

Cloudy in guifi.net: Establishing and sustaining a community cloud as open commons

Roger Baig^a, Felix Freitag, Leandro Navarro^b

^a*Foundation for the Neutral, Free and Open Network - guifi.net. Gurb, Catalonia*
<http://fundacio.guifi.net>

^b*Distributed Systems Group, Departament d'Arquitectura de Computadors,
Universitat Politècnica de Catalunya, Barcelona, Spain,*
<http://dsg.ac.upc.edu>

Abstract

Commons are natural or human-made resources that are managed cooperatively. The guifi.net community network is a successful example of a digital infrastructure, a computer network, managed as an open commons. Inspired by the guifi.net case and its commons governance model, we claim that a computing cloud, another digital infrastructure, can also be managed as an open commons if the appropriate tools are put in place. In this paper, we explore the feasibility and sustainability of community clouds as open commons: open user-driven clouds formed by community-managed computing resources. We propose organising the infrastructure as a service (IaaS) and platform as a service (PaaS) cloud service layers as common-pool resources (CPR) for enabling a sustainable cloud service provision. On this basis, we have outlined a governance framework for community clouds, and we have developed Cloudy, a cloud software stack that comprises a set of tools and components to build and operate community cloud services. Cloudy is tailored to the needs of the guifi.net community network, but it can be adopted by other communities. We have validated the feasibility of community clouds in a deployment in guifi.net of some 60 devices running Cloudy for over two years. To gain insight into the capacity of end-user services to generate enough value and utility to sustain the whole cloud ecosystem, we have developed a file storage application and tested it with a group of 10 guifi.net users. The experimental results and the experience from the action research confirm the feasibility and potential sustainability of the community cloud as an open commons.

Key words: Community networks; community clouds; Cloudy software stack; self-organised governance systems; self-provisioning; common-pool resources; commons.

1. Introduction

Commons are natural or human-made resources that are managed cooperatively [1, 2]. Based on that, we define open commons as commons with no restriction regarding the number of participants. Community networks (CNs), IP-based networks that are built, owned, and operated by citizens in a participatory and open manner, are a successful example of digital infrastructure managed as open commons. Hundreds of CNs operate across the globe in rural and urban as well as rich and poor areas. In these communities, the participants, including volunteers, enterprises, and public or private organisations share not only networking hardware but also time, effort, and knowledge. In Europe, several CNs have been operating for more than

Email addresses: roger.baig@guifi.net (Roger Baig), {leandro, felix}@ac.upc.edu (Felix Freitag, Leandro Navarro)

Please cite this article as: R. Baig et al., Cloudy in guifi.net: Establishing and sustaining a community cloud as open commons, Future Generation Computer Systems (2018).

This work is licensed under a Creative Commons “Attribution-NonCommercial-NoDerivatives 4.0 International” license.



a decade and have several thousand nodes.¹ In India, community-owned and operated wireless networks serve over 4,200 people in 38 districts,² and in South Africa, local ISPs are being created over a similar infrastructure.³ The CNs have been identified as one of the models for contributing to connecting the next billion [3] people that are still unconnected.

The generic term community clouds (CCs) refers to cloud systems designed to address the needs of a community [4]. The CCs have the potential to solve unattended needs of CN members and improve the efficiency of some existing services (mostly latency, performance, cost, and availability of services such as web, video, and data content) through the benefits of the cloud paradigm. These come mainly from the flexibility and savings that elasticity, pooling, and on-demand self-service bring in terms of performance, cost, and availability from the use of multiple alternative nearby servers in the same community or access network. The technical knowledge of the participants and their openness to research and innovation are other positive factors in this regard. The CCs deployed locally can bring benefits to communities and can address unattended needs, such as latency-critical applications; critical local sensing and control services that cannot rely on non-existent, fragile, or expensive Internet connectivity; local storage services with customised access control policies for content; applications for emergency and disaster scenarios; and privacy and data-security sensitive uses, where remote services may not be trusted or they gather, expose, or exploit sensitive data.

The CCs describe a model that can be organised in diverse ways, such as a competitive free market, a firm, a hierarchy, or a cooperative model.⁴ We claim that CCs can be organised as open commons, similarly to how CNs organise [5, 6]. The CNs offer a suitable model for CCs as open commons to emerge, given 1) the open and collaborative nature of CNs and 2) the piggybacking of pre-existent IP network infrastructure. The local management of CCs creates more opportunities for access to the cloud infrastructure adapted to the local socio-economic conditions, in terms of both production and consumption of resources and services. Thus, they have the potential to involve further participants and leverage new spaces of relationships, bringing new opportunities to entrepreneurship and innovation for more people. In the context of CNs, the capability of CCs to generate local value translates into incentives to strengthen the existing networks and to bootstrap new ones, especially in developing countries.

In this paper, we explore *CCs as open commons*: open user-driven (self-provided) clouds formed by community-managed computing resources, where the infrastructure as a service (IaaS) and platform as a service (PaaS) cloud components are organised as common-pool resources (CPR), on top of which the users deploy software as a service (SaaS) to consume resources and platform services or provide application services for free or for a fee.

As we develop various aspects, we look at two key interrelated concepts and related research questions:

1. *Feasibility*, as the possibility of being easily or conveniently done, which is related reaching technology and demand readiness. This leads to the first question: *Which social and technological artefacts and what factors are required to bootstrap CCs organised as open commons?*
2. If feasible, then we look at *sustainability*, as the ability to be maintained at a certain rate or level, which is related to organisational, governance, and economic aspects. This leads to the second question: *Which artefacts and factors are important to sustaining CCs, and how do these affect the value of CCs for their participants?*

We claim that this kind of cloud can emerge through a suitable cloud software stack (the artefact we have identified as the cornerstone of the initial phase) and an enabling networking infrastructure in place (not necessarily a CN). They can grow and become sustainable when the appropriate governance tools are in place and when they can generate value for the community. This value comes from the usefulness

¹For example, Freifunk (<http://freifunk.net>) in Germany and guifi.net (<http://guifi.net>) in Spain both have over 30,000 nodes (2017).

²The Wireless for Communities (W4C) programme (<http://wforc.in/> by the Digital Empowerment Foundation <http://defindia.org/>)

³the Zenzeleni.net project (<http://zenzeleni.net/>).

⁴In fact, Oliver E. Williamson and Elinor Ostrom shared the Nobel Prize in 2009 for the analysis of economic governance, the first for the boundaries of the firm, and the second for the commons.

and economic utility of the specific services the CC provides to their participants. With these conditions, feasibility translates into sustainability.

To prove our claims and answer our research questions in detail we have:

- a) Analysed a success case of digital infrastructure managed as an open commons, inspired in the guifi.net experience,
- b) Proposed a framework for the structure of a CC,
- c) Designed a CC software stack and implemented it in a Linux-based environment,
- d) Co-designed and promoted our cloud software stack in a community and assessed its initial uptake to confirm feasibility of this kind of cloud,
- e) Made a techno-economic analysis of the viability of a cloud storage service in this environment as a proof of concept of the sustainability of CC services.

The research method combined participatory action research [7] with iterative experimentally driven research [8]. We analysed guifi.net, the most successful and developed CN in terms of network size, number and variety of participants, governance model, and complexity of the ecosystem. We have also worked closely with the guifi.net community and specifically with a subgroup of volunteers to elaborate together our proposals, conduct our experiments, and discuss our results and implications. After an initial analysis, research, co-design, and prototyping during part of 2013 and 2014, we worked with the guifi.net community in two main iterations of the research over a period of two more years starting in June 2015, integrated in a community of practice with around 60 participants. After the intensive iterative development phase, the Cloudy system is still maintained and extended with new features jointly developed by the community of users and the research group. This work is supported by the netCommons and RIFE research projects until the end of 2018. The interactions with the community took place in two mailing lists for discussion and mutual support and in focus groups in several weekly community meetings. The group discussions were around the planning, co-design, transformation, and result phases of the action research, with a focus on the cloud deployment and the use of the software system, services, and applications. We used a combination of larger meetings to collect diverse opinions, and small group meetings for analysis and decision making. These interactions were useful to better understand the needs and preferences, to obtain feedback about new features and changes as the software evolved, and to refine the community cloud model and its governance instruments.

The rest of this paper is organised as follows. In Section 2, we discuss the research related to the main contributions of this paper. In Section 3, we describe the commons model that guifi.net applies for the access, operation, and governance of the network infrastructure. In Section 4, we propose a framework for managing CCs as open commons. In Section 5, we discuss the design and implementation of our CC software stack, and in Section 6, we evaluate the feasibility of its deployment. Section 7 explores the economic sustainability of a production storage service as a case for professional activity in the CC. Section 8 discusses and further elaborates the results as they relate to the main research objectives. Finally, in Section 9, we conclude outlining our findings and future work.

2. Related Work in Community Clouds as Open Commons

From the review of the related work, we can conclude that CCs as open commons have not been studied and presented beyond our research. We discuss the most relevant literature of the main topics as follows.

In terms of computing paradigms, fog [9] and edge computing [10] have been proposed to enable a new type of cloud services at the network edge, combining an edge cloud infrastructure in addition to that in large data centres. Fog computing migrates cloud computing to the edge of the network, where, through edge networks, more decentralised services are expected to replace centralised cloud services [11]. Edge cloud computing is well suited to perform local data processing for the Internet of things (IoT) [12]. From the

peer-to-peer (P2P) computing perspective [13], the users can be in the loop and participate in the edge cloud computing service provision. Our approach shares the focus on decentralised infrastructures and services.

This paper explores the potentials of the application of the CC concept [14] to edge cloud services that are provisioned by communities of citizens. A CC, in general, is a cloud deployment model in which a cloud infrastructure is built and provisioned for exclusive use by a specific community of consumers with shared concerns and interests, owned and managed by the community, by a third party, or a combination thereof [15]. Different examples of CCs are the ExoGENI⁵ distributed networked infrastructure for experimentation and computational tasks, the EGI⁶ federation of 350 resource providers offering computing, storage, and data infrastructure, and ELIXIR⁷, a European intergovernmental organisation that brings together life science resources including databases, software, and computing tools for researchers.

