided, the repositories answer with the associated URIs.
• Tag readers receive the URIs and can use them to access online digital information about products.

Automatic identification technologies do not envision embedding computers in mundane objects. Instead, objects are most of the time “read only”, hence enabling them to have an online digital identity. Tagged object like books, clothes, and food cannot react autonomously to changes in the environment nor can they be controlled remotely through the Internet. Research on automatic identification technology is predominantly focused on building infrastructures for the management of product data available during the whole products lifecycle: from raw materials to disposal [16].

Automatic identification technologies are crossing the boundaries of industrial sector and coming to our everyday lives [76]. In the IoT, object identification technologies will be embedded in objects without computation power that are in our homes (e.g. food, furniture) or public places (e.g. monuments). Currently, “manual” identification technologies are already widespread (e.g. bar codes or QR codes) and automatic identification are intended to replace them (e.g. RFID or NFC).

From a high level, the common point between Smart Devices (Ubiquitous Computing) and any other identifiable objects is that both categories have their online digital counterparts. Therefore, we define Things as:

*Thing in the IoT are Smart Devices or identifiable objects with online digital information about their identity and state. Their online digital information is accessible through Internet protocols.*

The previous analysis and the above definition enable to define categories of Things. Two simple categories of Things according to the autonomy of their interaction with online digital information are observed:

1. *Things* with read-write capabilities. This is a category of Smart Devices that can read and write online digital information using Internet protocols.
2. *Things* with read-only capabilities. This is a category of identifiable objects that require some proxy devices in order to access online digital information.
This categorization highlights the characteristics of Things, which are important for our framework by clarifying their role with respect to People and Places. In our framework, Place Masters create Places and register People. They are also the actors that arrange physical space and real world objects that are used. Consequently, the IoT service platforms should mimic real world roles and empower Place Masters to register Things (Figure 17).

Figure 17: People, Places, and Things

Things are initially connected or registered by Place Masters to Virtual Places that map physical places. A virtual representation of Things is created. Afterwards, Place Masters can register Virtual Things in other Virtual Places or Hybrid Virtual Places.

4.5 Marketplaces and Applications in the Internet of Things

In the IoT ecosystem, several service providers are involved: network operators, hardware providers, providers of cloud based services, and application developers. Although their roles are equally important, we focus here on application developers with a particular orientation on open innovation. Current IoT trends are still focused on industry-centered approaches based on “the one sensor one application” model (Figure 18). Such models do not involve third party developers or external software firms in service co-creation thus limiting the potential of IoT. The smart phone ecosystem, as part of the IoT, shows
another approach: an open innovation model with a plethora of software applications that are mostly created by external developers or software firms and almost not at all by hardware providers (Figure 18).

![Figure 18: Application development in IoT vs. smart phone](image)

With the abundance of available software applications, the smart phone ecosystem demonstrates the potential of developers to create new services. The IoT needs a similar approach as the connected sensors, devices, or identifiable objects are almost useless without the associated applications and services. The heterogeneity of things in the IoT (e.g. goods, home appliances, cars, smart homes, etc.) hinders the common standards (e.g. technologies, communication protocols and development tools). The realm of IoT contrasts with the smart phone ecosystem with bigger diversity of hardware platforms, which increases the barrier to entry for development of hardware dependent applications. Moreover, while some Things can be used as hardware platform for software applications (e.g. routers, embedded computers, smart cars, and smart phones) others do not have the technical capabilities to host applications directly (e.g. RFID tagged objects, sensors or actuators). Furthermore, Things also need to be designed to cover a range of interactions among each other, their environments and People. Although software applications can be developed to target specific hardware, the IoT hardware components are used more often only to connect objects to the Internet. Our review reveals that most of the IoT software applications, open or commercial, are created on top of cloud based platforms which expose digital representations of Things [11]. The smart phone ecosystem relies widely on cloud based services. We use the term “cloud based
services” to describe networked software services. The smart phone application market is a cloud based service distributing applications. Application functionalities are extended through external cloud services and the data from smart phones are frequently kept on cloud storage or synchronized through cloud services. For IoT, these digital artifacts are as important as the physical objects themselves and can be even more innovative for service creation. In IoT, this phenomenon is likely to be emphasized with the applications being hosted and used directly on cloud based services without installing them on particular devices. Cheaper network communication and higher transmission rates are favorable to cloud based services to ensure data persistency and portability for applications providing all the advantages of Software as a Service (SaaS) architectures. Moreover, the large amount of data that will be generated by an almost infinite number of connected objects [1] will require a high storage capacity. Application logic will require high computing power in order to analyze this amount of data [77]. The meaningful IoT services will need both: large amounts of data and application logic. Consequently, in the realm of IoT, cloud based services provide a common ground for the creation and distribution of software applications (IoT Apps).

In order to establish an open innovation model, the IoT ecosystem should enable business strategies towards products and services that bring together groups of People in two categories (Figure 19).

![Figure 19: Two-sided network for the IoT](image)

The IoT platforms need to provide infrastructure and rules that enable transactions between People as service users and People as application developers. The two groups are therefore interdependent, thus creating a
network effect between *People*. With such a two-sided network effect, the platform’s value to any given user largely depends on the number of users on the network’s other side. Value grows as the platform matches demand from both sides [58]. Representative examples of platforms for two-sided networks are application marketplaces in the smart phone ecosystem. These platforms ensure that innovation artifacts such as software applications (IoT Apps) are accessible by service users and that service users create an interesting market for application developers. The applications that become available for *People* as service users can then be used on service platforms as helpers in online social networks (Figure 20).

As shown on Figure 20, the software applications are registered by *Place Masters* either in *Virtual Places* or in *Hybrid Virtual Places*. Software applications are in fact, providing services for *People* and *Things* in Virtual Places.

On IoT service platforms, various services can be provided by *Applications*. *Applications* can provide simpler services for reporting and analysis (e.g. electricity consumption, shopping history, etc.), remote controlling of *Things* (e.g. remotely switch off the heating at home), and other. In addition, our model allows *Applications* to build services using online social context and virtual contexts of *People* and *Things*. In this case, more advanced services can be provided by *Applications*. For example: event recommendations according to social preferences, suggestions for events being held at the nearest places,
remote controlling of environments instead of Things, etc. Since Place Masters in Virtual Places register Applications, this empowers developers to create Applications that will not escape social and virtual context. Moreover, Applications registered in several Virtual Places created by a Place Master can provide services across several online social and virtual contexts.

In this section, we defined the concepts of People, Places, Things and Applications in the context of IoT. Figure 21 shows the core components that we will use to design a framework for user empowerment. The next step is to design a general framework using these core elements.

![Figure 21: Framework for User Empowerment and its Elements](image)

### 4.6 Designing the Framework

Our preliminary experiments with open source hardware and smart meters led as to understand that IoT service platforms are the common ground for user empowerment. These platforms, and therefore our general framework, are intended to take into account general requirements for three IoT stakeholders: hardware providers, developers, and users. Our experimentation with IoT hardware shows a high heterogeneity of IoT devices. It is unlikely that developers can have access to all kind of IoT heterogeneous hardware in order to develop and test their applications. Consequently, this leads us to define virtual proxies of Things on IoT service platforms. The virtual representation of Things allows developers to access an API of connected hardware and develop
software applications. The estimated number of heterogeneous devices in IoT also introduces high complexity for users to personalize each Thing individually. Our preliminary experimentation with smart meters confirms this observation. For example, monitoring the electricity usage requires configuration of each individual smart meter and personalization of the associated software application. It requires an important effort in time and knowledge for connecting hardware, analysis of electricity costs, periods of electricity usage, etc. In addition, the ubiquitous nature of IoT requires from users to configure and personalize pervasive Things therefore requiring dedicated time and effort at any place. In order to reduce this complexity, our framework uses virtual proxies of Things that interface with Applications. Created by developers and distributed on IoT service platforms, Applications are intended to simplify this task of hardware configuration and provide additional IoT services. However, the pervasive nature of IoT requires context aware Applications. As a first element of IoT context, we identify the online social networks that define relationships between People. These relationships defined by users provide an input for personalizing IoT Applications and services with social context (e.g. G. Biamino [78]). The second elements of IoT context are Virtual Places. They are proxies of physical places and define the context for interaction between People and Things. They also define the boundaries for Applications that are helpers for interaction between People and Things. Virtual Places are therefore the second element of IoT context available for Applications.

After having analyzed the concepts and defined the core components of our framework: People, Places, Things, and Applications, this section puts together these concepts and defines a model for the interaction of Virtual entities of People, Places, and Things. The aim of the model is to capture the core functionality that will further serve as a basis to provide patterns and guidelines for the design of IoT service platforms. Figure 21 shows these core components of People, Places, Things, and Applications as a general framework. In terms of notation at a high level, we represent this as a tuple of four elements: <People, Places, Things and Applications>. The framework presents an IoT service platform as a medium between physical and virtual worlds in IoT.

Recent trend of online social networks and cloud-based platforms covers the need of IoT service platforms to create and maintain the digital identities of People, Places, and Things. Emerging trends in IoT are also focusing on building the data storage and services that are offered on cloud based infrastructures. The limited or even inexistent processing power of some Things shifts the development of IoT services on cloud-based platforms. Our review revealed that most of the IoT hardware uses cloud based storage and services [11]. Consequently, the IoT service platforms are cloud based platforms that are a medium of interaction between the real and virtual world. Our aim here is to define the model of People, Places, Things, and Applications whose behavior is
mediated through *Virtual Places* on IoT service platforms. We define IoT service platforms as:

The **IoT service platform** is networked hardware and software infrastructure offering services for the creation of digital identities, storage and accessing of digital information for People, Places, Things, and Applications. It includes Virtual Places, online social networks, and Application marketplace. Design of IoT service platform relies on these core components in order to empower People to create virtual contexts for service personalization, create IoT Applications, use, and distribute these Applications.

For modeling *People, Places, and Things*, we consider the concepts as a collection of virtual entities on IoT service platforms. We define these entities as classes having one or more instances. Figure 22 shows a complete example of instances. Let us examine this example by using instances of *<People, Places, Things, and Applications>*. The aim here is to identify the cases that will establish design invariants for modeling interactions of *<People, Places, Things, Applications>* mediated by IoT service platforms.

- We first define classes of *People* as follows:
  - Class *People*
  - Class *Place Master*, with instances PM
  - Class *Virtual Friend*, with instances VF

The class *People* is a root abstract entity representing users and their virtual identities. This class has only general properties (e.g. unique identifier, name, and password) without specific role. The class *Place Master* is a subclass of *People*. It inherits the properties of *People* and adds a role for registering and managing the instances of Virtual *Places and Things*. The class *Virtual Friend* is also a subclass of *People*. It inherits the properties of *People* but its virtual identity acquires the role of *Virtual Friend* depending on *Virtual Place*. For example, the instances of *Virtual Friends* don’t have the rights to modify the properties of a *Place Master ‘s Virtual Things or Virtual Places*. They can contribute to a *Place Master’s* virtual place by registering new *Things* or interacting with the existing digital information. The class *Virtual Friend* represents an online social relationship as discussed in chapter 4.2.2.

- Classes of *Places* are noted as follows:
  - Class *Place*
  - Class *Physical Place*, with instances *P*
  - Class *Virtual Place*, with instances *VP*
  - Class *Hybrid Virtual Place*, with instances *HVP*
The class *Virtual Place* is the root entity mapping physical place. It describes user perception of *Physical Place* and user created *Virtual Place* as discussed in chapter 4.3. As the *Physical Place* has a physical position and embodies the physical and social context (i.e. *People* and *Things*), the class *Virtual Place* has a description, geographic coordinates, and contains a collection of *Virtual Friends* and *Virtual Things*. The *Virtual Place* must be associated with *Physical Place* and should not exist without it. Conversely, the class *Hybrid Virtual Place* only has a virtual representation. Instead of mapping the physical place, this class is a collection of *Virtual Friends* and *Things* independent from a specific physical position. It contains instances of *Virtual Friends* and *Things* that might have location information but the class *Hybrid Virtual Place* is rather mapped to activities that do not depend on one physical location. The purpose of this class is to allow *Place Masters* to manage a network of *People*, *Things*, and *Applications* that will support their online activities independently from physical locations.

- **Classes of Things** in Figure 22 are noted as follows:
  - Class *Thing* with instances $T$
  - Class *Virtual Thing*, with instances $VT$

The class *Thing* is the root class describing connected devices or identifiable object discussed in chapter 4.4. It has general properties for identifying and describing connected objects and their virtual counterparts (e.g. unique identifier, name, and description). Although the instances of *Things* might be enabled to sense location information, the stream of digital information from *Things* is defined by *Place Master* and shared by instances in its *Virtual Places*.

- **Finally, Applications** only have virtual representations and are noted as follows:
  - Class *Application*, with instances $A$

The class *Application* only has virtual instances. It is a helper for the interaction of collections of instances of *People* and *Things*. They are registered by *Place Masters*, in specific *Virtual Places*. *Applications* use *Virtual Things* (proxies) in order to interact with physical *Things*.

Any virtual instance of classes *People* and *Things* can belong to several instances of *Virtual Places*. For example, an instance of *Virtual Thing* $VT$ can be registered in the instance *Virtual Place* $VP$ and in *Hybrid Virtual Place* $HVP$. That allows *Place Masters* to build multiple virtual contexts on IoT service platforms according to their selection of *People* and *Things*.

The instances in Figure 22 are an example of our framework. In the next section, we will analyze interactions between these instances. The study of this example will allow deriving patterns for <People, Places, Things, Applications> to be applied in the design of IoT service platforms.
Figure 22: General Framework: Instances of People, Places, Things, and Applications
4.6.1 Defining the model of *People, Places, Things and Applications*

In order to show the behavior of the entities *People, Places, Things, and Applications* within our framework, we introduce a minimal set of primitives as “verbs” for interaction in the framework. We define two categories of verbs: verbs for communication with virtual instances on IoT service platform (i.e. proxies) and verbs for interaction between instances of *<People, Places, Things, Applications>*. This grouping helps define communication rules of physical entities with IoT service platform as well as rules for interactions between virtual instances. We do not take into account network protocols that should allow physical entities to communicate with IoT service platforms. We assume that the IoT infrastructure will allow the hard link connection between real world entities and their virtual counterparts. This means that we assume that the actions of physical *Things* are forwarded to their virtual counterparts and vice versa. For example, let us consider an Internet enabled device with remote control abilities. When the state of the device changes, the device sends a notification to its online instance and when its online instance changes its state, it notifies the physical device to change the state. Communication verbs for IoT service platforms have already been suggested by recent research on the Web of Things [41]. However, this proposal is based on HTTP verbs that are limited to network protocols and web architecture as defined by R. Fielding and R. Taylor [37]. We choose to define a higher abstraction of communication verbs that are better adapted to our framework and therefore help capture and express the richness of the IoT empowerment framework. In addition, communication verbs that are independent from underlying network protocols allow more flexibility and independence for further evolution of IoT service platforms.

Verbs for the communication of physical entities with IoT service platforms are defined as follows:

- **READ (Virtual Things, People, Virtual Places)** for retrieving digital information. Any instance of classes *People, Things, and Application* can use this communication verb.

- **SEND (Virtual Things, People, Virtual Places)** for submitting digital information to IoT service platform. Any instance of classes *People, Things, and Application* can use this verb in order to send digital information.


Virtual Things, and Applications in Virtual Places. Only the instances of subclasses of People can use this verb.

Verbs for interaction between virtual entities on IoT service platform are defined as follows:

- **NOTIFY (People, Virtual Things, Applications)** for interaction between instances of People, Virtual Thing, and Applications in a collection of Virtual Places. It assumes a hard link between Things and their proxies.

- **UPDATE (People, Virtual Things)** for updating digital information. Any instance of classes People, Things, and Application can use this verb in order to update its own status. In addition, an instance of Application or a Place Master can use it in order to update the state of Virtual Thing.

- **DELETE (People, Virtual Things, Virtual Places, Applications)** for deleting digital information. Only the instances of subclasses of People can use this verb in order to delete digital information within their Virtual Places.

We also define conditions under which physical and virtual entities can communicate and interact. In order to show the use of core components and interaction verbs, we define several use cases. As suggested by I. Jacobson [79], use cases are adapted for object oriented software engineering and will help us define interactions between classes and instances on IoT service platform. For each use case, we defined preconditions to be satisfied. Optionally, for some use cases we also define post actions (i.e. includes). We can now use the example shown in Figure 22 to demonstrate use cases and describe them using our interaction verbs.

4.6.2 Design

Before providing a definition of general use cases, we group them into three categories:

1. Creation of Virtual Places
2. Personalization of Virtual Places
3. Interactions of <People, Things, Applications> within Virtual Places

These three categories of use cases match our objective to empower users to create, personalize, and control their IoT ecosystem. Creating Virtual Places is a prerequisite for personalization of virtual contexts and deployment of IoT services. Personalization of Virtual Places consists in registering Things, Applications and online social relationships. Thus, personalized services can be provided by Applications within user-defined virtual contexts. Use cases for interactions in Virtual Places define the extent of user control over <People, Things and Applications> within his or her personalized virtual context.

A general precondition for all use cases is that the actors are registered on IoT service platform.
Creation of *Virtual Places*

These use cases show how to use general framework in order to empower users to prepare virtual contexts. By defining *Virtual Places*, users create virtual contexts for further interactions between online social networks and *Things*.

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Create Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Masters</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM creates Virtual Place</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Virtual Place VP is registered on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>REGISTER (VP, name, description, location)</td>
</tr>
<tr>
<td><strong>Assumptions:</strong></td>
<td>Physical Place that corresponds to the location of VP exists</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Create Hybrid Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Masters</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM creates Virtual Place</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Hybrid Virtual Place HVP is registered on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>REGISTER (HVP, name, description)</td>
</tr>
</tbody>
</table>

Personalization of *Virtual Places, Shared Virtual Places, and Hybrid Virtual Places*

These use cases show how to use general framework in order to empower users to personalize virtual contexts. By registering *People, Virtual Things* and *Applications*, users personalize context for interactions between these entities. In particular, they personalize the stream of data generated by online social networks and *Things* used by *Applications* in order to provide services. Consequently, it empowers users to personalize input for socially aware IoT services.
<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Registering &lt;People, Things, Applications&gt;</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Masters, Virtual Friends, Virtual Things, Applications, and Virtual Places</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM registers Virtual Friend VF, Virtual Thing VT, and Application A in Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VP is registered by PM</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>VF, VT, and A are registered in VP</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>REGISTER (VP, VF, VT, A)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Unregistering &lt;People, Things, Applications&gt;</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM, Virtual Friend VF, Virtual Thing VT, Application A, and Virtual Place VP</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM unregisters Virtual Friend VF, Virtual Thing VT, and Application A from Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VP is registered by PM</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>VF, VT, and A are unregistered from VP</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>UNREGISTER (VP, VF, VT, A)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Read or update or delete digital information in a Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM, Virtual Friends VF&lt;sub&gt;1-n&lt;/sub&gt;, Virtual Things VT&lt;sub&gt;1-n&lt;/sub&gt;, Applications A&lt;sub&gt;1-n&lt;/sub&gt;, and Virtual Place VP</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM reads, updates or deletes the digital information associated to virtual identities of Virtual Friend VF, Virtual Thing VT or Application A from Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VP is registered by PM</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Content is presented to PM or updated or deleted on IoT service platform</td>
</tr>
</tbody>
</table>
| **Normal Flow:** | READ (VP, VF<sub>1-n</sub>, T<sub>1-n</sub>, A<sub>1-n</sub>, content)  
UPDATE (VP, VF<sub>1-n</sub>, T<sub>1-n</sub>, A<sub>1-n</sub>, content)  
DELETE (VP, VF<sub>1-n</sub>, T<sub>1-n</sub>, A<sub>1-n</sub>, content) |
<p>| <strong>Notes and Issues:</strong> | READ (VP, VF&lt;sub&gt;1-n&lt;/sub&gt;, T&lt;sub&gt;1-n&lt;/sub&gt;, A&lt;sub&gt;1-n&lt;/sub&gt;) |
|                | Can be used without content parameter by PM in order to get description of virtual identities |</p>
<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[6]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Virtual Friend Registering &lt;Things, Applications&gt;</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Masters, Virtual Friends, Virtual Things, Applications, and Virtual Places</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Virtual Friend VF registers Virtual Thing VT and Application A in Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VF is registered in VP by PM</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>VT and A are registered in VP</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>REGISTER (VP, VT, A)</td>
</tr>
<tr>
<td><strong>Notes and Issues:</strong></td>
<td>Any Virtual Friend VF registered by a Place Master PM in Virtual Place VP can register Virtual Things VT and Applications A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[7]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Virtual Friend reads digital information from Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Virtual Friend VF, Place Master PM, Virtual Friends VF&lt;sub&gt;1-n&lt;/sub&gt;, Virtual Things VT&lt;sub&gt;1-n&lt;/sub&gt;, Applications A&lt;sub&gt;1-n&lt;/sub&gt;, and Virtual Place VP</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Virtual Friend VF reads Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VF is registered in VP by PM, VF&lt;sub&gt;1-n&lt;/sub&gt;, VT&lt;sub&gt;1-n&lt;/sub&gt; and A&lt;sub&gt;1-n&lt;/sub&gt; are registered in VP</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Digital identities (description) and associated digital information are presented to VF</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>READ (VP, PM, VF&lt;sub&gt;1-n&lt;/sub&gt;, T&lt;sub&gt;1-n&lt;/sub&gt;, A&lt;sub&gt;1-n&lt;/sub&gt;)</td>
</tr>
<tr>
<td><strong>Notes and Issues:</strong></td>
<td>Any Virtual Friend VF registered by a Place Master PM in Virtual Place VP can read all instances in that Virtual Place</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[8]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Virtual Friend updates or deletes digital information from Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Virtual Friend VF, Virtual Things VT&lt;sub&gt;1-n&lt;/sub&gt;, Applications A&lt;sub&gt;1-n&lt;/sub&gt; and Virtual Place VP</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Virtual Friend VF updates or deletes digital information from Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VT&lt;sub&gt;1-n&lt;/sub&gt; and A&lt;sub&gt;1-n&lt;/sub&gt; are registered in VP by VF</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Digital identities (description) or associated digital information are updated or deleted</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>UPDATE (VP, VT&lt;sub&gt;1-n&lt;/sub&gt;, A&lt;sub&gt;1-n&lt;/sub&gt;)</td>
</tr>
<tr>
<td><strong>Notes and Issues:</strong></td>
<td>Virtual Place VP can read all instances in that Virtual Place</td>
</tr>
</tbody>
</table>

**DELETE** (VP, VT<sub>1-n</sub>, A<sub>1-n</sub>)
Interactions in *Virtual Places, and Hybrid Virtual Places*

These use cases show how to use general framework in order to empower users to control interactions within virtual contexts. We focus on interactions that allow *People* to communicate and on interactions that allow *Things* and *Applications* to provide IoT services. As a result, users are empowered to control these interactions, benefit from socially aware and context aware services provided by *Applications*.

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[9]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Place Master sends digital information to instances in Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM, Virtual Friends VF(<em>{1-n}), Virtual Things VT(</em>{1-n}), Applications A(_{1-n}), and Virtual Place VP</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM sends digital information to instances in Virtual Place VP (e.g. sends a message)</td>
</tr>
</tbody>
</table>
| **Preconditions:** | VP is registered by PM  
VF\(_{1-n}\), VT\(_{1-n}\) and A\(_{1-n}\) are registered in VP by PM |
| **Post conditions:** | Digital information log is recorded on IoT service platform |
| **Normal Flow:** | SEND (VP, VF\(_{1-n}\), VT\(_{1-n}\), A\(_{1-n}\)) |
| **Includes:** | If an Application receives digital information, the use case “Application provides service [13]” is included |
| **Assumptions:** | Sending digital information to VF\(_{1-n}\) or VT\(_{1-n}\) assumes that these proxies will transmit it to their physical counterparts (e.g. verb SEND can be used by proxies) |

<table>
<thead>
<tr>
<th>Use Case ID:</th>
<th>[10]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name:</strong></td>
<td>Virtual Friend sends digital information to instances in Virtual Place</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM, Virtual Friends VF(<em>{1-n}), Virtual Things VT(</em>{1-n}), Applications A(_{1-n}), and Virtual Place VP</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Virtual Friend VF sends digital information to instances in Virtual Place VP (e.g. sends a message)</td>
</tr>
</tbody>
</table>
| **Preconditions:** | VP is registered by PM  
VF is registered in VP by PM  
VF\(_{1-n}\), VT\(_{1-n}\) and A\(_{1-n}\) are registered in VP by PM |
| **Post conditions:** | Digital information log is recorded on IoT service platform |
| **Normal Flow:** | SEND (VP, VF\(_{1-n}\), VT\(_{1-n}\), A\(_{1-n}\)) |
| **Includes:** | If an Application receives digital information, the use case “Application provides service [13]” is included |
| **Assumptions:** | Sending digital information to VF\(_{1-n}\) or VT\(_{1-n}\) assumes that these proxies will transmit it to their physical counterparts (e.g. verb SEND is used by proxies) |
### Use Case 11

**Use Case ID:** [11]

**Use Case Name:** Thing sends digital information to Virtual Place

**Actors:** Thing T, Virtual Thing VT, Virtual Place VP, Applications A<sub>1-n</sub>

**Trigger:** Thing T sends digital information to Virtual Place VP

**Preconditions:**
- VT is registered in VP
- Applications A<sub>1-n</sub> are registered in VP

**Post conditions:** Digital information log is recorded on IoT service platform

**Normal Flow:**
- Thing T: SEND (VP, VT)
- Virtual Thing VT: NOTIFY (VP, A<sub>1-n</sub>)

**Includes:** If an Application receives digital information, the use case “Application provides service [13]” is included

The use case 11 shows an example of communication between Things and IoT service platform. First, *Thing T* sends digital information to *Virtual Thing VT* (i.e., its proxy). Second, its proxy notifies Applications registered in the same Virtual Place. Finally, Applications can provide a service by communicating with People and Things through their proxies.