Cloud software, service, and business models abound. There are proprietary remote public cloud solutions offered by the major players [16]; closed-source CC solutions; and full stack open-source products in the market like OpenStack or OpenNebula, compared in [17], intended for rack or data-centre class computing clusters. These solutions are being used in several application areas, such as in the financial, governmental, and health sectors, fulfilling community-specific requirements (e.g., security, performance, and local content) [18, 19]. Considering the CAP theorem [20], decentralisation means 1) local availability, as the network may be unreliable, 2) eventual consistency, as data may not propagate synchronously, and 3) partition tolerance, as geo-replicated systems in large networks tend to suffer from network partitions. Therefore, software tools come with assumptions about the centralisation of the underlying computing and networking infrastructure that may not apply in a community environment. We have developed a model for managing and aggregating computing resources in decentralised networks. However, we envisage that these ‘industrial’ stacks will interoperate and be federated to our software stack through public APIs as hybrid clouds [21] do, combining the capabilities of in-house (private) and external (public) clouds.

Regarding user-oriented applications, there is a wide range of free software content management systems (CMS) [22], file sharing [23, 24], among many other web tools that are useful in a CC. However each requires a specific operating system environment, installation and configuration, so a single server host may not fit all. The need for security and resource isolation, resulted in the idea of process containers [25], that years later translated into Linux containers, an operating-system-level virtualization method for running multiple isolated Linux systems (containers) on a control host. This has evolved in products like Docker [26] or cluster managers such as Kubernetes [27] that also ease application and service management and even orchestration among complementary services. In our systems, we started with containerised services. Later added support for Docker images and Docker Compose to automate the deployment of applications in separate containers, and benefit from the wide range of pre-packaged Docker containers available that can be included and run as applications in our CCs. We found limitations in performance isolation required to avoid resource interference across containers, to be able to ensure the quality of the services provided. We developed solutions [28, 29] but these are not yet integrated.

In terms of governance models, the commons is a traditional model to manage access to critical resources in a community (e.g., irrigation systems or fishing grounds). This model was studied in depth by E. Ostrom [30]. According to Frischmann [31], infrastructures governed as commons generate positive externalities (positive effects) that benefit society by creating opportunities and facilitating many other socio-economic activities, although their benefits are sometimes hard to measure. An infrastructure that is cooperatively managed and sustained leaves a greater margin for added value activities compared to commercial networking infrastructures developed competitively. Examples of infrastructure commons are the Internet, with issues about congestion and network neutrality [32] or Internet/spectrum commons [33]. A first study of CNs as commons can be found in [34]. This work builds upon that study, covering CCs and the value of commons in the different cloud service layers.

Feasibility, the possibility of being made, has several levels of realisation. From a technology readiness level (TRL) [35] perspective, it relates to the results from the experimentally driven research [8] in terms

⁵<http://www.exogeni.net/>

⁶<https://www.egi.eu/>

⁷<https://www.elixir-europe.org/>

of the evaluation of technical alternatives, assessment of performance and cost, and optimisation of diverse aspects of the software system. Therefore, in terms of TRL, we target assessment of integration and validation (TRL 5), field demonstration (TRL 6), prototyping (TRL 7), and even complete system qualification (TRL 8) in some respects if the system is adopted by a community of users. From a market pull perspective of demand-readiness levels [36], we seek mature understandings about local demands and governance issues to allow a matching point between technology push and demand pull. Our aim is to reach an ‘adapted answer to the expressed need’ (DRL 9). The results of this research can reduce start-up costs and help lower the risks for success of related initiatives. That connects with the concept of sustainability, the ability to maintain an initiative after feasibility is proven.

Sustainability for CCs refers to the deployment of infrastructure and services with long-term availability and production usage, governed and used by a community. To be sustainable, edge cloud services must be able to generate a positive socio-economic balance for their participants [37, 38]. Enabling economic activity through commercial providers within commons infrastructures, such as in *guifi.net*, has been identified as a key element for enabling its transformation into a consolidated sustainable infrastructure [34]. Commercial providers addressing such environments need to face different risks in providing service-level agreements (SLA) [39]. However, edge cloud services open new business opportunities to better serve citizens with tailored and local services. These services can be related to local data processing, which may attract commercial providers. Such processing of data streams by devices at the edge was suggested in [40], with privacy as an important reason for providing local services [41]. The application scenarios for such services are diverse. Some examples are image processing [42] or sensor data processing [43]. The quality of service needs of these streams will require solutions to cope, for instance, with busy streams or real-time constraints [44, 45].

3. Digital Infrastructures as Open Commons: The Case of *Guifi.net*

Networking and computing infrastructures and services are critical resource systems for social inclusion and participation. As such, they should be accessible to all members of a society. The commons is a traditional and recognised organisational and governance model for shared resources [30]. Open commons are expressly open for participation by any stakeholder who is willing to contribute to its sustainability in exchange for the benefits it can extract (networking, computing, storage, and services). The participation is not limited to accessing the resource system for consumption or contribution, but it is also open for participation in the management of the resource system and the definition of its governance rules. Moreover, the commons, open or not, are self-organised structures; therefore, their sustainability depends on and benefits from contributions from all participants.

Our cloud model is rooted in the commons approach and has been deeply inspired by the work done by the *guifi.net* community to port the commons practices into the network infrastructure field. *Guifi.net* is a success case in terms of feasibility and sustainability of a digital infrastructure. For that reason, in this section, we present the key contributions of the *guifi.net* from the commons integration perspective. Section 3.1 discusses the fundamentals of the project, identifies the derived risks, and outlines the approach to their solution. Section 3.2 introduces the five main stakeholder groups of the ecosystem, and their interactions. Section 3.3 presents the two main legal documents that establish the participation framework (the Networks Commons Licence and the collaboration agreements), and Section 3.4 presents the three main practices that harmonise the socio-economic interactions (the conflict resolution system, the sanctioning system, and the economic compensation system). Section 3.5 discusses the main communication and coordination tools. Finally, Section 3.6 compiles the initial key enablers for *guifi.net* success and expansion.

3.1. Network Infrastructure as an Open Common-Pool Resource

The *guifi.net* network infrastructure is considered a commons, and therefore managed as an open CPR,⁸ with the network infrastructure as the core resource. Treating the infrastructure as an open CPR has some immediate positive effects, such as the avoidance of redundant infrastructure because all participants operate

⁸ A CPR typically consists of a core resource that provides a limited quantity of extractable fringe units.

on the same infrastructure and the increase of efficiency of the infrastructure in terms of cost savings and ease of participation. The CPR grows by the contributions that the participants make to expand or improve the network. In exchange, the contributors get network connectivity [34]. Participants can benefit from pooling with lower individual investments since resources are shared. The knowledge on the network is open, and the network is neutral; no barriers artificially limit the extent of contributions, such as expansion, content, or service creations. For commercial services, guifi.net as a CPR translates into a reduced entry barrier.

Nonetheless, guifi.net, as any other CPR, is fragile. More precisely, since it is (intentionally) non-excludable and the fringe unit (the connectivity) is subtractable (rivalrous), it is prone to congestion and exhaustion. Thus, efficient and effective governance tools are needed to protect the core resource (the network infrastructure) from depletion and ensure the maximisation of the production. Figure 1 presents the architecture of the body of normative instruments developed by the guifi.net community. They are all meant to be law enforceable and thus are written in accordance with current legislation. The basic normative layer consists of the Network Commons Licence (NCL) and the by-laws of the governing bodies. The NCL is a mandatory licence that any individual must accept to participate in the project. It establishes the fundamental principles and defines the universal rights and duties of all participants. The governing bodies are responsible for ensuring the compliance with the normative instruments, among other tasks. They establish the specific participation conditions of those parties for which the NCL does not suffice (e.g., for-profit participants.). The regulations, the third normative layer, develop specific aspects of the two previous layers (e.g., the sanctioning system). Finally, the good practices layer covers a heterogeneous set of rules and conduct codes in which compliance is desirable but not obligatory in general.

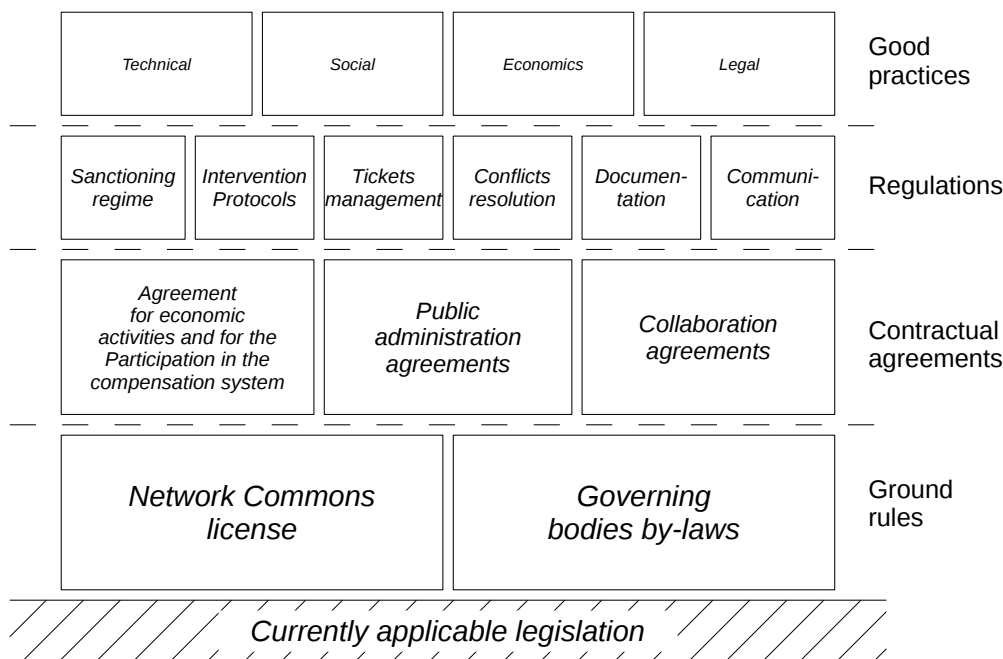


Figure 1: Body of normative instruments.

3.2. Stakeholders

Abstracting from the guifi.net community, we have identified five main stakeholder groups according to their roles in the ecosystem and their motivations for participating in it: the *volunteers*, the *governing bodies*, the *professionals*, the *customers*, and the *public administrations*, which form three sets, *non-profit*, *for-profit*, and *public interest*, as shown in Figure 2.