### Use Case 12

**Use Case ID:** [12]

**Use Case Name:** Application reads Virtual Place

**Actors:** Place Master PM, Application A, and Virtual Place VP

**Trigger:** Application A requests contextual information from VP

**Preconditions:**
- VP is registered by PM
- Application A is registered in VP by PM
- VF<sub>1-n</sub>, VT<sub>1-n</sub> and A<sub>1-n</sub> are registered in VP by PM

**Post conditions:** Virtual identities and associated digital information are presented to Application A

**Normal Flow:** READ (VP, PM, VF<sub>1-n</sub>, VT<sub>1-n</sub>, A<sub>1-n</sub>)
Use Case ID: [13]

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Application provides service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM, Application A, and Virtual Place VP, Virtual Friends VF_{1-n}, Virtual Things VT_{1-n}</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Application A sends digital information to instances in Virtual Place VP</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>VP is registered by PM</td>
</tr>
<tr>
<td></td>
<td>Application A is registered in VP by PM</td>
</tr>
<tr>
<td></td>
<td>VF_{1-n} and VT_{1-n} are registered in VP by PM</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Notification log is recorded on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>READ (VP, PM, VF_{1-n}, VT_{1-n})</td>
</tr>
<tr>
<td></td>
<td>SEND (PM, VF_{1-n}, VT_{1-n})</td>
</tr>
</tbody>
</table>

**Assumptions:** Sending digital information to VF_{1-n} or VT_{1-n} assumes that these proxies will transmit it to their physical counterparts (e.g. verb SEND can be used by proxies)

Let us clarify the title of the use case 13: “Application provides service”. Application provides IoT services, for example use information about ambient temperature and take corresponding actions for energy saving. A Thing (e.g. sensor) with the measure of ambient temperature would notify this Application. It could “read” the Virtual Place and determine the context (e.g. if a person is at home) and take corresponding action (i.e. provide service). This action could be sending instructions to other Things (e.g. actuators) to adjust home appliances.

The framework that defines behavior of <People, Places, Things and Applications> is independent from underlying technologies. It is intended to be used as a reference framework when implementing user centered IoT service platforms. The next section aims at presenting this framework through an illustration.

4.6.3 Illustration

In order to show how the above design can be applied, this is an example with concrete instances of <People, Places, Things and Applications> (Figure 23).

- First, the classes of People are defined as follows:
  - Class Place Master, with instances PM1, PM2, and PM3
  - Class Virtual Friend, with instances VF1 and VF3
- The classes of Places are noted as follows:
  - Class Place
  - Class Physical Place, with instances P1, P2, and P3
o Class Virtual Place, with instances VP1, VP2, and VP3
o Class Hybrid Virtual Place, with instance HVP1
- The classes of Things in the Figure 23 are noted as follows:
  o Class Thing with instances T1, T2, and T3
  o Class Virtual Thing, with instances VT1, VT2, and VT3
- Finally, Applications only have virtual representations and are noted as follows:
  o Class Application, with instances A1, A2, and A3

The Figure 23 shows three simple scenarios, each of them represents the concepts defined in our framework.
Figure 23: Illustrative Scenario for General Framework
It allows us to analyze the behavior of People, Things, and Applications in Virtual Places defined as VP1 and VP3, Virtual Place VP2, and Hybrid Virtual Place HVP1. The instances of Virtual Place presented in figure 23 are following:

- **Virtual Place VP1** is proxying \(<P1, PM1, T1>\). It is a collection of following instances:
  \(<VPM1, VT1, A1>\)

- **Virtual Place VP2** is proxying \(<P2, PM2, T2>\). It is a collection of following instances:
  \(<VPM2, VT2, A2, VF1>\)

- **Hybrid Virtual Place HVP1** is a virtual collection of following instances:
  \(<VPM1, VT1, VT3, VF3>\)

To illustrate this example we will only show few representative use cases. The entire listing of use cases can be found in Appendix 2.

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Place Master creates Virtual Place</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM1 registers Virtual Place</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Place Master PM1 is registered on IoT service platform</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Virtual Place VP1 is recorded on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>REGISTER (PM1, name, description, location)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Place Master Registers Thing to Virtual Place</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM1, Virtual Place VP1, Thing T1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM1 registers Thing T1 in Virtual Place VP1</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Place Master PM1 is registered on IoT service platform Virtual Place VP1 is created by Place Master PM1</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Virtual Thing VT1 is created</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong></td>
<td>REGISTER (PM1, VP1, T1)</td>
</tr>
<tr>
<td>Use Case Name</td>
<td>Place Master Registers Application to Virtual Place</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM1, Virtual Place VP1, Application A1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM1 registers Application A1 in Virtual Place VP1</td>
</tr>
</tbody>
</table>
| **Preconditions:** | Place Master PM1 is registered on IoT service platform  
Virtual Place VP1 is created by Place Master PM1 |
| **Normal Flow:** | REGISTER (PM1, VP1, A1) |

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Place Master Registers Virtual Friend to Virtual Place</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM1, Virtual Place VP1, Virtual Friend VF1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM1 registers VF1 in Virtual Place VP1</td>
</tr>
</tbody>
</table>
| **Preconditions:** | Place Master PM1 is registered on IoT service platform  
Virtual Friend VF1 is registered on IoT service platform  
Virtual Place VP1 is created by Place Master PM1 |
| **Post conditions:** | Virtual Thing VT1 is created |
| **Normal Flow:** | REGISTER (PM1, VP1, VF1) |

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Thing sends digital information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Virtual Thing VT1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Thing T1 sends digital information to IoT service platform</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Virtual Thing VT1 is registered on IoT service platform</td>
</tr>
<tr>
<td><strong>Post conditions:</strong></td>
<td>Digital information log is recorded on IoT service platform</td>
</tr>
</tbody>
</table>
| **Normal Flow:** | SEND (VT1, content)  
NOTIFY (A1)  
NOTIFY (HVP1, VT1)  
NOTIFY (HVP1, A3, A4) |
Using this concrete example, we examined use cases for instances of \(<People, Places, Things, Applications>\) that are mediated through IoT service platform. We also showed how the framework could be used.

### 4.7 Summary

In this chapter, we analyzed important groups of stakeholders in the IoT. Although the IoT hardware providers are important for establishing the enabling infrastructure for creation of IoT services, we mainly focused on empowerment of two user profiles: end users and developers. We defined the physical and virtual entities of \(<People, Places, Things, Applications>\) that should be taken into account for the design of IoT service platforms promoting empowerment of end users and developers. Based on existing literature, we analyzed the concepts of \(People, Places, Things, and Applications\). We defined each concept and their virtual counterparts. Based on these concepts, we defined an instance of our general framework. This instance allowed us to show the use cases of our general framework. A thorough exploration of use cases allowed us to describe design elements of IoT service platforms that empower users to create their virtual contexts, personalize this context with their online social relationships, and benefit from services (e.g. software applications) that can be personalized with the virtual context. We also illustrated this with an example on how to apply the use cases of our framework.

Design elements defined in this chapter are independent from underlying technologies. They are intended to serve as guidelines for implementing IoT service platforms. In the next chapter, our objective is to implement design elements into a platform prototype that validates our general framework. We will define requirements for implementation, choose the underlying technologies, and test the implementation of our general framework.
Chapter 5. **Platform Prototype**

In this chapter, we will use design elements from our framework in order to implement an IoT service platform. We will define a working scenario that allows for a definition of system requirements. The system requirements will provide elements for architectural strategies and implementation. Our aim here is to implement the core component from our general framework in order to test and demonstrate the feasibility of the implementation of our proposal.

5.1 **Scenarios**

5.1.1 **The Model of People, Places, Things, and Applications in the working Scenarios**

In order to support our model we defined scenarios that describe situations and identify further system requirements. The first scenario is based on one user creating his IoT environment and context of use. The second scenario describes the developer persona distributing software application. The third scenario describes a hardware provider as a stakeholder rather than as a persona. We will present the benefits for hardware providers using our approach while remaining focused on users. We assume that the IoT service platform will be provided by hardware providers although it could also be provided by third party platform providers. Our model of *<People, Places, Things, Applications>* is applied furthermore in order to define design element of IoT platforms.

The scenario: *Friendly-TV Passion.*

**Persona: service user**

John buys an IoT enabled TV (IoTV) and brings it home. The IoTV automatically connects to the Internet and uploads the usage information to an IoT cloud platform. John does not want to share this information with all of his online social relationships, but only with Alice and Harry. Sharing of usage information allows him to get the opinions and recommendations related to the TV shows from them. His IoTV also becomes more “social” and allows him to recommend TV shows to Alice and Harry. In addition, John would like to use a software application that would analyze usage of his IoTV and similar devices of Alice and Harry and recommend new TV shows personalized for them. However, although John uses an IoTV at work for other purposes, he does not want Charles, his supervisor, to know which TV shows he is watching at home.

In this scenario, John owns an IoTV (*Thing*). When the IoTV is connected to the Internet, it starts uploading the usage information to IoT service platform. Digital information from IoTV is sent to John’s *Virtual Place*. In this example, we assume that John had previously created a *Virtual Place* and called it *Home*. 
John wants to create social events while watching the IoTV. His online social relationships (Virtual Friends) should not get the digital information about other Things in John’s Home. John creates a virtual context for social interactions called for example Friendly TV Passion (Hybrid Virtual Place). In this case, John has a full control over Friendly TV Passion (he becomes Place Master). He must register the IoTV from Home (Virtual Thing from Virtual Place) to Friendly TV Passion. The next step is to register Alice and Harry (Virtual Friends) to Friendly TV Passion. Friendly TV passion now holds the virtual context for social interactions between John and his friends. However, John would like more than the interactions with his friends, he would like to get suggestions about new TV shows. Suggestions for new TV shows should be adapted to his preferences but also according to the preferences of his online social relationships. In order to get these recommendations John registers a software application (Application) to the Friendly TV Passion. The application makes suggestions about new TV shows but it does not share these suggestions with John’s office. Assuming the application that John registered in Friendly-TV Passion is compatible with IoTV in his Home and in his Office, it should not outflow the Home context and send suggestions to the IoTV in the office. Therefore, John’s supervisor, Charles, is not informed about John’s private life.

Persona: developer

Eric is a web programmer. He wants to create and distribute an IoT enabled software application. He would like to maximize the number of users that will use his application. He registers to IoT software platform and searches for the number of connected Things. He notices an interesting number of Smart TVs and he has an idea to create an application for rating and suggesting TV shows with respect to the social preferences of users. He reads the information on IoT services platform that describes how his application will be used and what is the virtual context that his application can use. He notices that an application will be installed by users in their virtual places and shared with their social networks within the virtual places. He develops a software application according to that specification. He does not have to worry about which social relationships are appropriate to comment and to receive suggestions because users will define their social context. To develop his application, Eric does not have to buy a Smart TV. He can look up on IoT service platform which hardware is connected and which API to use in order to create compatible software applications. Once Eric finishes his application, he registers it on the application store provided by the IoT service platform. Users can now “install” his application in virtual places and start using it.

In this scenario, the web based IoT service platform helps developers create software applications. Although using web technologies is not required, it empowers developers in creating IoT enabled application using well-established programming languages and protocols. In order to empower developers to
maximize the number of users, the IoT service platform presents the API of connected hardware in an anonymous way. This means that the information about specific hardware users does not need to be presented to developers. The IoT service platform exposes an API allowing developers to use the social graph defined by users in virtual places. The IoT service platform provides application markets or application registries that empower developers to distribute their applications. Developers register applications in application registries and users search for applications and register applications in their virtual places.

**Stakeholder: hardware provider**

Smart TV Pro is a hardware provider of Internet enabled TV hardware. The company is specialized in hardware manufacturing and in order to focus on their manufacturing process, they do not need to spread their efforts on a large offer of software components. They want to benefit from software applications created by external developers and third party software firms in order to provide additional services to their customers. The company creates a Smart TV that is easy to connect (i.e. in plug and play) to IoT service platforms. As additional service, they develop an “official” software application for the Smart TV. This application reminds their customers about the warranty expiration date and about nearby services. They can now distribute the Smart TV hardware knowing that the more they sell, the more third party software applications for that model will be available on IoT services platform. They can soon have an additional sales argument for their customers including many software applications and services.

Although the hardware provider is an important stakeholder in the IoT ecosystem, we remain focused on users and developers. This scenario shows the interest of hardware providers to improve their hardware, focus on main applications, and benefit from applications as additional services created by external developers.

From these three scenarios, we define the functional requirements for IoT service platforms. We assume that the progress in standardization such as Internet of Things (IoT) REST API [80] and ISO RFID standards [81] will enable common plug-and-play protocols (e.g. RESTful Plug and Play Experience [55]) for connecting Things to IoT cloud based platforms. We also assume that the IoT enabled devices (i.e. Things) do not require users to program, prototype, or have any technical knowledge on how to connect Things to the IoT service platforms. We further assume that the Internet link between People, Things and IoT service platforms rely on secure connections. We focus here on textual messages as types of digital information exchanged by People, Things, and Applications. Simplifying the type of digital information allows to efficiently implement our general framework. Based on this, we can now focus on defining the functional requirements for our IoT service platform prototype.
These scenarios imply following functional requirements that the system must fulfill:

1. When John brings home his IoTV, it should be able to connect to the Internet in a plug and play way.
2. In order to participate in John’s online interactions, the IoTV should be enabled to send autonomously standardized messages to an IoT service platform with or without John’s intervention.
3. The IoT platform should store messages from John’s IoTV to a collection of messages owned by John.
4. In order to allow John (Place Master) to create Virtual Places and register People, Things, and Applications, the IoT service platform must feature a graphical user interface.
5. The IoT platform must have a registry and search functions for Virtual Places, Virtual Friends, Virtual Things, and Applications.
6. It is essential that the IoT platform allow John to create Virtual Places and associate them to physical location, in this case Home and Office.
7. Virtual Places on the platform must mediate the exchange of messages only between virtual identities of People, Virtual Things, and Applications that it contains.
8. The IoT platform must allow John to create Hybrid Virtual Places, in this case Friendly TV Passion. The Hybrid Virtual Places on the platform must mediate the exchange of messages between People, Things, and Applications. In this case, John, Alice, Harry, and one Application can exchange text messages.
9. The IoT platform must feature an online social network. The users (People) must register in order to access the platform. It also has to store the links between social relationships (i.e. social graph). The social graph might be defined at a global level between People, but the platform must allow creating social graphs within Virtual Places. The system should allow John to choose Virtual Friends that can see his IoTV usage.
10. John should be allowed to choose software applications that he wants to use with his IoTV. Although the IoTV provider could create recommendation software, an application store would provide other third party applications and services to John.
11. In order to empower developers to easily create software applications IoT service platform should be designed using web-based technologies.
12. IoT service platform should expose an API allowing application to access the social graph of users that developers can use in order to create socially aware applications.
13. IoT service platform should allow access to social graph in the Virtual Place only if the Place Master registers the application in this Virtual Place.

14. IoT service platform should expose an API allowing developers to create applications using Virtual Places (i.e. virtual context).

15. IoT service platform should allow access to Virtual Places only if the corresponding Place Master registers the application in this Virtual Place.

These functional requirements lead us to the system requirements that will define the system behavior and architecture in order to fulfill the required functionalities.

5.2 System Requirements

The scope of system requirements will cover the software architecture. Our aim here is to define system requirements for software architecture needed to implement our framework into an IoT service platform. However, we take into account portability and scalability of software assuming that the underlying hardware can be chosen according to the system load.

![Diagram of System Requirements](image-url)
We attempt to define requirements for a software prototype of IoT service platforms that leaves the possibility to be ported or scaled over various underlying hardware architectures. System requirements could be divided into two main software layers: application and persistent storage. The application layer contains the use cases defined for classes and instances of *People, Places, Things, and Applications*. The persistent storage layer stores the digital information needed for application logic (Figure 24). Each component and its sub component must be layered. In Figure 24, each entry to the system must start from the top layer (Authentication) and must be treated by at least one component on each layer. System requirements for implementation of our framework are further detailed by component.

1. **Application**
   1.1. **Authentication.** The system must have an authentication layer. The authentication layer must feature user authentication and provide authentication through an Application Programming Interface (API) for *Things* and *Applications*. The user interface must allow users to register on the platform. User authentication must allow registered users to authenticate using their login and password. It must also provide password reminder functionality. The authentication for accessing the API module must not use authentication by login and password. The system must prevent *Applications* and *Things* to use user credentials. Applications and Things must use a separated authentication mechanism based on access tokens. The access tokens must be unique for each Application or Thing. Both authentication mechanisms should be implemented over a secure network layer (e.g. Secure Sockets Layer – SSL). Software modules that are below authentication must be accessible only after authentication and should be used over secured channels.

   1.2. **Things and Applications API.** The system must feature a software layer for connecting hardware components over the Internet. Hardware components can be IoT enabled devices (*Things*). It should also allow external software components i.e. *Applications* that are executed on networked computers to access digital information stored on the IoT service platform. This component of the system must implement the communication verbs READ and SEND as defined in chapter 4.6.1.

   1.3. **User Interface.** User interface must allow *Place Masters* to create *Virtual Places* and *Hybrid Virtual Places*. In addition, it must provide search function for *Virtual Friends, Things*, and *Applications*. It must provide an option to register *Virtual Friends, Things*, and *Applications* to any *Virtual Place* of a *Place Master*. It must also provide a possibility to remove *Virtual Friends, Things*, and *Applications* from any *Virtual Place* of a *Place Master*. The user interface must allow *Place Masters* to see
and monitor the stream of messages exchanged between People, Things, and Applications. It must allow Place Masters to send messages to Virtual Friends, Things, or Applications. The user interface must allow Place Masters to delete their Virtual Places. The user interface must as well allow developers to register Applications, search for Things connected to the IoT service platform and display their API. The Things displayed to developers must not contain any personal information about Place Masters or People. This software component must implement communication verbs REGISTER and UNREGISTER as defined in chapter 4.6.1.

1.4. Access Control List (ACL) Manager. ACL manager must be a software layer invoked on requests from either user interface or Things and Applications API. It must allow Place Masters to have full control over the virtual instances of Virtual Places, Virtual Friends, and Virtual Things that they register. It must filter and distinguish the role of People in each Virtual Place (i.e. Place Master vs. Virtual Friend). It must equally prevent all People, Things, and Applications to read, update or delete any digital information from the virtual places where these instances are not registered by Place Masters. It must be designed to uniquely identify any Thing or Application by reading their access tokens and finally, it must be designed to uniquely identify People based on their access credentials (i.e. login).

1.5. Socialization Manager. Socialization manager must be a software layer designed to maintain social networks of People, Places, Things, and Applications. It must be designed as a registry for these components. This software layer must map the instances of People, Places, Things, and Applications from and to the persistent storage. Socialization Manager has to implement the interaction verbs NOTIFY, UPDATE, and DELETE as defined in chapter 4.6.1. It can be used only after passing the authentication, user interface, and ACL layers. It must execute required actions received from user interface in order to record relationships between People, Places, Things, and Applications as defined by Place Masters. This component must generate the stream of messages from persistent storage to the user interface or Things and Applications API. It must also receive the messages from the user interface or Things and Applications API and associate them to the Virtual Places.

2. Persistent Storage

2.1. Collections of People, Places, Things, and Applications. The persistent storage is a database management system. It stores the following data: People, Places, Things, and Applications profiles, messages, Virtual Places and access credentials. It uses at least one database for access credentials and at least one for collections of
instances and messages. Adequate database management system must be chosen in order to be scaled on networked hardware. It should provide high availability and support for big data storage. Collections of People, Places, Things, and Applications must be logically arranged in order to provide consistent and accurate data to the software components in higher layers of the system. Ideally, the data should be stored in the same format as it could be delivered through the Things and Applications API in order to avoid unnecessary data format conversion that could increase system load.

These system requirements define the behavior of the system and its underlying components. They do not address the technical requirements, system performance, or hardware architecture. At this stage, the focus is on a software prototype that should lead to the testing of the framework’s concepts. Architectural choices will be defined according to system requirements.

### 5.3 Implementation

#### 5.3.1 Architectural Strategies

For the system’s implementation, we use Web based architecture. Although the choice of technologies is large, we follow the Web of Thing approach [41] for system prototyping. This approach allows for rapid development and prototyping. It requires a low learning curve for developers that will reuse and extend the software prototype in order to test their ideas. Since humans use the Web for communication, our aim is to bring Things to the same level. The goal is to use the same technologies for all the concepts of our framework: People, Places, Things, and Applications.

There are several communication protocols that could be used for our system’s implementation. The Message Queuing Telemetry Transport [82] protocol aims at enabling machine-to-machine (M2M) communication. It was designed as an extremely lightweight publish/subscribe protocol. Constrained Application Protocol (CoAP) [83] is a web transfer protocol for use with constrained devices and networks. The protocol is also designed for machine-to-machine (M2M) applications such as smart energy and building automation. IoT6 project [84] attempts to use the potential of IPv6 to overcome current shortcomings and fragmentation of the IoT network protocols. While Message Queue Telemetry Transport (MQTT) or CoAP are lightweight and adapted for constrained devices, the IPv6 could offer two unique advantages addressing and identifying Things. However, the performance of communication protocols is not the focus of this work. In addition, IPv6 is not widely deployed yet. Our aim is to empower users to create services on IoT service platforms and therefore a higher level of communication protocol should be used. The Web is currently the most used tool for communication between humans and we aim at bringing Places and
Things to equal level. Our objective is also to avoid creating yet another middleware for integration of Things. Consequently, our architectural strategy for the implementation of IoT service platforms focuses on the Web architecture.

Common Web architectures include web server (or application server), dynamic language module, and database management system. For the purpose of our work, we use the Apache web server for its robustness, portability, and reliability. The web server can easily be replaced by other software as long as this software can treat HTTP request and forward it to our application. Several lightweight servers can be used instead (Nginx, Lighttpd, etc.).

In order to implement our framework we used the PHP programming language. It is a widely used web scripting language. Its simplicity allows for rapid prototyping and its embedded modules can be used to implement communication verbs defined in chapter 4.6.1. In addition, a large number of available frameworks and libraries facilitate rapid prototyping. Particularly, we used two libraries in order to implement our system: ion-auth [85] and rest-server [86]. The first library was used for user authentication and the second provided the foundation for building Things and Applications API. For the implementation of our application, we used CodeIgniter [87] framework that is a proven, agile and open PHP web application framework with a small memory footprint.

For persistent storage, we used the MySQL relational database in order to implement the storage of digital identities of People and for the authentication module. For the data collected by digital identities of <People, Places, Things, Applications> we used NoSQL database (approach [88]). NoSQL databases provide lightweight mechanisms for storage and retrieval of large amount of data. These systems provide a high scalability and availability for our persistent storage needs. The NoSQL “data stores” rely on models that allow for horizontal scaling of our IoT service platform. The scaling can be achieved by organizing “nodes” of hardware and software in distributed network where data is distributed over these “nodes”. This model allows for the addition of “nodes” when needed to increase the system performances and automatic rebuilding of the data in case of failure of a “node”. The flexible data structure of NoSQL databases is highly optimized for data recording and retrieval but offer little functionality beyond record storage. Consequently, implementing our model requires leveraging the interaction model of <People, Places, Things, Applications> at the application level. Finally, it will lead to the implementation of loose coupling between the application and persistent storage in order to design a system that is minimally dependent on specific database management system. For our implementation we chose MongoDB [89] for persistent storage. It is a document-oriented NoSQL database system. Although other systems are available, MongoDB presents several advantages that match the requirements of the implementation of our framework. Data is stored in Binary Json format.
(BSON) and the widely used data exchange format on the Web is Javascript Object Notation (JSON). That allows minimal data conversion or formatting between the application’s API and the storage. As many NoSQL databases, MongoDB is a distributed database management system and offers high scalability. The functions such as Map – Reduce [90] are supported natively and provide high performances for manipulating large amounts of data. Collections in MongoDB allow for adding properties in document objects without the need for predefined storage types of properties. Therefore, MongoDB covers our requirement for less structured data storage due to the high diversity of Things properties. In addition, MongoDB features native geospatial support [91]. It simplifies the implementation of our concepts that have an associated geographical position.

In the next section, we will describe the system architecture, based on architectural strategies and system requirements.

### 5.3.2 System Architecture

The choice of system components as shown in Figure 25 provides the input for our system architecture. In this part, we detail this architecture and the responsibilities of its components. Figure 25 details the system according to architectural strategies and system requirements. It shows the stack of software components of IoT service platform. Every request through the Internet from People, Things, or Applications passes through each component starting from the top of the platforms stack. The entire architecture is composed of three main building blocks: application server, application, and persistent storage. Each building block has the responsibility to handle requests and forward the result of the treatment to the component below in the stack.

The application server is the first component in the software stack. It accepts HTTP requests, handles execution time and forwards requests to the application. We do not detail this component further as it can be any standard HTTP server that can forward requests to our application. The second component, the application, implements virtual entities of \(<People, Places, Things, Applications>\). Persistent storage maintains the relationships between virtual instances of\(<People, Places, Things, and Applications>\). It also stores the messages exchanged between these entities.

Application component is implemented using an open Web framework. From the top of the application stack, access to the platform is designed either for Things and Applications (OAuth 2.0) or for People (User authentication).
Figure 25: System Architecture of IoT service platform

The authentication by OAuth 2.0 [60] uses access tokens. Access tokens are random strings that are requested by People: for each Thing or Application registered by Place Masters, this component generates a unique string that can be afterwards used for authentication. This component is in charge of lightweight authentication for Things with constrained resources and for Applications accessing the platform. This authentication also allows the system to identify whether requests come from People or Things and Applications. In addition, with this component People do not need to save their credentials on Things or to provide them to third party Applications. The main role of this component is therefore to identify Things and Applications and forward requests to the REST Server component.

REST Server formats request and responses from and to the Things and Applications. It is in charge of reading request parameters and generating responses in the following formats: JSON, XML, and CSV. This component has
a role of middleware that allows for integration of several data exchange formats that are used by heterogeneous Things or Applications. Each incoming or outgoing data has to be treated by this component.

In parallel with Authentication OAuth 2.0 (Figure 25), access to the IoT platform is managed by User Authentication and User Interface modules (Figure 25). Access to the platform is exclusive: requests from Things and Applications are treated by OAuth 2.0 and REST Server and requests from People are managed by User Authentication and User Interface. User Authentication is an implementation of authentication by commonly used credentials (i.e. username and password).