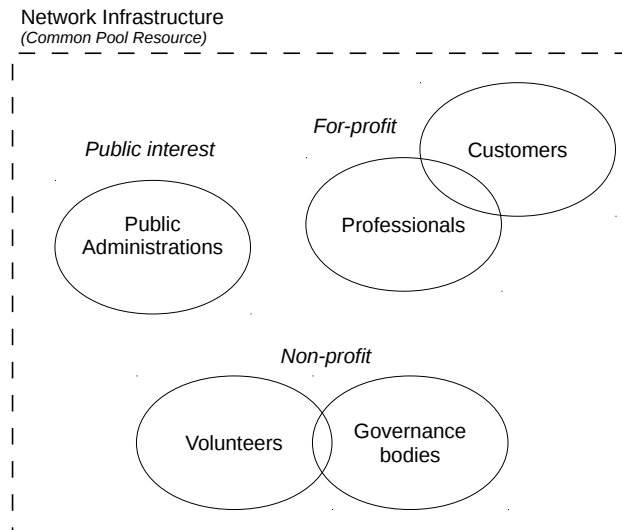


Figure 2: Identified stakeholders.

The *non-profit* set comprises the volunteers and governing bodies. The governing bodies deserve an explicit group because their mission is restricted to governance matters. In contrast, the types of activities of the volunteers are much more diverse. In *guifi.net*, there is a single governing body, *Guifi.net Foundation (Fundació Privada per a la Xarxa, Lliure i Neutral guifi.net)*.

The *for-profit* set is formed by professionals (e.g., ISPs, installers, and maintainers) who sell their services over the network and the customers who buy these services in exchange for money. Managing this commercial activity properly—one of the crucial issues of the governance model – has two direct positive effects on CPRs. On one hand, it brings income that makes the ecosystem economically sustainable; on the other hand, it promotes maintenance and upgrade of the infrastructure by these professionals because they depend on it.

Finally, the public administrations are responsible for regulating the interactions between the network deployment and *public interest*, such as public domain occupation. In the typical distinction of network providers or consumers, given the openness of *guifi.net*, all stakeholders can play both roles. This is reflected in the participation framework described next.

3.3. Participation Framework

The participation in the community is organised by agreements: the network licence shared by all participants and a set of bilateral collaboration agreements between the entity representing the community and other organisations, such as professionals or public administrations. Public written rules and their strict observation are effective to manage open communities.

Network Commons Licence The NCL⁹ is the licence that establishes the fundamental rules for participation. It is mandatory to subscribe to it to join the network. Its preamble sets the fundamental principles, and the articles precisely establish the participants' rights and duties to join and use the network. It allows offering services, either for a fee or for free. It is written to be enforceable under Spanish legislation, as legal certainty is essential to stimulate participation and investment, which in turn is the basis of any economic activity and therefore of the sustainability of the project. It sets the basis for any further licence or agreement. It has been modified over time mostly to adapt to new legal frameworks or better serve community needs.

⁹<http://guifi.net/en/FONNC>

Collaboration agreements These are agreements built on top of the NCL to further detail specific aspects, such as the rights and duties of the professionals or how a given public administration participates. The *agreement for economic activities and for the participation in the economic compensation system* plays a critical role in sustainability because it determines who must participate in the economic compensation system discussed in the following subsection.

3.4. Socio-economic Tools

These are tools aimed at keeping the infrastructure and community operational and balanced. A balance is needed not only between volunteers and professionals but also among professionals themselves since, although they compete for customers, they must coordinate and collaborate because they are sharing the infrastructure to reach their customers.

Conflict resolution system This system defines a precise and documented method to facilitate addressing conflicts in a quick and standard way. It consists of three gradual stages: conciliation, mediation, and arbitration. In the first, the disputing parties are formally required by the foundation to make all efforts to settle the case. In the second, a third neutral party, the mediator, is incorporated to help the conflicting parties reach an agreement. Finally, the last is driven by a lawyer, the arbiter, who at the end of the process renders a binding award. The award can only be appealed to the ordinary courts and can only be revoked due to procedural defects.

Sanctioning system A gradual system of penalties, ranging from a reprimand to limited or indefinite expulsion. The conflict resolution and sanctioning systems have a powerful deterrent effect.

Economic compensation system The economic compensation system [46] was developed and implemented to compensate for imbalance between investment in the commons infrastructure and its usage. It is mandatory for *those who make significant use of the resources* (professionals, investors, etc.). The expenditures declared by the participants are periodically cleared (balanced among participants with credit or debit entries or cash transfers) according to network usage. The foundation is responsible for the usage measurements, the validation of the expenses declared by the other parties, and the calculations. The results are made available to all participants in the compensation system.

3.5. Communication and Coordination Tools

The participants have developed several tools to facilitate communication and coordination among them and to manage the components of the infrastructure. The most relevant are:

Software tools for network management and provisioning The community of guifi.net has developed a set of software tools to ease the design, deployment, management, and operation of the network in a self-provisioning style. Most are integrated in the guifi.net website.

Communication tools Internal communication is structured around mailing lists with global, territorial, and thematic scope (open by default); social media (open by default with a few exceptions to protect sensitive information); and face-to-face meetings that play a key social role in community engagement and knowledge sharing. The external dissemination is based on face-to-face meetings, presentations, media, and social media.

3.6. Initial Key Enablers

What follows are, in our view, the key contributions that made guifi.net reach an unprecedented development in CNs.

Firm commitment to the fundamental principles The firm commitment to the network neutrality, freedom of use, and openness of participation and, consequently, the acceptance of the right of the private enterprises to carry out remunerated activities and the right of public administrations to become involved played a key role in the emergence of the current prosperous collaborative ecosystem.

Network licence The early adoption of a licence helped to rapidly establish common ground around rights and duties, to focus efforts, to reduce disputes, and avoid misappropriation of the CPRs. The inherent legal certainty made the prompt participation of public administrations and private companies possible, players that had never been involved in CNs until that point.

Clear definition of boundaries A direct consequence of the network neutrality is the clear definition of the scope of the project. The project is exclusively limited to the network layer; thus, it can be used to deliver any content.

Ease of participation The self-service approach implemented through a public website has been key to democratise the participation and incorporate ordinary citizens. An automated system of network resource management (IP address ranges and autonomous system numbers) and network device configurations reduces dependency on specific members, enhances the ecosystem resilience, and makes it scalable. It played a critical role in the growth of the network during the initial years of the project.

Professional activity At some point (around 2006), the network reached the critical mass to enable business opportunities. The emergence of companies delivering professional services enabled the unforeseen growth rate.

Reinvestment The governance tools, especially the economic compensation system, not only avoided the depletion of the CPRs but made the system healthier, which translated into an even higher growth rate.

Conception of the assets as a CPR The explicit identification of the network as a CPR was made few years ago (around 2014). The review of the literature on the commons showed that most of the best practices identified by the researchers were already in place. The remaining (essentially the sanctioning system) had already been identified as a need and was developed shortly after. Consolidated external references help to gain internal confidence and ease the communication with third parties.

Third-party technology availability The specific technology enablers for CNs were affordable WiFi technology and open-source operating systems for WiFi embedded devices.

4. Framework for Community Clouds as Open Commons

As with CN resources, cloud resources can be developed as open CPRs, and for that we have build on the analysis in the previous section. In the case of a CC, the core resource is nurtured by the diverse contributions of networking, computing, storage, and service elements that the participants deploy to expand or improve the CPR, and the fringe unit is the services they obtain. An architectural comparison of clouds to networks in the context of CPRs, detailed in [47],¹⁰ leads to the following main differences:

Diversity of building elements For a CC, the building elements are more diverse than in the underlying network. In addition to the physical level where host devices (servers) provide computing and storage capacity and the network provides connectivity, the cloud software stack components (IaaS, PaaS, and SaaS) are also building elements that provide diverse additional services (e.g., resource allocation and user authentication). This higher complexity has a direct effect on the CPR model, architecture, and implementation, as described in the following sections.

Inter-dependency among resources At the network level, the CPR is a primary infrastructure, that is, it has no inherent dependencies on other infrastructures. This is not the case for the cloud, which inherently depends on network connectivity for the interaction among building elements and users. In addition, there are inter-dependencies among physical resources with services and across different

¹⁰The comparison of models, conceptual and system architectures, and identification of requirements was part of Clomunity (EC FP7-317879).

services (e.g., a PaaS depending on IaaS). Moreover, some resources or services are more critical or demanded than others. Thus, the consequences of the deployment of an infrastructure with such dependencies must be studied not only from the viewpoint of usage/demand/traffic of a specific class of resources but also from a holistic approach. This will allow us to answer questions concerning the following: the complexity of interaction and balance across classes of cloud resources, resource bundles required in services, congestion management and fairness across services, and the influence of related infrastructures, such as the underlying network, the power grid, the environment, or the socio-economic aspects of the community of users and organisations. The answers to these questions have a direct effect on the design of the CC software architecture and particularly in the mechanisms for service allocation and discovery used to select service instances, described in Section 5.

Boundaries of the CPR The rule applied for the network infrastructure level, *the guifi.net community takes care of the infrastructure as a CPR, the content is left up to the users* (considering content to be pure usage and therefore external to the CPR), can also be applied to a cloud. However, the criteria to determine what must be considered external (as content) and what is considered infrastructure are flexible at the design level (see Figure 3 of Section 4.1) and might not be so obvious at the implementation level. In hybrid clouds, where in-house (private) infrastructure is complemented with external (public) infrastructure, that boundary is clear; external owned vs external rented and therefore metered. In our context, the main distinction depends on the nature of the exchanges that can be done cooperatively (to cover the costs declared) in a CPR, or competitively (at profit, with market prices ignoring the costs) in a commercial market, and even for free for some amount of exceeding resources. The different nature of the exchanges has a direct effect on the application of the CPR model to the governance of the different cloud layers, described in Section 4.1.

As for any other open CPR, the implementation and sustainability of a cloud requires effective rules and tools for the governance, maintenance, expansion, etc. In the following sections, we discuss our proposals to meet these challenges according to the same structure as in Section 3 and compare to the network infrastructure case, paying special attention to the differences just outlined.

4.1. Community Clouds as Open Common-Pool Resources

The fundamental principles of guifi.net also apply to a CC. It must be fully inclusive, that is, it must ensure the openness of access (usage) to the infrastructure, and the openness of participation (construction, operation, and governance) in the development of such an infrastructure and its community. The application of these fundamental principles results in a CC resource and service infrastructure that is a *socially produced collective good* governed as a CPR. Likewise, the high-level design requirements (e.g., standardisation of resource management, interoperability of individually contributed resources, and ease of participation) are equally valid.

With a set of essential IaaS and optional PaaS cloud services given as a CPR, enhanced and aggregated SaaS services can be built upon them and be offered on a cost-sharing or for-profit model. Similar to how the network CPR reduces the entry barrier (through transparency, network neutrality, and cost sharing, resulting in reduced CAPEX and OPEX costs) and enables the market niche of proximity services, a cloud infrastructure held as a CPR appears to contribute to making cloud computing even more accessible for entrepreneurs or small and medium enterprises (SME) as described next in general, and specifically in the comparison with commercial cloud providers in Chapter 7.4.

Scenarios for CCs

Figure 3 illustrates different scenarios for CCs. In CNs like guifi.net, although the network infrastructure is a CPR, the network can be provided according to different models by single or multiple commercial providers in a market [48]. Over the network layer, for the cloud service layers (i.e., IaaS, PaaS, and SaaS), different combinations of service provider solutions (SP) and community solutions (CS) are possible to satisfy the users' needs.

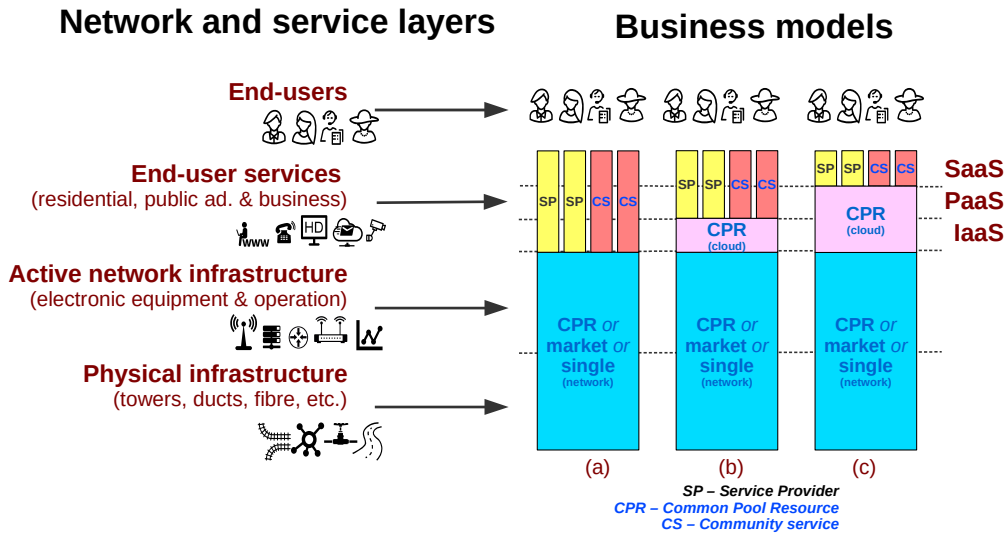


Figure 3: Alternatives in the layering of community cloud service provision over any kind of network service.

The scenario on the left of Figure 3 (a) corresponds to the *vertical integration* approach, the most widespread implementation, with no cloud CPR. In this scenario, a dedicated hardware deployment and software stack must be developed and maintained without the possibility of benefiting from any cooperative resource federation among different providers. Examples of that are the well-known commercial cloud providers, or private clouds, the combinations (hybrid) without horizontal integration at the IaaS or PaaS layers.

Scenario (b), the *infrastructure commons*, addresses a novel cloud scenario where the resources for the IaaS, such as virtualised or bare-metal computing, storage, or networking resources in community locations like cabinets, warehouses, or mini data centres, are provided by the participants and pooled cooperatively. One of the benefits for such a case is inherited from the characteristics of the cloud resources on which the IaaS is built. For instance, a geographically distributed heterogeneous IaaS enables new types of services that cannot be provided by centralised data centres. The proximity to its users translates into lower latency, lower network partition sensitivity, and higher trust, among others. Examples with various levels of cooperation are the ExoGENI or EGI, mentioned in the related work, and the Brighton Digital Exchange,¹¹ a cooperatively-owned and run data centre, where member businesses can peer and co-locate equipment and provide services to Brighton businesses and the world.

In scenario (c), the *platform commons*, commercial or community organisations provide end-user services through cooperative community-based IaaS and PaaS. In this case, the community-owned PaaS can provide valuable platform services of local interest, with tighter integration with the local environment. This setup, built on community-managed platforms can, for example, deliver activity logs that enable higher levels of transparency and auditability of cloud applications. A practical application is in the IoT domain, where a publish-subscribe platform service can be audited in terms of data input and output, thus providing more local control about personal data flows (such as running cloud applications in community or home locations, firewalled from the open Internet, and monitoring or auditing data transfers for trust and privacy reasons). An example of a platform commons is ELIXIR, mentioned in the related work. The experimental guifi.net CC is another case detailed in Section 5.

4.2. Stakeholders

Our experience over two years working with the guifi.net community and its community cloud deployment confirms a similar stakeholder structure of the guifi.net network commons. The coexistence of volunteers

¹¹BDX: <http://bdx.coop/>

and for-profit participants, already happening at the network level, has started to expand to all cloud layers, with initiatives, experiments, or offers at all cloud levels, but not to a significant scale yet. In a comparable way, the presence of a minimum number of customers (and the corresponding service providers) is crucial to ensure the required income to sustain the system. The participation of public administrations is also desirable but is not there yet. Nevertheless, the *proactive* motivations (*we participate because we want to, either to self-satisfy our cloud needs or to contribute to the initiative*) must be explicitly promoted because, at the cloud level, public administrations have much less legal obligations to intervene than at the network infrastructure level. Ideas such as the digital exchanges mentioned above, create opportunities and incentives for local partnerships among all stakeholders in creating local data centres to host computing and service infrastructure and to promote local public and private services, socio-economic development, and digital inclusion.

One topic in the discussion between the research team and members of the community involved in the pilot was about the inter-dependency between the different cloud components that can lead to a separation between volunteer and professional initiatives given the quality of service expectations, unless the software can provide effective quality control (e.g., isolation, prioritisation, interference, and congestion control), and a socio-economic mechanism exists to quantify and compensate the contribution and consumption of resources and services among all participants. However, this is still under research, regarding examining virtualisation [49], incentives [50], activity logs, and payments ledgers using blockchains or digital currencies [51].

During the pilot, the central role of software development was clear to all. The CCs require an international community of open-source software developers (voluntary or professional but not necessarily local) that develop and maintain a set of core tools, APIs, and compatible applications that can be shared by many CCs. The software, far from trivial, requires not only initial developments but also adaptation of cloud software intended for other scenarios (e.g., data centre), integration, and maintenance.

Finally, working with the community of users, the need for a reference authority was clear, similar to Guifi.net Foundation in the case of the network, as proposed by the guifi.net community. This role was assumed experimentally by the Guifi.net Foundation. This allows us to bypass the overheads of creating a new organisation, but in the future, if the project consolidates into a stable CC, a dedicated organisation is imperative to preserve the independence of the two ecosystems.¹²

4.3. Participation Framework

As a result of the pilot, and the discussion with the guifi.net community, we propose a separation of the body of normative structure into layers with a general mandatory licence and a set of complementary dedicated agreements. The details of the structure can vary for other CC, but the principles will be equivalent.

Licence A CC commons licence (CCCL), which harmonises the contribution and usage of the cloud resources, eases the take-up process of the CC model in a similar way as the network licence has had on the network infrastructure. The discussions about CCCL have started, but the licence has not been established yet. Similar to the NCL process, the steps to draft the CCCL licence go through deliberation with the community and evolve as the commons develops and transforms. The discussion with the community, with a rights-based approach [52], has resulted in the following proposals. The licence must consider aspects like the relationships between users and service providers and between the cloud layer and network layer. We propose that the licence must cover at least the following aspects:

Neutrality The requirement of public access without discrimination, that is, providing the same treatment (service) unless there is some compelling reason.

Fair use Rules of conduct and means of control to avoid abuse of the resources in commons.

¹²Guifi.net Foundation has actively participated in the whole conceptualisation process as well as the development of the software stack and in the implementation of the experiments.

Transparency and accountability As discussed, access to information and accountability is essential in any CPR and is an enabler for participation in the operation and governance of the commons.

Privacy In an architecture where sensitive data are distributed across the network, privacy respect and protection must start from the licence. This implies precautions to handle private data, which may be collected deliberately or not, with proportional care, according to data protection laws.

Collaboration agreements As with the network infrastructure, the level of commitment of the operators with the commons is expressed through a supplementary agreement detailing the specific implications of the licence, considering service-level objectives (SLO). The set of collaboration agreements for the cloud contributes to enhancing confidence among operators offering cloud services, comparable to our experience with the network infrastructure commons.

4.4. Socio-economic Tools

As in other communities, the governance involves all actors to drive a CC infrastructure through challenges and changes to keep it operational and balanced, which is key to resilient and adaptive CPRs. From the discussion with the guifi.net community, the following tools were proposed as necessary, although they must be adapted to local conditions:

Conflict resolution system In the case of guifi.net, the already existing system for the resolution of conflicts can be applied ‘as is’ to CC related issues.

Sanctioning system The general structure as well as the administrative provisions of the guifi.net sanctioning system is a good starting point. The technical provisions must be tailored to the specific requirements of CC.

Economic compensation system It must clarify the terms of participation to promote investment and reduce the number of disputes. The already existing network compensation system (see Section 3.4 and [46]) can be adapted to fit the cloud requirements and can be used to balance expenditure at the cloud layer. In addition, the effect that the usage of the cloud services can have on the network infrastructure and its consequences on the economic compensation system of the network must be investigated to determine whether the current calculation system, which is based on the total amount of network traffic at the points of presence of the infrastructure, suffices or needs to be adjusted. This system must balance the exchanges of disparate resources (IaaS, PaaS, and SaaS) between the different participants (volunteers, coops, and for-profit enterprises).

4.5. Communication and Coordination Tools

As with the network infrastructure, the diversity in the requirements for the coordination of the collaboration can only be covered through the combination of several tools. The discussion with the guifi.net community raised the need for the following tools:

Software tools for cloud management and provisioning The guifi.net website success shows that efficient and easy-to-use solutions for participant needs are key for the project uptake and harmonisation of the participation. As for the CC, most of the needs can be grouped per cloud layer as follows: for IaaS, contribution and request for computing and storage capacity; for PaaS, deployment and discovery of services, user authentication, and access policy management; and for SaaS, an initial set of appealing applications.