Things and Applications API is a set of classes and libraries available for accessing the virtual identities and digital information of Things and Applications. It provides programming interface for interaction between virtual entities on the IoT service platform. This component implements most of the concepts and interactions from our general framework (chapter 4.6.1). It exposes virtual identities and interaction possibilities in form of a Web API. We can illustrate the concept with an example from our general framework where Thing 1 updates its state:

\[ \text{UPDATE (VT1,VP1, message)} \]

In this example, we first bring up the class Thing. The instances of this class have a unique identifier and the method update State (id, place id, message). The API allows invoking an instance of Thing and its methods by a call to a URI. In this example, the call would be as follows:

```
https://thingvibe.net/api/apikey/thing/updatestate/id/placeid/message/format/json
```

<table>
<thead>
<tr>
<th>1</th>
<th>thingvibe.net</th>
<th>URI of IoT service platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Api</td>
<td>platform component</td>
</tr>
<tr>
<td>3</td>
<td>Apikey</td>
<td>access token for authentication</td>
</tr>
<tr>
<td>4</td>
<td>Thing</td>
<td>class Thing</td>
</tr>
<tr>
<td>5</td>
<td>Updatestate</td>
<td>method of the class Thing</td>
</tr>
<tr>
<td>6</td>
<td>Id</td>
<td>unique identifier of Thing</td>
</tr>
<tr>
<td>7</td>
<td>Placeid</td>
<td>unique identifier of place</td>
</tr>
<tr>
<td>8</td>
<td>Message</td>
<td>digital content</td>
</tr>
<tr>
<td>9</td>
<td>Format</td>
<td>used by REST Server to format the data</td>
</tr>
</tbody>
</table>

Table 1: Elements of a request to IoT service platform
Table 1 shows how this request URI is composed.

The instance of class *Thing* (i.e. VT1) would accept request, invoke its method *update State* with received parameters and appeal to the next component of the platform stack: Access Control List (ACL) Manager.

*User Interface* is a Web based graphical interface allowing users to connect to IoT service platform, manage friends, things, applications, virtual places, and digital information. It is based on Twitter Bootstrap framework that can be easily adjusted with the screen size of user devices (e.g. desktop or smart phone screen). It also provides authentication interface to users. After a successful authentication, this component invokes the Access Control List (ACL) Manager.

*Access Control List Manager* (ACL Manager) is a set of classes and libraries that control the access rights of *People, Things, and Applications*. Its first function is to distinguish the roles of *Place Masters* from *Virtual Friends*. Any instance of *Virtual Thing, Virtual Place, and Application* contains information about its owner. The ACL Manager uses this information in order to grant the rights to execute communication and interaction verbs. Typically, *Virtual Friends* can only use READ on digital information in *Hybrid Virtual Places*, while the *Place Masters* can use UPDATE, and DELETE. The second function of ACL Manager is to grant access to particular *Virtual Places* only to *Things* and *Applications* registered by *Place Masters*. This function ensures that the digital information from *Things* and *Applications* do not escape the virtual context defined by *Place Masters*. If a request addressed to the platform is successfully validated by the ACL Manager, it can then be treated by the next component in the platform stack: Socialization Manager.

*Socialization Manager* is a set of classes and libraries helpful for searching and social network system on IoT service platform. Its first function is to allow *Place Masters* to search for *Virtual Friends, Virtual Things, and Applications*. The search can be based on several parameters: name, tag, identifier, etc. In addition, the search results can be refined by category and also by compatibility between *Things* and *Applications*. In fact, the IoT service platform allows developers to define the compatibility of *Things* with *Applications*. This property can be used in order to personalize searches for *Applications*. The second function of this component is to keep track of registered *Applications*. It is therefore a management system for an *Application* store. Its third function is to allow *Place Masters* to create *Virtual Places* and register *Virtual Friends, Things, and Applications*. Resulting networks of *<People, Places, Things, and Applications>* represent social graphs of virtual entities. Social graphs can afterwards be used to infer social relationships not only among *People* but also among *Things* and pairs of *<People, Things>*. Although the basic role of Socialization Manager is quite similar to the ACL Manager, this component is autonomous as it can be extended with recommendation functionalities. For
example, the social graph can be used by the Socialization Manager in order to personalize searches and suggestions according to recommendations made by friends. All digital information on IoT service platform is recorded and managed by the next component in the platform stack: Persistent Storage Server.

*Persistent Storage Server* is composed of two sub modules: Relational Database Management System (RDBMS), and NoSQL Database Management System (NoSQL DBMS). The responsibility of RDBMS is to keep track of user accounts. This approach is chosen for its ability to manage ACID transactions and consistently maintain identities of *People*. However, requirements for big data storage and scalability suggest using NoSQL for recording most other digital information. NoSQL DBMS is implemented in order to keep the social graph and messages from <People, Places, Things, Applications>. Appendix 1 shows the informational schema of the implemented persistent storage. One database (i.e. collection in MongoDB) has been implemented per component. The RDBMS has been used for user accounts. The NoSQL DBMS contains collections of <People, Places, Things, Applications>.

Software components implemented in IoT service platform allow us to demonstrate feasibility of implementation of IoT service platforms. Next, we evaluate how this software prototype supports the general framework by applying design science evaluation methods.

### 5.4 Prototype Evaluation and Results

Our general framework provides vocabulary and symbols that define interaction rules between *People, Places, Things, and Applications* mediated by IoT service platforms. Definitions, communication and interaction verbs from the framework were successfully implemented in our platform prototype. We named our IoT service platform *ThingVibe*, available on [92]. It is a functional platform for demonstration purpose. We built a presentation website with a blog in order to present the platform and eventually gather opinions from IoT community. The source code of platform prototype is also available as open source [93] under general public license allowing IoT community to use it as a reference implementation of our framework. The platform, as an instance of our general framework demonstrates the feasibility of our design process and of our designed artifact. This software platform is an innovation in the way that it defines ideas, practices, and technical architecture through which we can analyze, design, and effectively implement IoT service platforms. During the design process and the implementation, our aim was to identify still undeveloped capabilities of IoT service platforms needed to expand the IoT into a new realm of user empowerment. Consequently, the evaluation methods of our design artifact are chosen accordingly.
The design science research community has provided several evaluation guidelines [13, 94-97]. For example according to Hevner et al. [95] Information Technology (IT) artifacts can be evaluated in terms of functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes. They suggest a knowledge base of five evaluation methods that should be matched appropriately with the designed IT artifact:

1. Observational
2. Analytical
3. Experimental
4. Testing
5. Descriptive

Vaishnavi et al. suggest seven evaluation patterns [98] from which we chose demonstrative pattern for evaluation of our IoT service platform. This pattern states that the construction of a prototype demonstrates the solution to an identified problem. The aim of the evaluation is to demonstrate that our solution, IoT service platform, is reasonable for a set of predefined situations.

Our IoT service platform is at this stage a prototype and as such represents a proof of the concept. Therefore, evaluation methods intended to optimize systems or evaluate performances would not be appropriate at this stage. In addition, our IoT service platform is not an innovation in terms of technologies or communication protocols; it is an innovation in terms of design elements that uncover new properties of IoT service platforms. It is a technical and conceptual foundation which itself should be further developed and used. For example, Socialization Manager Component opens a field of study on online social networks and recommendation systems within the context of IoT. As the acceptance of IoT service platform depends on several user profiles and IoT stakeholders (firms, end users, developers, engineers, etc.) it is not possible, in short term, to conduct user tests and apply observational evaluation methods. In terms of user tests, this IoT service platform would encourage studies on how the power granted to one group of users influences other groups in the realm of IoT (i.e. empowerment). However, in order to collect data by conducting significant user tests, the evaluation would require a study of yet inexistent communities of IoT users and stakeholders.

For these reasons, we have chosen to use the following evaluation methods:

1. Experimental
2. Demonstration
3. Description
4. Analysis
5.4.1 Experimental
The first evaluation method of our IoT service platform is based on experimentations. Our initial experimentation focused on connecting Things with online cloud based data brokerage platforms (e.g. [42]). We performed a test with Arduino hardware using a light sensor. Arduino sensor was assembled from Arduino Uno board with an Ethernet Shield [99]. The hardware is a smart sensor assembled in Do It Your Self (DIYS) way. The board features an embedded Web server and the Ethernet connectivity to a home router. The hardware used the light sensor to measure the luminosity in office and sent this information to our IoT platform. The board received light amounts from the light sensor and used the Ethernet shield to transmit this information to a home router. The home router forwarded this information to our IoT platform. The experiment shows that it is relatively easy to integrate Things with our IoT platform. However, the embedded server does not feature secured connections (e.g. SSL) which would be required for product commercialization in order to protect user’s privacy. At this early stage of development of Arduino boards, this feature has not been implemented but we can expect it in the near future.

We also performed a software simulation. We built simulators for Things and Applications and tested connections to our platform through the Things and Applications API. The software that simulates Things and Applications could successfully use the RESTful API in order to interact with People, Virtual Places, and Virtual Things. Requests for digital information passed to the platform were successfully treated by each platform layer. The purpose of Virtual Places and Hybrid Virtual Places was also experimented. Digital information in Virtual Places defined by Place Masters was accessible and shared only by virtual entities registered by Place Masters. This simulation was performed using an exchange of text messages between <People, Places, Things and Applications>. Each message was logged on the platform and its compliance with the sharing properties defined by Place Masters was verified.

5.4.2 Demonstration
The IoT service platform prototype demonstrates that our theoretical framework is realizable and valid in predefined scenario. The prototype demonstrates that it is possible to leverage Things through Web platforms and online social networks. It also demonstrates that several Web technologies backed by database management systems can be used in order to develop a novel IoT service platform (i.e. design artifact).

If Web technologies are carefully selected, the integration of heterogeneous Things can be accomplished on IoT service platforms. Our platform does not feature predefined descriptions or properties for heterogeneous Things (e.g. only unique identifier and name are mandatory) that would allow computer systems to automatically bind applications with compatible Things. Instead, it allows
developers to use descriptions and properties of Virtual Things available on IoT service platform. It is a manual process for application developers without an automatic binding of Application with compatible Things. However, we recommend this method because it allows for more flexibility in two groups of stakeholders: hardware providers and application developers. Currently, there is no unique description for standard or semantic resources on the Internet and there is no reason to believe that the IoT will rely on a unique specification of Things properties. This approach attempts to promote agreements between hardware providers and application developers by providing the most used properties for Virtual Things. This is justified by an observation that the aim of both stakeholders is to maximize the number of users (e.g. customers). Therefore, being able to identify those most common properties is a big advantage intended to drive hardware providers to build Things with such commonly used properties in order to benefit from already available applications. Thus, Things with a larger number of compatible Applications will provide more services to end users. It should also naturally engage developers to target the most used properties of Things (based on Virtual Things) in order to expand the user base. Therefore, the aim of integrating heterogeneous Things with innovative services is a key objective.

We also demonstrate that it is possible to create an IoT application store on IoT service platforms. Application stores act as distribution channel for IoT services created by developers and software firms. An application store on IoT service platform empowers two groups of IoT stakeholders: end users and developers. End users are empowered to personalize their services by registering applications in their Virtual Places. Developers are empowered to maximize the number of application users by using a unified channel for distribution of applications. As a whole, this empowerment stimulates innovation.

Finally, we demonstrate that the control of IoT services can be accomplished through the Virtual Places created by end users. Place Masters can use applications registered in their Virtual Places in order to control an environment instead of interacting with each Virtual Thing. For example, an application registered in Virtual Place “Home” can be used in order to remotely put an entire Home in energy saving mode instead of each system. In addition, Virtual Places implemented on IoT service platforms empower end users to control the sharing of digital information between different online social contexts.

### 5.4.3 Description

The third evaluation method is based on description and logical reasoning. The related research in domain in Web of Things [41] demonstrates that it is possible to integrate the IoT with the Web. They describe a model of creating universal plug and play protocols [55] in order to connect the Things to Web platforms. This research demonstrates that the IoT technology is nowadays mature enough
to be integrated with the Web used by humans. Our platform is an extension of this research in the way that we built an online social network, *Virtual Places*, and Application Store as the core elements of user empowerment. Our platform features reasonable protection for sharing digital information through these social networks and *Virtual Places*. Although both, the social network and *Virtual Places* are not new concepts they were not yet studied under the light of IoT. In particular, sharing digital information through existing online social networks is often intentionally designed to expand the social network of users. This leads to unintentional over sharing of digital information based on a lack of privacy awareness [70]. Our concept of *Virtual Places* is implemented in IoT service platform in order to help People reduce over sharing of digital information in irrelevant online social contexts. The scenario defined in chapter 5.1 shows the usefulness of *Virtual Places*. It provides an example of sharing the digital information with two distinct online social contexts, “Home” and “Office”, by using *Virtual Places*.

5.4.4 Analysis
The fourth evaluation method is analytical [95]. The static structure of our IoT platform comprises five software layers: authentication, interface, access control list, socialization manager, and persistent storage. This architecture provides a clear definition of each layer and its responsibilities. It also allows modularity of software and further evolution of each layer. For example, user authentication module could be replaced by widely used online social network authentication systems [100, 101].