Communication tools From the experience during the pilot, tools (mailing lists, web fora, etc.) and strategies (face-to-face meetings, presentations, etc.) were indispensable. In an equivalent way, the provisioning and management of the infrastructure and services can benefit from specific solutions and separate communication channels.

4.6. Initial Key Enablers

Our experience shows that the start-up phase of any community initiative is critical, including our action and experimentally driven research. From the initial discussion with the group of volunteers to prepare the launch of the pilot and from the ongoing discussions and experience with the guifi.net community during the pilot, these are our collective findings on the key conditions (feasibility factors) that must be met to ensure a successful bootstrap of a CC:

Demand As a result of an increasing global awareness on data privacy, security concerns, etc., there is a growing demand worldwide for user-driven (self-provided) cloud services. We confirmed during the pilot that this demand also exists within that community. We collected requirements and suggestions from the participants, which determined the priorities in the implementation of the service offering.

Early adopters The start-up phase is critical for any cooperative project. Among the guifi.net members, there is a respectable number of technology enthusiasts interested in experimenting with innovations. Moreover, the authors have been involved in guifi.net for more than a decade, which facilitates the introduction of the product. The involvement of a few key active and recognised adopters in the start-up phase accelerated the involvement of the rest of the community in the pilot.

Third-party technology availability The two main requirements are network connectivity and computing and storage capacity. Affordable off-the-shelf devices powerful enough to run our cloud software stack have been available in the market for more than half a decade. Thus, a CC can be bootstrapped anywhere with local or Internet connectivity. In terms of software, the existence of many open-source base components (e.g., GNU/Linux, services, and applications) were enablers for the start-up phase.

Ease of participation As explained in Section 1, our working hypothesis is that CCs can emerge through a comprehensive cloud software stack that meets local needs. According to Section 2, such software solutions did not exist yet before our work. Although the early adopters in our pilot were not hindered by a more complex system, ease of participation resulting from the ease of use of the CC software was their main concern, and the usability of the user interface became a central aspect. Its design and implementation is discussed in Section 5, and Section 6 reports on its initial uptake within the guifi.net community.

Cloud licence In view of the positive effects of the guifi.net NCL (e.g., CPR protection, encouragement of investment, and dispute avoidance) and given that the introduction of major changes becomes more difficult in larger communities, the precise definition and approval of the *CCCL* is the main next step to ensure the sustainability of a larger community beyond the pilot.

5. Cloudy: A Community Cloud Software Stack

As part of our participatory action research, we have worked with the guifi.net community to understand the needs, and design, develop, test, and evaluate the solution over at least two cycles. Several iterations were made with many formal and informal meetings with groups of community members across all phases of the research, with key plenary meetings in two of the annual guifi.net conventions.

We envisioned the CC in the scenario (c) of Figure 3 (infrastructure and platform commons), realised through the combination of two main types of devices on top of the guifi.net network commons. The first consists of low-power consumption mini-computers contributed by the end users and installed in their homes and offices. The second are more powerful dedicated servers located in more centralised points, like local data centres, warehouses, or street cabinets, usually contributed as a result of a coordinated action of end users or by organisations, such as companies, universities, cooperatives, or public administrations.

Regarding software development and re-utilisation, we prioritised the open-software solutions for our own convenience (licences, access to the code source, and ethics) but also because this model is the best aligned with our target audience (e.g., activists, SMEs, and educational institutions). We have also prioritised the utilisation of existing solutions over our own developments, restricting the last to the minimum strictly

necessary. In accordance to this open spirit, we made all the code and related documentation publicly available.

In the remainder of this section, we describe how the cloud software stack presented in Section 4 has been implemented as an operational Linux distribution that we named Cloudy¹³ that runs a combination of IaaS, PaaS, and SaaS on end-user oriented cloud nodes (user devices), presented in Section 5.1, and includes the Community-Lab¹⁴ software that can run multiple service IaaS instances on server nodes (resource devices), presented in Section 5.2. Sections 5.3–5.5 discuss the specific aspects related to IaaS, PaaS, and SaaS, respectively.

5.1. Cloudy Software for User Devices

Cloudy¹⁵ is a Debian GNU/Linux-based software distribution aimed at participants of a CC that we have developed as part of our research.¹⁶ Conceived as an open-software platform to bootstrap and manage decentralised CCs and services in them, it includes a set of cloud services and applications that are managed through a common web front-end interface. As an open platform, it integrates a set of pre-installed applications, which the user can activate through the web interface, and allows users to install additional services (see Sections 5.5). As a cloud software stack, it offers IaaS and PaaS services (see Sections 5.3 and 5.4). These include common services needed by every participant to join and participate in the CC as well as most common network services used in guifi.net. In addition, users can extend Cloudy with their own services. This flexibility allows the service offerings (for free or for a fee) to grow and adapt to specific needs of the community. The source code as well as the binaries are freely available in public repositories.¹⁷

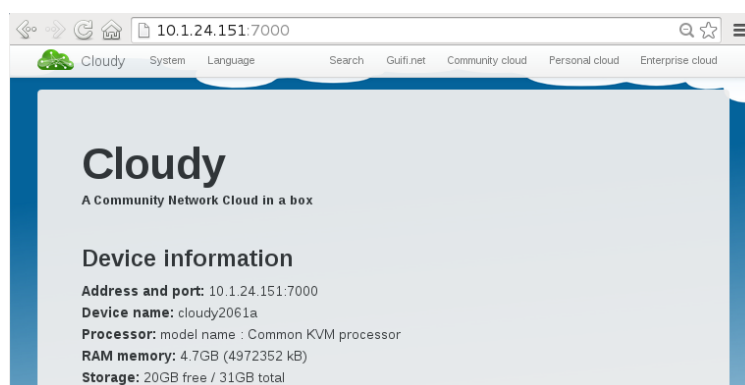


Figure 4: Cloudy web user interface with the main menu bar.

Figure 4 shows the Cloudy web interface right after its installation in a user's device. Services can be either private, only accessible at the owner's discretion, or public (i.e., discoverable and accessible by other participants in the CC through their Cloudy instances).¹⁸ The services are managed through five main tabs: *Search*, *guifi.net*, *Community cloud*, *Personal cloud*, and *Enterprise cloud*. The *Search* service is a service discovery that allows users to find Cloudy instances in the CC and to discover their public services. Through *guifi.net services*, the users can install the most common network services to participate in the guifi.net CN, some of them also common in other CNs. The *Community cloud* services currently consist of a set of distributed applications that can be deployed among the Cloudy nodes. *Personal cloud* services refer

¹³Cloudy was started as part of the Clommunity research project (EC FP7-317879).

¹⁴<https://community-lab.net/>

¹⁵Documentation for users: <http://cloudy.community/> and for developers: http://en.wiki.guifi.net/wiki/What_is_Cloudy/.

¹⁶In the Clommunity and netCommons (EC H2020-688768) projects.

¹⁷The source code is available as a GitHub project <https://github.com/Clommunity/> or <http://dev.cloudy.community>. The binaries are available as ISO images and as Linux containers <http://cloudy.community/download/>.

¹⁸A Cloudy instance is a piece of the Cloudy software stack running on a dedicated device or on a virtual machine.

to user-specific applications that may not necessarily be shared with the community. Finally, the *Enterprise cloud* refers to services that the user can install to enable third-party applications. All the services are either bundled with Cloudy or installable as Docker containers, obtained from an external container image repository (Docker Hub), as seen in Figure 5. The use of Linux containers brings the advantage of OS virtualization, with separate OS environments for each software component. Docker allows the use of public container images and their composition. Although we found this isolation acceptable in terms of security and privacy, we identified resource isolation and admission control as an area of improvement to ensure the quality of the services provided (see Section 2). We did research and proposed solutions [28, 29, 53] for performance isolation, including resource prioritisation and pinning, elastic resource provisioning, reactive tuning, and admission control, but any of these are included yet in our software.

Given the wide range of essential and optional software and services (IaaS, PaaS, and SaaS) for a decentralised CC, Cloudy can have a similar role in terms of standardisation and unification at the cloud infrastructure and service level as the guifi.net website has had at the network infrastructure level. Two mailing lists have been set up, one for the standard users¹⁹ and another for coordination of the development.²⁰

5.2. Community-Lab Software for Resource Devices

Resource devices (RD) are network-attached low-power computers (desktop, mini, or rack) deployed in several locations in the guifi.net CN. The RDs provide computing and storage resources in the form of virtual machines (VMs) implemented as Linux containers with access control, some degree of resource isolation, and management capabilities to grant a trusted remote user full access to the processing, storage, and network resources allocated to a given container. The software²¹ for RDs is based on the OpenWRT GNU/Linux distribution extended with a remote control (REST API) resource manager service that can manage the life-cycle of multiple containers running concurrently in the same host. We call each of these containers a *sliver*, and the set of slivers on diverse RDs belonging to the same service is called a *slice*.

A collection of RDs delegates its resources to a (single) resource controller service, as seen in Figure 5. The resource controller²² is a software package and a service that provides a web interface as well as a REST API to a resource registry. It allows users to create and manage slices in all associated RDs as well as register and offer new RDs, among other tasks. The resource controller allows service owners to select and request several computing resources from a collection of associated RDs. The RDs periodically poll the controller and launch the jobs requested, thereby downloading an operating system template and launching a container where a service can be installed: a sliver of a given slice.

5.3. Community Cloud Infrastructure

The infrastructure layer (IaaS, scenarios (b) and (c) in Figure 3) includes the infrastructural resources of the guifi.net CC. These are i) user-oriented cloud devices, acting as local interfaces or proxies between users and the cloud, so that participants can interact with these devices and the rest of the cloud through a web user interface, ii) dedicated computing and storage devices, devoted to running service instances and storing pieces of data, iii) the underlying network infrastructure, and iv) the locations where these devices are housed. All these resources can be sliced (virtualised) to be shared and provide isolation across multiple users (containers for computing and storage, VLAN or SDN for networking, and rack space in shared locations).

Cloud user devices Home devices running Cloudy as described in Section 5.1. The hardware used is either desktop PCs (e.g., Dell Optiplex 7010SF), mini PCs with Atom CPUs (e.g., Jetway, Minix NEO

¹⁹<https://l1listes.guifi.net/sympa/info/cloudy-users>.

²⁰<https://l1listes.guifi.net/sympa/info/cloudy-dev>.

²¹The software was developed in the CONFINE (EC FP7-288535) research project, as the Community-Lab.net testbed, with a set of RD devices in multiple locations of several CNs. Initially intended to run research experiments, it is integrated in the Cloudy software stack to run experimental services distributed across several RDs.

²²Also known as CONFINE controller or Community-Lab controller: <https://wiki.confine-project.eu/soft:server>

Z64, Jetway JBC372F36W-2600-B), or single-board computers (e.g., Beaglebone Black, Intel Galileo, or Raspberry Pi).²³

Cloud resource devices Pure server devices run the Community-Lab resource manager, as described in Section 5.2. The hardware corresponds to network-attached low-power computing devices (desktop, mini, or rack computers) deployed in diverse community locations. In the peak of the deployment, during the CONFINE project (2011-2015), there were 80 RDs. All these devices are managed through a single resource controller where a set of containers (a slice) can be deployed on several RDs in multiple locations.

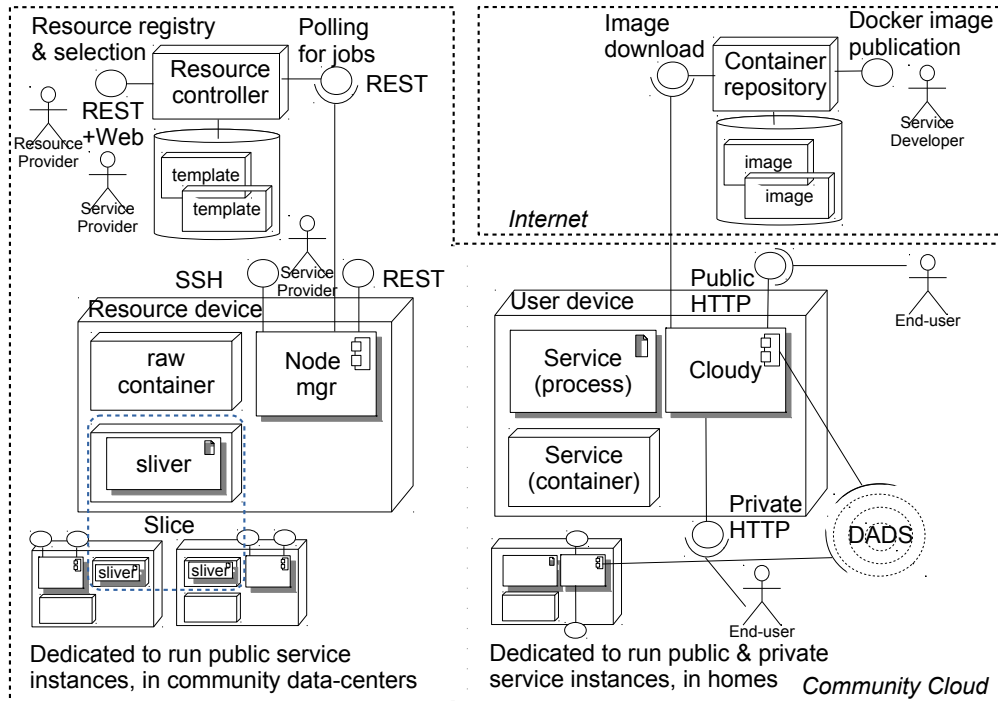


Figure 5: Structure and components of a community cloud deployment.

Underlying network infrastructure All server devices are connected to the guifi.net network, most of them in the Barcelona mesh network. The rest are spread across other rural or urban areas. We also have a few servers in the campus network and connected to other ISPs. A tinc overlay can be used to create a distributed overlay network with tunnelling and encryption across multiple ISPs, even bridging over NATs. Therefore, the model, software, and deployment can run on any IP network, including the open Internet.

Locations The devices are typically deployed at homes together and connected to the network router as well as in-service rooms, offices, or our research laboratory. Without restrictive environmental requirements, the devices just need power and network (wired or wireless), without any I/O connected device, as management is done remotely over the web or ssh interface. Most other guifi.net locations for routers qualify (e.g., homes, service rooms in buildings, street cabinets, warehouses, and public data centres). Only the resource controller server is a rack server hosted in the university data centre.

²³More details on the hardware installation procedures with performance and typical workloads in <http://cloudy.community/es/hardware/>.

5.4. Community Cloud Platform

The Cloudy platform layer (PaaS) offers services oriented to develop and support CC applications in a decentralised setting. The current platform includes three essential generic services, and three specific guifi.net services are also part of the platform. The generic services are:

Distributed announcement and discovery of services (DADS) This is responsible for automatically discovering the services available in the network and to present them in a meaningful manner. This is a critical service in a decentralised community context because the services are made available in an uncoordinated fashion. Moreover, DADS is based on Serf, a tool for cluster membership, failure detection, and orchestration that is decentralised, fault-tolerant, and highly available [54], which uses a gossip protocol based on SWIM.²⁴ The Cloudy web interface presents the discovered services grouped by category (see Section 5.1) and allows sorting the services according to several metrics including locality (based on latency) and availability.

User authentication service This service provides user authentication through a recognised independent third party that manages a registry of users. The concept results from the evolution of the solution to the authentication needs of the guifi.net federated proxy system. Currently, it is implemented using the LDAP protocol in a redundant master-slave architecture hosted and operated by Guifi.net Foundation.

Service activation and deployment Pre-configured applications or generic application containers can be started as Docker containers. These applications can be private or public (shared) in the CC (not in the open Internet).

The three main guifi.net specific services that have been integrated in the Cloudy platform layer are:

DNS service To participate in the guifi.net DNS system for the resolution of internal addresses (RFC1918). It is based on BIND.

Network monitoring To contribute data to the network monitoring system. It is implemented using SNMP feeding RRDtool ring buffers.

Web proxy To be part of the hundreds of Internet gateways contributed by volunteers that allow other guifi.net members to have free best-effort Internet access. Any validated user can access any of the federated web proxies. The service is based on the Squid proxy software.

5.5. Community Cloud Application Services

To identify the most demanded SaaS components, we did a survey of the application-oriented services offered in several European CNs, including guifi.net. Given that our work was primarily intended for this CN, we complemented our work with interviews with guifi.net participants in several face-to-face workshops. Our analysis [47] showed, on one hand, that remote file storage (backup, replication, and remote access) and video streaming of community events were almost common factors in all the cases, and on the other hand, that a solution to the diversity of other applications was mandatory to ensure the success of our development. The second finding motivated the *activation and deployment* service presented in Section 5.4. In response to our first, the following third-party applications have been integrated (pre-installed) in the current Cloudy software stack:

Syncthing A decentralised cloud storage system with cryptographic features for privacy, which gives full control to the users over where their data are replicated and shared with a group.

²⁴ SWIM: Scalable Weakly-consistent Infection-style Process Group Membership Protocol.

Tahoe-LAFS A fault-tolerant encrypted decentralised cloud storage system, which distributes user data across multiple servers in replicated data chunks. Even if some of the servers fail or are taken over, the entire file store continues to function correctly while preserving the users' privacy and security.

WebDAV server A set of extensions to the HTTP protocol, which allows users to collaboratively edit and manage files on remote web servers. Implemented with the Apache Web server DAV module.

PeerStreamer A peer-to-peer media streaming framework with a streaming engine for the efficient distribution of media streams, a source application for the creation of channels, and a player application to visualise the streams.

To ensure a satisfactory user experience, each application was carefully tested and some were tweaked (e.g., PeerStreamer) before becoming part of the pre-installed applications of the Cloudy software stack [55]. Assessments were done for Tahoe-LAFS [56], PeerStreamer [57], and the search service [58]. The evaluations included the deployment of the applications under realistic conditions on several nodes in the CN, considering the relevant metrics affecting the user experience in each case (e.g., read times in the case of Tahoe-LAFS, video chunk losses in the case of PeerStreamer, and service discovery effectiveness in the case of the search service).

6. Feasibility Analysis of Community Clouds

The feasibility of our concepts and developments were tested in guifi.net. For this purpose, we bootstrapped a CC inside this CN. The first Cloudy instances were deployed in March 2015 and are still active (September 2017). The main objectives were:

- a) Confirm the feasibility of the deployment (hardware and software) with a group of volunteers, and evaluate its performance and overhead in a realistic environment, starting from the first release of the software. Section 6.1 summarises the features of our experimental deployment, and Section 6.2 presents quantitative results.
- b) Work with a group of volunteers from guifi.net to co-design, fine-tune, and evaluate the benefits and limitations of the software system. Section 6.3 presents the main qualitative results from that experience.
- c) Discuss and develop the governance model with the participants and their experience as participants in the guifi.net CC. The results were the basis for Section 4.
- d) Discuss and assess the interest, feasibility, and sustainability of specific community services of interest. Section 7 describes our analysis and experience for the case of file storage.

The main findings of the feasibility analysis of the software system, which provide an answer to our first research question, are presented in Section 6.4.

6.1. Experimental Deployment

To test the software system under realistic conditions of number, distribution and heterogeneity of users, devices, locations, and connectivity, we prepared an experimental deployment in the guifi.net network. In June 2015, as part of our research in the Clomcommunity project, we made a call in the guifi.net mailing lists for volunteers to host MINIX NEO Z64 Android TV devices (150 € each) with Cloudy installed to be connected to active network nodes. The initial plans were to donate 25 units, the number funded by the project. Nevertheless, given the positive response, we finally donated 28 in the first call, and we made an additional call in July in which we donated another 16. In the second call, half of the costs were sponsored by Guifi.net Foundation and the other half by the end users. Beyond the sponsored devices, we know that, until the end of the year, around 15 additional devices were contributed by community members, making a total of at least 59 user devices involved. In addition, during that period, we maintained around five cloud

community instances of our Community-Lab cluster online. There is a public Cloudy instance available online²⁵ for demonstration purposes.