The OAuth 2.0 protocol had to be adapted for authentication layer of *Things* and *Applications*. In fact, this protocol is designed to allow users grant access to their digital information stored on one service platform while using third party web sites. This approach assumes that user interacts with an external web site and the web site requests an access to IoT services platform. In our case, users are already adding explicitly the *Applications* to their *Virtual Place* on IoT service platform. Consequently, we simplified the OAuth 2.0 protocol in order to achieve better user experience.

Regarding the socialization manager, it clearly opens new research directions. It is a mandatory component for IoT service platforms. This module should enable personalized and socially aware services to be built by exploiting the social graph between *<People, Places, Things>*. Our platform prototype exposes this social graph to *Applications* but it should be extended with publish – subscribe functionality. This functionality would allow external *Application* to subscribe to events and changes of state of real world and virtual entities (i.e. *People* and *Things*). For example Backplane protocol [102] could be used for publish – subscribe functionality if an *Application* is accessed by user’s browser. Publish – subscribe protocol would allow for near real time sharing of digital information
between the Applications registered in Virtual Places. However, when access to the platform is not performed by a Web browser another protocol with push functionalities could be chosen and implemented. On our platform, socialization manager is also used to incentivize application developers towards better Application – Thing compatibility. It features a search engine presenting the number of registered Things in anonymous way. The search results present the number of Things, category, description, and common properties. For example search result for smart TV can produce following results: 223 Things, smart TV, categories: entertainment, home appliances, properties: ID, Name, Serial Number, State on, State off, Consumption, Channel in use. This approach is rather suggesting developers to adopt a common API instead of imposing common description or methods to implement in their applications. This functionality provides a common ground for building an application marketplace. End user searching for applications could have personalized search results according to the compatibility of applications with the Virtual Things in his/her Virtual Places.

5.5 Summary

In our general framework and during the implementation of IoT service platform, we focused on the development of Virtual Places, Social Network, and Application registry. In this chapter, we designed an IoT service platform by defining a simple scenario, system requirements, and system architecture. According to these elements, we implemented our platform prototype. As a result, we demonstrated the feasibility of the implementation of our general framework. We applied evaluation methods from the field of design science in order to assess our IoT service platform. We showed that our IoT service platform is a valid implementation.

In the next chapter, we shall revisit our general framework for user empowerment and our IoT service platform.
Chapter 6. Revisiting the Framework and Generalization

In this chapter, we will generalize application scenarios where our IoT service platform can be used. We will first identify valid scenarios in which design properties of our IoT service platform are valid. This analysis will also allow us to identify the limitations of our design artifact.

Secondly, we will revise the elements of user empowerment in the IoT that are suggested in our framework and implemented in our IoT service platform. We will observe two groups of IoT communities: end users and developers. It allows us to improve the design of IoT service platform. Furthermore, we will compare the elements of our general framework and platform prototype with the initial aim to empower users in the IoT.

6.1 Design Invariants

Sports

Our IoT service platform model can be applied in several application scenarios like sports or leisure activities. In a sport activity like running, one can use a smart phone application to track position, speed, distance, and other relevant parameters. The person that uses the Application would create a Shared Hybrid Virtual Place on IoT platform and register the smart phone as a Thing. The smart phone application would upload this data to the Hybrid Virtual Place called for example “running place”. This person would be the Place Master and could add to this running place his/her online social relationships interested in running. The Place Master and his online social relationships (i.e. People) would share their running activities on IoT service platform. In addition, a software Application on IoT service platform could be created to analyze performances, competitions, time for activities, and suggestions for new running paths by analyzing statistics of People within the running place.

Product Management

The model can be applied in product management. For example in the smart product application scenario [17]. In this scenario, an employee from supermarket would be the Place Master. He/she would create a Virtual Place called for example virtual supermarket with public access. The Place Master would register products (Things) to the virtual supermarket. Customers could use an application on their smart phones to read the tags on products and obtain extended information about products from virtual supermarket. This information could include vendor specification, customer ratings, comments and other. In addition, software Applications on IoT service platform can be used in order to
analyze customer satisfaction or to compare prices with similar virtual supermarkets.

**Waste Management**

The model can be used for optimization of waste management in the cities. In the “Smart Urban Waste Management” application scenario [17] product packages are tagged with RFID technologies. At the waste disposal, each bin can be scanned with RFID readers and its content can be analyzed. This information can be stored on IoT service platforms in order to analyze and optimize the garbage collection through the city. In this application scenario the Things are tagged objects, the Hybrid Virtual Place contains several physical places and is created by person in charge of waste management i.e. Place Master. Software Applications can be developed by the local authority that would analyze the waste disposal and plan the route for its collection.

Design properties of Web based IoT service platform validate our general framework. However, platform development provides an insight regarding the fact that one IoT service platform cannot be unique service for connecting People, Things, and Applications. If we consider that in our instance of scenario “friendly TV watching”, the smart TV at home is connected to one IoT service platform and the smart TV at work to another (Figure 26) our model has to be extended.

![Figure 26: Scenario instance on distributed IoT platforms](image)

In this scenario, a software Application is registered on two IoT service platforms. It should perform separate request to each platform in order to create a service. Consequently, there should be a mechanism on each platform allowing Applications to perform requests at the platform level instead of Virtual
Places. That means that the platform should aggregate and provide information about all Virtual Places where an Application is registered and instances within Virtual Places. The above figure shows an Application that should present requests to two IoT service platforms in order to get the Virtual Places (i.e. virtual context) and receive the digital information that the Virtual Places contain. Each request should be presented separately and would be described as follows:

Precondition: Application is registered on IoT service platforms
Normal Flow: READ (Platform 1, VP_{1-n}), READ (Platform 2, VP_{1-n})

On Application request, each platform would treat request as follows:

Precondition: Application is registered by Place Masters in each Virtual Place VP_{1-n}
Normal Flow: READ (VP_{1-n}, P, VT)

Response from the platforms should contain collections of Virtual Places with collection of tuples <People, Virtual Things>. This request should also be treated by ACL manager module on IoT service platform in order to ensure that the Application gets the Virtual Places only where the Place Masters registered this Application.

In order to get relevant digital information, these requests can be extended by adding the date or location attributes:

Precondition: Application is registered by Place Masters in each Virtual Place VP_{1-n}
Normal Flow: READ (VP_{1-n, date, location}, P, VT)

The IoT service platform should return collections of Virtual Places with the collection of People and associated Virtual Things having the date and location specified in the request.

We implemented this extension of general framework in our platform prototype. The first tests confirmed that it is possible to extend the platforms with these features and that access to Virtual Places specified by Place Masters is respected.

By logical reasoning, we argue that our solution is also valid in other IoT scenarios where real time communication between Things and People is not of
major importance. Real time communication is required in ambient environments where Things are reacting autonomously at People’s presence. Typically, home automation systems would need a local communication and coordination system. In this scenario, Things connected locally over a short range of wireless communication use a local gateway in order to orchestrate interactions in a timely way. The reason for this architecture is lower power consumption and the need for real time reaction of Things to People’s actions. For example, a person at home, moving from one room to another while the system has to adjust to her/his presence. In this case (and in similar scenarios), using our Web based IoT service platform would not be adapted. Communication delay that can occur over the Web might result in activating the system in a room when the person has already passed through.

Another scenario can be theft control in groceries using RFID tags. In this scenario, at the exit of a grocery RFID readers are used to read the tags from products and to verify if these products have been paid. In the case of a product that is not registered as paid, a sound alarm notifies the possibility of theft. As several customers might be passing through the tag reader in very short time, the speed of alarm reaction is of major importance. Using Web based IoT service platform for these systems might result in delays that trigger the alarm when the next person is already passing through the reader. That would result in the control of the wrong person at the exit of the grocery. Hence, these errors would invalidate People’s trust in the system. From these two scenarios, we can conclude that a Web based IoT service platform is not adapted when real time actions are required.

In spite of the lack of real time reactivity, our platform can be used as an extension for local systems. Local environments can be connected to our platform in order to upload the interaction logs. IoT service platform can be used afterwards in order to perform analysis of interactions and aggregate results into user models [62]. IoT service platform can store and aggregate history of user’s actions into a “user model” that is afterwards used to predict user behavior and preferences. It would allow for better personalization of services that could be offered through the Internet. For example in shopping scenarios the purchase history can be stored and analyzed on our IoT service platform in order to request product offers from physically distinct locations. In this case, the socialization manager module on our platform should be extended in order to manage user models of People. The Applications registered on IoT service platform could afterwards use this model in order to personalize products’ offers.

The services on our platform are based on the third party, Applications. This model allows for a high extensibility of the model for various domains of applications. Conclusively, the model is valid in most of scenarios where real time communication between <People, Things, Applications> is not of major importance. More so, our IoT service platform model is valid and general enough
for IoT ecosystem. The general framework and the platform prototype narrowed our attention to design properties of IoT service platforms. We can now take an outlook and analyze the impact of this model on user empowerment in the IoT.

### 6.2 Elements of User Empowerment

Foremost, we defined two general user profiles: end users and developers. Both profiles are users of IoT infrastructure and services. End users are using services provided by IoT infrastructure and by application developers. Application developers are users of IoT infrastructure provided by hardware providers. Our general framework and platform prototype empowers both user profiles: end users and developers. We first analyzed how our framework empowers end users to personalize services and then how it empowers developers to create and distribute IoT services.

#### 6.2.1 End Users Empowerment

The current industrial vision of IoT that promotes vendor lock-in and vertical silos reduces the power given to end users to use new IoT services. Our model of distributed IoT service platforms empowers end users to benefit from new IoT services. The architecture of distributed applications allows for the integration of services provided by applications through several IoT service platforms (Figure 27).

![Figure 27: End user empowerment to choose services in IoT](image)

In this architecture, hardware providers can use industry-centered approaches and provide an IoT service platform so long as end users register external applications. IoT service platforms can also be provided by several hardware...
providers or even by third party service providers. In each case, end users are empowered to use services provided by distributed software applications. Consequently, the first element of end user empowerment that is provided by IoT service platform is personalization of services with Applications.

There are many issues of end user privacy and data protection in IoT that have been widely discussed by the research community [73, 103-105]. When surrounding objects can sense the environment user privacy is at risk. Reporting precise physical location of users, tracking Things(e.g. [106]) and identifying users are privacy issues that are still not solved. In the IoT ecosystem, the problem relies on a tradeoff between privacy and personalization of services. In order to personalize services, service providers need precise information about service users [8]. In IoT, it implies that personalized services will require access to personal information, physical location, Things, and usage history. In this work, we focused rather on identity management than on privacy. Although these two terms are tightly related, identity management does not imply completely isolating personal information in order to protect privacy. As identity management, we defined how People are interacting with services in IoT ecosystem and how they define what is known or not known to service providers. The concept of Virtual Places empowers People to manage their identities that can be used for personalization of services in IoT. Virtual Places are acting as containers for interaction between Place Masters (i.e. users) and Applications (i.e. services). These virtual artifacts allow users to define how to present their virtual identities available for Application. It does not mean that user profiles or personal information (e.g. name) are different from one Virtual Place to another, the personal information remains the same. It simply allows user to choose social relationships and Things to be shared with an Application as a service. In other words, the history of interaction between <People, Things> at each user defined Virtual Place, presents virtual identity and preferences that are available for service personalization. Consequently, the second element of user empowerment is the control over the share of digital information between Virtual Places and between different service providers.

Online social networks have the capacity to alter traditional power dynamics [107]. They empower users to influence buying decisions, mobilize for events or gather in communities around various political and social interests. By adding Things to online social networks, People will gain better insights about their physical and social environment. Digital information from physical Things are becoming available in online social contexts thus allowing People to gather in communities of interest around physical objects. At the community level, our model of IoT service platform empowers People to influence each other’s opinion about Things (e.g. clothes, energy consumption, etc.). At the individual level, our model of IoT platform empowers People to define how Things will be used within their online social network. Digital information about every
User Empowerment in the Internet of Things

connected *Thing* can be shared with online social relationships within *Virtual Places*. The socialization manager on our platform fulfills the task to maintain this social context. Consequently, the third element of user empowerment is the control of sharing digital information between different online social contexts.

Elements of end user empowerment provided by our approach are shown in Table 2. In this table, we summarized how our general framework and IoT service platform satisfy requirements for user empowerment in the IoT.

<table>
<thead>
<tr>
<th>Empowerment</th>
<th>General Framework</th>
<th>IoT Service Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Personalization of Online Social Context</td>
<td>Social Network</td>
<td>Socialization Manager</td>
</tr>
<tr>
<td>2 Control of Information Sharing</td>
<td>Virtual Places</td>
<td>Socialization Manager</td>
</tr>
<tr>
<td>3 Service Personalization</td>
<td>Application</td>
<td>REST API</td>
</tr>
<tr>
<td>4 Control of Virtual Environment</td>
<td>Virtual Places</td>
<td>Application</td>
</tr>
</tbody>
</table>

Table 2: Elements of general framework and IoT service platform that empower end users

6.2.2 Empowerment of Developers
The IoT service platform model empowers developers and third party software firms to provide services to end users in the form of software applications. Our IoT service platform provides RESTful API allowing application developers to program software applications using their favorite programming language and tools. It enables developers to use virtual abstraction of *Things* instead of developing hardware dependent software. Consequently, our framework and IoT platform prototype empower developers to build applications for heterogeneous IoT hardware including the *Things* that only have virtual identity (e.g. products tagged with RFID).