6.2. Quantitative Evaluation

With the aim to assess the ability of the Cloudy-based CC to operate in the experimental deployment, we performed experiments to qualify its operation. We evaluated the performance and availability of several services and applications, reported in [59]. However, the performance results reflect the performance and variability of the underlying guifi.net network across Cloudy devices and the network influence on each specific application, but does not help to judge Cloudy itself. The most representative metric we found about the value of Cloudy comes from the number of nodes and service instances reachable from a typical Cloudy node. As the discovery service presents to users the list of discovered service instances for a given service, in order of growing latency, the number of accessible nodes and services represents the level of choice that a Cloudy instance offers to its users. Beyond that, the choice and usage belongs to each CC service instance, and the resulting performance will depend on the specific devices and networks used.

Figure 6 shows the number of Serf network nodes and the accessible services seen by one of the Cloudy nodes in a typical week during the pilot study (end of November 2015). The figure is representative of what was observed over the duration of the pilot. The variations over time are due to the changes in the network conditions (load and network partitions) and the number of cloud nodes and services made available by the participants. The ‘Serf’ (top) line corresponds to the number of nodes of the Serf network seen by that Serf node. It is interesting to note that, despite all nodes having DADS activated by default, the ‘DADS’ line is usually below the ‘Serf’ line. This is due either to a manual deactivation of the DADS, which is very unlikely because there is no objective reason for that, or probably due to a timeout in the discovery of the DADS service. This parameter, set to 5 seconds by default, determines the maximum time allowed for responses to the search service queries. Thus, the most plausible explanation for the differences is that some of the nodes of the Serf network are too far away in terms of latency to be able to respond to a search query on time. The rest of services presented in Figure 6 are self-explanatory, except for the ‘OpenVZ’, the initial solution for computing virtualisation, which was later replaced by Docker. Current values can be seen at any time at publicly available Cloudy instances.

We know that at least two of the sponsored devices were not properly installed. Thereby, the rate between the total number of Cloudy instances deployed during the second half of 2015 that we are aware of (62), and the average number of online cloud nodes at the end of that year (50) was around 80%.

6.3. Qualitative Evaluation

Our main source to get qualitative feedback was informal interaction with the community, which we already started during the design process, and it involved at least 64 members. These interactions took place in two plenary meetings with a year of difference, several smaller meetings in different locations (guifi-labs), short and small weekly follow-up meetings between a few members of the community and the research team, and the continuous interaction in two mailing lists. According the discussions with the users, Cloudy was considered a useful instrument that effectively addressed some of the practitioners’ needs and has a lot of potential. The most frequent motivations for adopting Cloudy are the easiness to install and activate services, particularly the guifi.net specific services that are otherwise difficult to set up (see Section 5.4), and the flexibility to decouple services from servers. The users have acknowledged the ease of installation of the distribution as a key aspect for a successful uptake, and the search service as a satisfactory alternative to the current publishing services system (a static web page). The unification of management tasks through a meaningful website has also been well received. The main drawbacks were the need to go through an installation process, the limited set of applications, and the difficulty to expand the system with new features or customise the software.

²⁵<http://demo.cloudy.community> User: *guest*, Password: *guest*

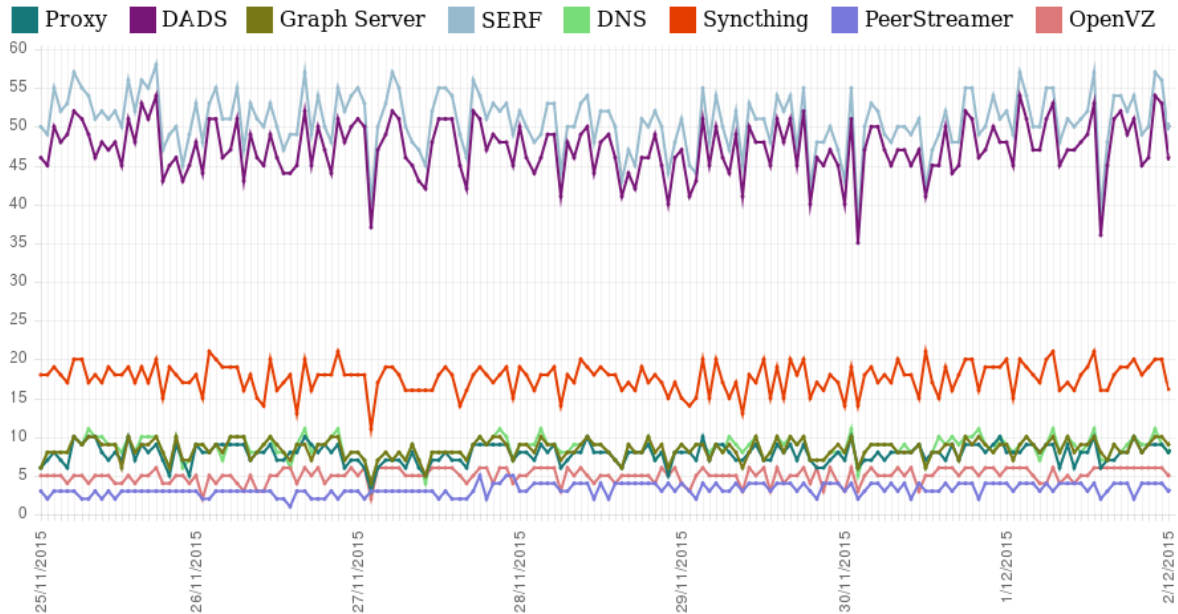


Figure 6: Number of Cloudy instances and services in the community cloud during a typical week (sampling period: 1 hour).

6.4. Findings and Implications

The positive qualitative feedback matches the quantitative results. The high rate of online nodes (around 50 on average) and the estimated successful adoption rate (around 80%) are well aligned with the positive comments received. This is also supported by the number of nodes contributed by the community members during the first half year: 15 fully paid by them plus another 16 partially paid (50% contribution).

The experience with the design of a CC, implemented through the Cloudy software stack as a key element, combined with the co-design and adoption of the CC governance model and the quantitative and qualitative evaluation of adoption by the guifi.net community validates the feasibility of CC as open commons. Feasibility is not only achieved at the level of ‘system prototype demonstration in operational environment’ (TRL 7) but, as being adopted by the guifi.net community, it can be considered ‘actual system proven in operational environment’ (TRL 9). The identification of key social and technological artefacts and factors that result in showing feasibility at certain levels is the answer to our first research question, further discussed in Section 8.

The difficulties found are mainly intrinsic to any experimental software system with a small developer community and were helpful to drive the development towards including more applications, particularly an extension that allows running third-party services, such as Docker containers, to increase the range of applications available. The fact that the system has a web interface was considered another limitation. In fact, while the majority found the system very useful to offer specific guifi.net services, a few users expressed decreasing interest given the software was less polished than other free commercial offerings. Regardless, these negative results were expected from such a pilot with limited resources, and this was not our ambition in the research.

The range and quality of the applications have an effect on the interest and value of a CC. The next section provides a detailed analysis of the technical feasibility and the economic sustainability of an application of wide interest among the participants in the pilot.

7. Sustainability Analysis of Community Clouds Through SaaS: The Case of File Storage

The analysis of long-enduring CPR depends on two key factors: 1) the design of a comprehensive governance system, which implies the existence of convenient tools to implement it [30] and adapt to change [60]

and 2) the overall goal to ensure the preservation and maintenance. As far as CCs is concerned, we have already discussed the first in Sections 4-6, and in this one, we explore the second.

A system can reach economic sustainability after its total cost is below or equal to its total revenue. In our scenario, the total revenue is the sum of donations, in kind contributions, direct contributions (active participation), and sales. As discussed in Section 3.6, the emergence of professional activity, which is mostly based on sales, permits overcoming the limitations of the dependency on amateur contributions and increases the growth rate. Therefore, it is essential to elucidate whether such professional activity can emerge on CCs, that is, whether CCs can offer business opportunities. We focused our research efforts on the SaaS layer because we believe that there are few chances to find business opportunities either in the IaaS or PaaS, at least in the initial stages. We were interested in a service to meet the break-even point (the point at which cost or expenses and revenue are equal) as quickly as possible. Thus, we were interested in a service with a good balance between the potential to attract users and the costs to put it in place and operate it.

We eventually analysed a low-cost file backup and synchronisation (BFS) service, a service we developed over Syncthing, which was already part of the Cloudy distribution. The rest of the section is structured as follows. First, in Section 7.1, we explore the software design, and then, in Section 7.2, we review the technical implementation assessment of a pilot deployment with real end users. In Section 7.3, we analyse the economic viability and sustainability of the file storage service, which is compared to the existing cloud commercial offers in Section 7.4. Finally, we present our findings and their implications in Section 7.5.

7.1. Application Design

The BFS service we developed is called SwebFS. SwebFS is essentially a web interface for Syncthing,²⁶ an open-source software that enables file synchronisation between devices that already was part of Cloudy (see Section 5.5). Syncthing provides users with control over where the data are stored and how often they are updated. It addresses several features relevant to end users: communication is encrypted, nodes are authenticated, the user can choose where the data are stored, and data can be shared with a group if desired. The Backup Syncthing (BST)²⁷ is a web application that manages the provision of remote, private Syncthing instances for end users through Docker containers. SwebFS, BST, and Syncthing are part of Cloudy, and combined they implement the BFS service.

There are other BFS-like cloud services offered by large commercial providers,²⁸ like Dropbox, Google Drive, SpiderOak, Box, SugarSync. Most of them offer a *freemium* service, with a limited amount of free storage space available to engage customers into one of the paid subscription plans offering more space. These services are based on closed-source proprietary solutions (e.g., the Dropbox client) with data lock-in in the hands of a foreign private company. Privacy handling of these services is a concern if the data of the users are hosted in countries with different data protection laws or the users do not trust the provider.

As an alternative, a few open-source software solutions,²⁹ provide file synchronisation between devices. To our knowledge, Syncthing is the only open-source application that provides easy multi-device file synchronisation without the need for a central server.

7.2. Technical Implementation Assessment

Even though typical storage services are intended to be decentralised, we prepared a simple single-server experiment to evaluate the technical feasibility of the service. This is an experimental SwebFS server for the BFS service. The hardware of this server exceeds the needs of the service (8 CPU cores, 8 Gb RAM, and 1 TB HDD), but the experimental results are not significantly affected by the hardware choice. Comparable results were obtained with smaller servers of equivalent CPU architecture in preliminary experiments. The

²⁶Syncthing: <http://www.syncthing.net>.