The socialization manager provides an application registry empowering developers to register and distribute their *Applications* through our IoT service platform. It is a unified channel for software distribution that allows developers to maximize the number of potential application users. Application registry is a foundation for an IoT application marketplace. It should be extended with features that allow monetization of *Applications*. Creating application
marketplace will allow developers and third party software firms to monetize their investments in innovative IoT applications and services.

With suggestions of “Application – Thing” compatibility IoT service platforms also empower communities of developers to merge towards common semantic of device descriptions and standards.

Elements of developer empowerment provided by our framework are shown on Table 3. In this table, we resumed how our general framework and IoT service platform satisfies the requirements for empowerment of developers in the IoT.

<table>
<thead>
<tr>
<th>Empowerment</th>
<th>General Framework</th>
<th>IoT Service Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Creation of Services</td>
<td>Virtual Things</td>
<td>REST API</td>
</tr>
<tr>
<td>2 Distribution of Services</td>
<td>Marketplace</td>
<td>Socialization Manager</td>
</tr>
<tr>
<td>3 Portability of Services</td>
<td>Heterogeneous Things</td>
<td>Socialization Manager</td>
</tr>
</tbody>
</table>

Table 3: Elements of general framework and IoT service platform that empower developers

6.3 Summary
In this chapter, we first identified design invariants of IoT service platform and scenarios under which it can be used.

We revisited the general framework for user empowerment in the IoT after implementation of IoT service platform.

Finally, we identified the elements of user empowerment in the IoT that can be initiated to adopt our framework.
Chapter 7. **Conclusions and Outlook**

This thesis takes a design science approach to address the issue of users’ empowerment in the Internet of Things. Rather than focusing on one particular aspect of user empowerment we considered the IoT ecosystem as a whole and how this empowerment can be initiated. Hence, the thesis provides a holistic view of an Internet of Things created “by the user and for the user” and it opens new research directions and challenges.

We started by identifying groups of IoT stakeholders and their role in the creation, distribution, and personalization of IoT services. Our observation of current research directions and practices shows two extreme approaches for creating IoT enabled services: an industry-centered “closed” IoT or an open “Do It Yourself” IoT. We observed that neither approach empowers users to effectively shape IoT ecosystem. This research suggests a common ground that empowers users to create and to benefit from new services enabled by IoT technologies [11]. Our approach takes into account the core elements that empower IoT users to develop and use services on top of IoT service platforms. As a result, it does not attempt to break vertical silos of industry-centered IoT. Instead, it is an extension that takes into account three IoT stakeholders: hardware providers, end users, and developers. Consequently, our approach for user empowerment in the IoT depends on the engagement of all identified stakeholders.

### 7.1 Contribution

Our general framework presents concepts, definitions, and use cases that are steps towards user empowerment in the IoT. This general framework is the main contribution of this thesis to the IoT research community. In particular, the framework:

- Relies on the *People, Places, Things*, and their virtual counterparts in IoT. The framework suggests working with these concepts as an abstraction in order to empower users to create, personalize, and distribute IoT services.

- Identifies that IoT service platforms are a common ground for user empowerment in the IoT. The *People, Places, and Things* concepts can be implemented on these platforms, thus allowing the adoption of an approach in between industry-centered and opening IoT. IoT service platforms lower barriers for user empowerment by making the *People, Places, and Things* concepts abstract.
• Suggests the design of IoT service platforms in open ways to include external developers and software firms. We defined software applications as artifacts that can provide meaningful services on top of IoT service platforms. We also defined the role of Applications as containers for interactions between People, Places, and Things.

• Defines how users should be empowered to personalize the execution of IoT applications by controlling their online social networks and Virtual Things through Virtual Places and Hybrid Virtual Places.

• Suggests the design of application marketplaces on IoT service platforms for the distribution of software applications. Application marketplaces empower developers to distribute services as software applications.

• Defines the core design elements for the creation of IoT service platforms favoring user empowerment.

The framework provides design elements to be implemented in IoT service platforms in order to empower communities of end users and developers in IoT. These elements were successfully implemented in our platform prototype [92]. The contributions of IoT service platform prototype are the following:

• The platform prototype validates our framework. Qualitative and descriptive analyses demonstrate the utility of the framework and the platform.

• The platform demonstrates that Web technologies can be used to create IoT service platforms. It leverages the Things on the Internet used by people. Thus, allowing unified services for People and Things to be created.

We also provided an open source IoT service platform for the developer community to extend the implementation of our framework.

To summarize, the contributions of this thesis are twofold: first a general framework for user empowerment and second a prototype of an IoT service platform. The framework has been tested and validated by implementing an IoT service platform. The model of Virtual Places and online social networks on IoT service platforms has been presented at Internet of Things 2010 [108]. The model of IoT service platform and Application Marketplaces has been presented
at Internet of Things 2012 [11]. The IoT service platform is in prototype stage, as it should be tested more exhaustively with communities of end users and developers. Most importantly, this thesis suggests a novel approach for user empowerment in the Internet of Things for the creation, personalization, and distribution of IoT services.

7.2 Future Work

This thesis illustrates how user empowerment can be initiated in the IoT. We demonstrated how design elements of IoT service platforms can be composed in order to support this empowerment. Empowering users leads to an IoT ecosystem that is more adapted to their needs. It is a basis for creation of innovative services enabled by IoT technologies. It provides a foundation for future work towards more intelligent Things and services and presents the challenges for their realization.

Bootstrapping Application Markets

User empowerment for distribution of IoT applications through marketplaces provides an opportunity to incentivize the creation of a plethora of new services. The challenge here is to attract a critical mass of users and to maintain the right balance between end users and developers. First, the Things will have to be designed as open platforms for new services instead of products for extending customer relationships. Currently, Things in the IoT are seen as tools that provide opportunity to an enterprise to maintain post-transaction relationships with their customers [109]. Connected Things can report their usage information to an enterprise that in turn will adjust its offers to their customers. In order to change this vision, the challenge is to suggest new business models that would integrate a vision that a Thing “has its own life” after it has been purchased. Things that are open for service creation could create demand from end users for new services and attract developers. Secondly, IoT service platforms would need to define new business models in order to maintain the right balance between end users and developers.

Deploying “Intelligent” Services

We suggested a framework to empower users to create, personalize, and distribute IoT services. IoT service platforms are enablers for this empowerment. Applications on IoT service platforms are at the stage of analysis of data or remote controlling of Things. For example, software applications for smart meters are designed to analyze power consumption of home appliances or remind users to remotely switch off home appliances if they forget. However, IoT service platforms open the research directions for “smarter” applications that can react autonomously according to user preferences. When a sufficient amount of usage data is acquired on IoT services platforms, Virtual Places, online social
networks, and usage history, this information can be exploited to create applications able to better assists users in their preferences and react autonomously.

Towards a Social Networks of Things

The vision of Internet of Things begins with the idea that Things and computer systems will help us to gain better insights into the physical world. Early stage of this vision copes with problems of disruptive technologies that change life. User empowerment is required in order to adapt technologies and services and make them less disruptive. The next stage of the Internet of Things services is yet to come with the empowerment of Things. Organized in social networks of Things, they should act together in order to improve our lives. This social network of Things can be initiated with IoT service platforms that offer a starting point to empower Things to become first class citizens of the Internet.

For example, a service whose objective is to further strengthen our proposition of empowerment would be the Facebook of Things. Things could also autonomously organize in social networks taking decisions and actions without human intervention (e.g. [110]). However, they would need to comply with our society and physical world.

Figure 28: Proposal of a “Thing Book”
Figure 28 expands this idea based on our framework. Example functionalities could include:

1. Social network of Things following the now prevalent approach to social networks of People (e.g., Facebook).
2. Self-organized “social activities” of Things. For example agendas for maintenance, propositions for renting of Things that are not used etc.
3. “Sensitive” objects that react to our interactions, have their own “emotions”, and react autonomously as peers (e.g. addictive products [110]).
References


29. A. Badii, D. Fuschi, and D. Thiemert. ELLIOT: Responsible open innovation and living lab based UI-REF enabled evaluation for wider adoption of IoT-enabled solutions. in Concurrent Enterprising (ICE), 2011 17th International Conference on. 2011.


39. T. Cohen. *Now even houses can "tweet"... as homeowner wires his cottage to Twitter*. 2009 [cited 2010 10/09]; Available from:

61. ETAG, RFID and Identity Management in Everyday Life. 2007.


75. L.R.a.F.M. Matthias Kranz. Things That Twitter: Social Networks and the Internet of Things. in Accelerators and Drivers of the Internet of Things. 2010.


Appendix

1. Informational Schema of Implemented Permanent Storage
2. Use Cases – Illustrative Example

Use cases for Place Master PM1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Create Virtual Place P1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Place Master PM1</td>
</tr>
<tr>
<td>Trigger:</td>
<td>Place Master PM1 create Virtual Place</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Physical Place P1 should exist</td>
</tr>
<tr>
<td></td>
<td>Physical location for P1 is specified</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>Virtual Place VP1 is recorded on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow:</td>
<td>REGISTER (PM1,name,description,location)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Register Application A1 in Virtual Place VP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Place Master PM1</td>
</tr>
<tr>
<td>Trigger:</td>
<td>Place Master PM1 registers Application A1</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Virtual Place VP1 exists</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>A1 is recorded within VP1 on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow:</td>
<td>REGISTER (PM1,VP1,A1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Register Thing T1 to Virtual Place VP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Place Master PM1</td>
</tr>
<tr>
<td>Trigger:</td>
<td>Place Master PM1 registers Thing T1</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Virtual Place VP1 exists</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>VT1 is recorded within VP1 on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow:</td>
<td>REGISTER (PM1,VP1,T1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Create Hybrid Virtual Place HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Place Master PM1</td>
</tr>
<tr>
<td>Trigger:</td>
<td>Place Master PM1 registers Thing T1</td>
</tr>
<tr>
<td>Preconditions:</td>
<td></td>
</tr>
<tr>
<td>Postconditions:</td>
<td>HVP1 is recorded on IoT service platform</td>
</tr>
</tbody>
</table>
Normal Flow: REGISTER (HVP, name, description)

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Register Virtual Thing VT1 to Hybrid Virtual Place HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>Place Master PM1</td>
</tr>
<tr>
<td>Trigger:</td>
<td>Place Master PM1 registers Virtual Thing VT1 in HVP1</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>VT1 is registered on IoT service platform</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>VT1 is recorded within HVP1 on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow:</td>
<td>REGISTER (HVP1, VT1)</td>
</tr>
</tbody>
</table>

Use Case Name: Register Application A3 in Hybrid Virtual Place HVP1

| Actors:       | Place Master PM1                                       |
| Trigger:      | Place Master PM1 registers Application A3              |
| Preconditions:| Hybrid Virtual Place HVP1 exists                       |
| Postconditions:| A3 is recorded within HVP1 on IoT service platform     |
| Normal Flow:  | REGISTER (HVP1, A3)                                    |

Use Case Name: Register Virtual Friend VF3 in Hybrid Virtual Place HVP1

| Actors:       | Place Master PM1                                       |
| Trigger:      | Place Master PM1 registers Virtual Friend VF3          |
| Preconditions:| PM1 is Place Master of HVP1                           |
|               | Hybrid Virtual Place HVP1 exists                       |
| Postconditions:| VF3 is recorded within HVP1 on IoT service platform    |
| Normal Flow:  | REGISTER (HVP1, VF3)                                   |

Use cases for Place Master PM2:

Use cases for Place Master PM2 are similar to use cases for Place Master PM1. Place Master PM2 creates Virtual Place VP2, registers Thing T2 and Application A2. In addition his Virtual Place VP2 becomes shared Virtual Place VP2 when a Virtual Friend VF1 is registered to this virtual place VP2. Therefore, shared Virtual Places are created by Place Masters and an additional use case for shared Virtual Place VP2 is following:
<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Add Virtual Friend VF1 into Virtual Place VP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Place Master PM2</td>
</tr>
<tr>
<td>Trigger</td>
<td>Place Master PM2 registers Virtual Friend VF1</td>
</tr>
<tr>
<td>Preconditions</td>
<td>PM2 is Place Master of VP2</td>
</tr>
<tr>
<td></td>
<td>VF1 exists</td>
</tr>
<tr>
<td></td>
<td>VP2 is created</td>
</tr>
<tr>
<td>Postconditions</td>
<td>VF1 is recorded within VP2 on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow</td>
<td>REGISTER (VF1,VP2)</td>
</tr>
</tbody>
</table>

**Use cases for Place Master PM3:**

For creating Virtual Place VP3 and its content, the use cases for Place Master PM3 are similar to use cases for Place Master PM1. Place Master PM3 creates Virtual Place VP3, registers Thing T3. This is a minimal use case for a Place Master. No Application or Virtual Friends are registered to VP3. However, PM3 can use Hybrid Virtual Place HVP1 as a Virtual Friend VF3 of PM1.

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Register Virtual Thing VT3 to the Hybrid Virtual Place HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Virtual Friend VF3</td>
</tr>
<tr>
<td>Trigger</td>
<td>Virtual Friend VF3 registers VT3 in HVP1</td>
</tr>
<tr>
<td>Preconditions</td>
<td>VT3 exists and VF3 is registered by PM1 in HVP1</td>
</tr>
<tr>
<td></td>
<td>Hybrid Virtual Place HVP1 exists</td>
</tr>
<tr>
<td>Postconditions</td>
<td>VT3 is recorded within HVP1 on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow</td>
<td>REGISTER (HVP1, VT3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Register Application A4 in hybrid Virtual Place HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Virtual Friend VF3</td>
</tr>
<tr>
<td>Trigger</td>
<td>VF3 registers Application A4 in HVP1</td>
</tr>
<tr>
<td>Preconditions</td>
<td>A4 exists and VF3 is registered by PM1 in HVP1</td>
</tr>
<tr>
<td></td>
<td>Hybrid Virtual Place HVP1 exists</td>
</tr>
<tr>
<td>Postconditions</td>
<td>A4 is recorded within HVP1 on IoT service platform</td>
</tr>
<tr>
<td>Normal Flow</td>
<td>REGISTER (HVP1, A4)</td>
</tr>
</tbody>
</table>

Until now, we defined creation and personalization of Virtual Places by Place Masters. Let’s observe interactions of our instances <PM1, P1, T1, VP1, VT1, A1> assuming that the VP1 is a collection of instances: <PM1,VT1,A1>.
Use cases for instances within VP1

Place Master PM1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Read digital information about Thing T1</td>
</tr>
</tbody>
</table>

**Actors:** Place Master PM1

**Trigger:** Place Master PM1 reads Virtual Thing VT1

**Preconditions:** Virtual Thing VT1 exists

**Postconditions:** Digital identity (description) of VT1 and associated digital information are presented to PM1

**Normal Flow:** READ (VT1)

**Notes and Issues:** Digital information from all Virtual Places and from all HVP associated with VT1 should be presented

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Update digital information of Virtual Thing VT1 in Virtual Place VP1</td>
</tr>
</tbody>
</table>

**Actors:** Place Master PM1

**Trigger:** Place Master PM1 updates VT1

**Preconditions:** PM1 is Place Master of VP1
  Virtual Place VP1 exists
  Virtual Thing VT1 exists

**Postconditions:** Digital identity (description) of VT1 or associated digital information are updated in VP1

**Normal Flow:** UPDATE (VP1,VT1)

**Notes and Issues:** If digital identity of VT1 is updated it should be propagated to all Virtual Places and Hybrid Virtual Places where VT1 is registered

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Deleting digital information of Virtual Thing VT1</td>
</tr>
</tbody>
</table>

**Actors:** Place Master PM1

**Trigger:** Place Master PM1 deletes digital information of VT1

**Preconditions:** PM1 is Place Master who registered VT1
  Virtual Place VP1 exists

**Postconditions:** Digital information “content” associated with VT1 is deleted within VP1 on IoT service platform

**Normal Flow:** DELETE (HVP1, VT1, content)

**Notes and Issues:** If this action is called without specific virtual place and content it could be used to delete VT1 and its all associated content from all Virtual Places: DELETE (VT1)
### Use Case: Unregistering the Virtual Thing VT1

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Unregistering the Virtual Thing VT1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM1 unregisters VT1</td>
</tr>
</tbody>
</table>
| **Preconditions:** | PM1 have registered VT1  
Virtual Place VP1 exists  
PM1 is Place Master of VP1 |
| **Postconditions:** | VT1 is removed from VP1 |
| **Normal Flow:** | UNREGISTER (VP1, VT1) |

### Use Case: Send digital information

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Send digital information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Place Master PM1</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Place Master PM1 sends digital information to Virtual Place VP1</td>
</tr>
</tbody>
</table>
| **Preconditions:** | PM1 is Place Master of VP1  
Virtual Place VP1 exists  
A1 and VT1 are registered in VP1 by PM1 |
| **Postconditions:** | Digital information log is recorded on IoT service platform |
| **Normal Flow:** | SEND (VPM1, VP1)  
NOTIFY (A1) |

### Thing T1

<table>
<thead>
<tr>
<th>Use Case Name:</th>
<th>Send digital information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong></td>
<td>Thing T1, Virtual Thing VT1, Applications A1, A3, A4</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>T1 sends digital information to IoT service platform</td>
</tr>
</tbody>
</table>
| **Preconditions:** | T1 is registered on IoT service platform  
VP1, HVP1, A1, A3, A4 exist  
A1 is registered in VP1  
A3 and A4 are registered in HVP1  
VT1 is registered in VP1 and in HVP1 |
| **Postconditions:** | Digital information log is recorded on IoT service platform |
| **Normal Flow:** | SEND(VT1,content)  
NOTIFY (A1, A3, A4, content) |
### Application A1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Description</th>
<th>Actors</th>
<th>Trigger</th>
<th>Preconditions</th>
<th>Postconditions</th>
<th>Normal Flow</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Read the digital information about the Virtual Thing VT1</td>
<td>Application A1, Virtual Thing VT1</td>
<td>A1 reads digital identity of VT1 or associated digital information</td>
<td>A1 is registered in VP1 by PM1 VT1 is registered in VP1 by PM1</td>
<td>Digital identity (description) of VT1 and associated digital information from VP1 are presented to A1</td>
<td>REGISTER (VP1, VT1)</td>
<td>“Hard link” between VT1 and T1</td>
</tr>
</tbody>
</table>

### Use Case Name: Notify T1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Description</th>
<th>Actors</th>
<th>Trigger</th>
<th>Preconditions</th>
<th>Postconditions</th>
<th>Normal Flow</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Notify VT1</td>
<td>Application A1, Virtual Thing VT1</td>
<td>A1 notifies VT1</td>
<td>A1 is registered in VP1 by PM1 VT1 is registered in VP1 by PM1</td>
<td>Digital information log is recorded on IoT service platform</td>
<td>A1: NOTIFY (VP1, VT1) VT1: SEND (T1, content)</td>
<td>“Hard link” between VT1 and T1</td>
</tr>
</tbody>
</table>

### Use Case Name: Read the digital information about the Place Master PM1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Description</th>
<th>Actors</th>
<th>Trigger</th>
<th>Preconditions</th>
<th>Postconditions</th>
<th>Normal Flow</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Read the digital information about the Place Master PM1</td>
<td>Application A1, Virtual Place VP1</td>
<td>A1 reads VPM1</td>
<td>A1 is registered in VP1 by PM1</td>
<td>Digital identity of PM1 and associated digital information in VP1 is presented to A1</td>
<td>READ (VP1, PM1)</td>
<td>“Hard link” between VT1 and T1</td>
</tr>
</tbody>
</table>

### Use Case Name: Notify PM1

| Use Case Name | Description | Actors | | | | | |
|---------------|-------------|--------| | | | | |
| Use Case Name | Notify PM1 | Application A1, Virtual Place Master VPM1 | | | | | |
The second scenario on the Figure 23 integrates the concept of online social networks and sharing of Virtual Place VP2. Assuming that the VP2 is a similar collection as VP1, we can reuse the use cases as in the case of <PM1, P1, T1, VP1, VT1, A1>. However, there are additional use cases for <PM2, P2, T2, VT2, A2, VF1> in shared Virtual Place VP2.

**Additional use cases for instances within P2 and SVP2**

**Application A2**

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Notify other instances within VP2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>Application A2, Virtual VF1</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>A2 notifies VF1</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>A2 is registered in VP2 by PM2</td>
</tr>
<tr>
<td></td>
<td>VF1 is registered in VP2 by PM2</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>Digital information log is recorded on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow</strong></td>
<td>A2: NOTIFY (VP2, VF1)</td>
</tr>
<tr>
<td></td>
<td>VF1: SEND (PM1, content)</td>
</tr>
<tr>
<td><strong>Assumptions</strong></td>
<td>&quot;Hard link&quot; between VF1 and PM1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Read Virtual Friend VF1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>Application A2, Virtual VF1</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>A2 reads VF1</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>A2 is registered in VP2 by PM2</td>
</tr>
<tr>
<td></td>
<td>VF1 is registered in VP2 by PM2</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>Digital identity of VF1 and associated digital information within VP2 are presented to A2</td>
</tr>
<tr>
<td><strong>Normal Flow</strong></td>
<td>READ (VP2, VF1)</td>
</tr>
</tbody>
</table>
### Virtual Friend VF1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Use Case Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read other instances within VP2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actors</th>
<th>Place: Virtual Friend VF1, Virtual Place Master VPM2, Virtual Thing VT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>VF1 reads VP2</td>
</tr>
</tbody>
</table>
| Preconditions |VF1 is registered in VP2 by PM2  
VT1 is registered in VP2 by PM2 |
| Postconditions | Digital identity of VPM2, VT2, and their associated digital information within VP2 are presented to VF1 |
| Normal Flow | READ (VP2, VPM2, VT2) |

### Uses cases for *Hybrid Virtual Place HVP1*

#### Place Master PM1

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Use Case Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Virtual Thing VT1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actors</th>
<th>Place: Place Master PM1, Virtual Thing VT1, Hybrid Virtual Place HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>PM1 registers VT1 in HVP1</td>
</tr>
</tbody>
</table>
| Preconditions | VT1 exists and it is registered in VP1  
HVP1 is registered by PM1 |
| Postconditions | VT1 is registered within HVP1 |
| Normal Flow | REGISTER (HVP1, VT1) |

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Use Case Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Application A3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actors</th>
<th>Place: Place Master PM3, Application A3, Hybrid Virtual Place HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>PM3 registers A3 in HVP1</td>
</tr>
<tr>
<td>Preconditions</td>
<td>PM1 is Place Master of HVP1</td>
</tr>
<tr>
<td>Postconditions</td>
<td>Application A3 is recorded within HVP1</td>
</tr>
<tr>
<td>Normal Flow</td>
<td>REGISTER (HVP1, A3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Use Case Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Virtual Friend VF3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actors</th>
<th>Place: Place Master PM3, Virtual Friend VF3, Hybrid Virtual Place HVP1</th>
</tr>
</thead>
</table>
**Virtual Friend VF3**

**Use Case Name:** Register Virtual Thing in Hybrid Virtual Place

<table>
<thead>
<tr>
<th>Actors</th>
<th>Virtual Friend VF3, Virtual Thing VT3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>VF3 registers Virtual Thing VT3 in HVP1</td>
</tr>
<tr>
<td>Preconditions</td>
<td>VF3 is registered in HVP1 by PM1</td>
</tr>
<tr>
<td>Postconditions</td>
<td>VT3 is registered in VP3 by PM3</td>
</tr>
<tr>
<td>Normal Flow</td>
<td>REGISTER (HVP1, VT3)</td>
</tr>
</tbody>
</table>

**Postconditions:** Virtual Thing VT3 is recorded within HVP1

**Normal Flow:** REGISTER (HVP1, VT3)

For reading digital information within *HVP1*, applications *A3* and *A4* have the same use case whether they are registered by *PM1* or *VF3*:

**Use Case Name:** Notify T1

<table>
<thead>
<tr>
<th>Actors</th>
<th>Application A3 or Application A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>A3 or A4 read HVP1</td>
</tr>
<tr>
<td>Preconditions</td>
<td>A3 is registered in HVP1 by PM1</td>
</tr>
<tr>
<td>Postconditions</td>
<td>A4 is registered in HVP1 by VF3</td>
</tr>
<tr>
<td>Normal Flow</td>
<td>READ (HVP1, VPM1, VF3, VT3, VT1)</td>
</tr>
</tbody>
</table>

However, *Applications* can send notifications within *HVP1* only to instances registered by *People* that register the *Application*. As the *Application* *A3* is registered by *PM1* notifications (e.g. remote controlling *Things*) can only be sent...
to Things registered by PM1. In the same way A4 can only send notifications to Things registered by VF3.

**Application A3**

<table>
<thead>
<tr>
<th>Use Case Name: Notify instances in HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong> Application A3, Virtual Place Master VPM1, Virtual Thing VT1</td>
</tr>
<tr>
<td><strong>Trigger:</strong> A3 notifies VPM1 and VT1</td>
</tr>
<tr>
<td><strong>Preconditions:</strong> A1 is registered in HVP1 by PM1, VT1 is registered in HVP1 by PM1</td>
</tr>
<tr>
<td><strong>Postconditions:</strong> Digital information log is recorded on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong> NOTIFY (HVP1,VPM1,VT1)</td>
</tr>
</tbody>
</table>

**Application A4**

<table>
<thead>
<tr>
<th>Use Case Name: Notify instances in HVP1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors:</strong> Application A4, Virtual Friend, Virtual Thing VT3</td>
</tr>
<tr>
<td><strong>Trigger:</strong> A4 notifies VF3 and VT3</td>
</tr>
<tr>
<td><strong>Preconditions:</strong> A4 is registered in HVP1 by VF3, VT3 is registered in HVP1 by VF3</td>
</tr>
<tr>
<td><strong>Postconditions:</strong> Digital information log is recorded on IoT service platform</td>
</tr>
<tr>
<td><strong>Normal Flow:</strong> NOTIFY (HVP1,VF3,VT3)</td>
</tr>
</tbody>
</table>