²⁷BST: <https://github.com/Clommunity/bst-mux>.

²⁸Dropbox: <http://www.dropbox.com>, Google Drive: <http://drive.google.com>, SpiderOak: <http://spideroak.com>, Box: <http://www.box.com>, and SugarSync: <http://www.sugarsync.com>

²⁹rsync: <https://rsync.samba.org/>, OwnCloud: <http://owncloud.org/>, Seafile: <http://seafile.com>, SparkleShare: <http://sparkleshare.org/>, Duplicati: <https://www.duplicati.com/>

server is connected at 100 Mbps to the guifi.net backbone. We had 10 users involved as participants in the experiment, running a Syncthing instance each, and 10 instances on the server.

The experiment consisted of monitoring the resources used to provide the BFS service to these 10 participating users during a period of two weeks. To do so, the users were required to sign up to the SwebFS server, install Syncthing on at least one of their devices (desktop computer, laptop, mobile device, etc.), link them, and share at least one file repository between them. The users were asked to copy at least 1 GB of data of any kind to the shared repository across their devices and the SwebFS server during the first week and to use this service to synchronise and backup their files as they would with any other commercial service they might be using. No specific protection measures were taken for this experiment to ensure privacy and security. Thus, users were recommended not to use files with personal information.

The following metrics (per user) were measured as a reference to estimate values in larger scenarios: *CPU usage*, *RAM consumption* and *network traffic* with a group of 10 users (1 server and 10 client devices). The results were very encouraging: only 0.1 (10%) CPU usage during active periods and negligible daily CPU usage averages, 27.6 MB RAM usage per user during active periods, and 12.5 MB on idle periods, less than 0.25 kbps data traffic on idle periods, and negligible control data traffic overhead during synchronisation periods, given the amount of data exchanged and the number of nodes involved. We use these numbers in the following economic analysis assuming resource usage scales linearly with the number of users or files, due to infrequent activity periods and file changes, and independent users. A super-linear growth in resource usage would require a redesign of the file storage service, with a lower ratio of users per server, or a redesign of the algorithms to have a sub-linear growth.

7.3. Economic Analysis

We use the previous experimental technical assessment to extrapolate to many more users for an economic assessment regarding hardware, setup, operation, and service pricing. The summary of costs per resource appear in Table 1 and the costs per number of users appear in Table 2.

| Hardware requirements | | | Costs | | | |
|-----------------------|--------------------------|--|------------------|-----------------|------------------------------|---------------|
| Resource | Per user | Per unit | Cost | Base | Per unit | Per year |
| RAM | 27.6 MB (Active sync) | 37 users (1 GB) | Server | 3,000 € | Depreciation over 4 years | 750 € |
| CPU | 0.1 (Active sync) | 10 users (1 core) | Operation | 1 hour/month | 23 €/hour | 276 € |
| Storage | 50 GB (Average) | 20 users (1 TB) | Hard disk | 0.1 €/GB | Depreciation over 4 years | 0.025 €/GB |
| Energy | – | 135 W server 30% load avg. | Energy | 355 kWh/year | 0.14 €/KWh | 50 € |

Table 1: Estimated costs of resources.

Hardware requirements Although a storage service can have multiple servers, here we focus on a single service instance. Extrapolating from the experimental results, any server instance of the service with similar hardware specifications to the one used in the experiment, per each GB of RAM, can provide service to 82 concurrent idle users or to 37 concurrent active users.³⁰ In terms of CPU, one core can serve 10 concurrent active users³¹ or virtually any number of idle users. The concurrent all-active

³⁰ $((1024\text{Mb})/(12.5\text{Mb}/\text{user}))$ and $((1024\text{Mb})/(27.6\text{Mb}/\text{user}))$, respectively.

³¹ $((1\text{cores})/(0.1\text{core}/\text{user}))$

scenario considered above, however, is very unlikely to occur, as files typically do not change often. Thus, active periods are quite infrequent and short-lived. In addition, we expect that most of the users to be disconnected most of the time, with periodic or sporadic connections to synchronise, which is a short active period between the long idle periods. Although usage patterns may vary significantly according to scenarios, we can assume a 100x ratio of disconnected over connected users at any time. In fact, memory, processing, or network congestion due to many concurrent active sessions would only lead to longer synchronisation time, which is unnoticeable in most cases. Consequently, the service resources can probably be oversubscribed with much higher order of magnitude without affecting the quality of the service.

Therefore, our 10 GB RAM and eight-core server can probably effectively handle thousands of users. In terms of data storage usage, we have considered an average of 50 GB per user (assuming an uneven distribution, with many users storing less, and a few storing much more), that is, 1 TB for every 20 users. However, in a group file-sharing scenario, or with de-duplication mechanisms, the average user usage estimation can be safely reduced.

Costs The yearly operation cost of a server, with similar specifications to the one used in the experiments in a data centre facility, excluding the hardware storage for the service, plus the personnel costs for maintaining it online is estimated as 1,076 €/year³² (last column of Table 1), for which 70% corresponds to equipment depreciation, 25% is server operation, and 5% is electrical power.³³ Although values can vary significantly in different contexts, it is interesting to note the dominance of the hardware purchase costs. The server consolidation approach can help mitigate the hardware costs notably.

We estimated the cost of the hardware for storage for the service (HDD) at 0.025 €/GB/year.³⁴ For 1,000 users (see Table 2), this represents 1,250 € of yearly pure hardware storage cost that, including the operation cost, nearly duplicates reaching 2,500 €.

The network usage can generate substantial amounts of traffic but only internal to the CN. According to the compensation system (see Section 3), that would result in a compensation fee for the network maintenance and improvement, proportional to the share of network traffic for each user. In this section, we assume that this contribution is already included in whatever network service fee the participants pay (e.g., 12 € per month in the guifi.net community in Barcelona). Traffic to and from the Internet to exchange data with a traditional Internet-based cloud provider would be more expensive, particularly in peak hours. This is an important cost saving in a CN compared to a remote cloud provider.

Service pricing Considering a constant yearly operation cost of 1,075 € for the server and a variable cost according to the required amount of storage capacity, as the number of users increases, the total storage costs grow, but the cost per GB decreases towards the pure cost of the raw storage (HDD), as the effect of operation cost becomes less significant. Table 2 summarises the minimum service pricing according to the number of subscribed users, assuming 50 GB per user, all part of the CN.

7.4. Comparison to Commercial Cloud Providers

The comparison of the results of Section 7.3 with estimates for minimum prices of commercial storage services³⁵ of large cloud providers (last column of Table 2), shows that, beyond 65 users, the cost of our service matches the standard *hot* storage service costs of the main cloud providers (Amazon S3, Microsoft Azure,

³²750 + 276 + 50

³³Reference values in our experiment. Yearly server costs: 3,000 € CAPEX over four years depreciation, 30 % load of a 135 W server at 0.14 €/KWh = 50 € electricity, 12 hours of maintenance at 23 €/hour.

³⁴Yearly storage costs: assuming 100 € purchase cost per TB of low-cost HDDs with redundancy, depreciated over four years.

³⁵We consider three classes of storage service: *hot* means high performance, such as in SSD; *cold* means low performance, with infrequent and slow read access, such as Amazon Glacier; and *warm*, with medium performance and cost.

| Total # users | Storage (TB) | Storage cost/year | Storage cost/GB/y | Service cost/year | Service cost /GB/month | Market prices (for comparison) |
|---------------|--------------|-------------------|-------------------|-------------------|------------------------|--------------------------------|
| 50 | 2.5 | 62.5 | 0.45 | 1,125 | 0.038 | N/A |
| 65 | 3.25 | 81.25 | 0.36 | 1,170 | 0.030 | 0.030 (hot) |
| 100 | 5 | 125 | 0.24 | 1,200 | 0.02 | 0.024 (warm) |
| 180 | 9 | 225 | 0.14 | 1,260 | 0.012 | 0.012 (cool) |
| 1,000 | 50 | 1,250 | 0.05 | 2,500 | 0.004 | N/A |
| 5,000 | 250 | 6,250 | 0.029 | 7,250 | 0.002 | 0.002 (HDD) |

Table 2: Estimated costs per user (monthly and yearly, in €) of the BFS service for different number of subscribed users. However, market prices that are heavily influenced by network transfers are not included (e.g., price can x4 for 10% transferred, x10 for 100%).

| Offers for a server (no storage or traffic) | Monthly cost |
|---|--------------|
| Amazon (Central Europe zone, September 2017) | |
| ‘Disaster recovery and backup’ (per server) | 233 |
| Single EC2 4 cores (m3.xlarge) | 179 |
| Single EC2 4 cores (c4.xlarge, m4.xlarge) | 138 |
| Smallest dedicated host (c3) | 1,528 |
| Community cloud (Small 4 core Intel NUC server, 33 W, 400 €) | |
| Depreciation over 3 years + energy (0.14 €/KWh) | 12 + 4 |

Table 3: Price (monthly, in €) comparison for a server between a commercial offer and a community cloud.

or Google Cloud Storage). However, these costs do not include data transfers or replication. Depending on the read/write traffic patterns of the storage usage, these costs could easily multiply, making data transfer more relevant than pure storage in the contribution to overall cost.³⁶ The references for the *warm* and *cool* storage costs are matched with 100 users and 180 users, respectively. Increasing the number of users, the server part becomes less relevant. As a reference, the pure cost of hardware storage only (HDD) corresponds to 0.002 €/GB/month (or 25 €/GB/year).

Table 3 compares the prices of a commercial server offer (Amazon, September 2017) with the cost of a small home server. A CC with these small servers running Cloudy could benefit from the difference in the cost of a server (16 € vs. 138 € or 1528 €). However, there are obvious huge differences between both models: one at home inside a community network managed by volunteers or professionals from the local community not relying on an Internet connection, the other in a remote data centre over the Internet professionally managed at huge scale. We need to account for the maintenance and operation costs of the infrastructure and the software, and the required level of replication for comparable reliability. However, the table shows that, after certain costs are included, the economies of sharing in a CC infrastructure leave an economic margin for local initiatives to provide useful local services given the small cost of sharing a set of computing, storage, and communication resources on stand-by for a pool of running services.

7.5. Findings and Implications

The technical measurements provided data to estimate costs and prices for a commercial initiative in our CC. The viability of the commercial service depends on the number of users. With 65 users, the BFS service matches the minimum current prices of commercial cloud providers operating on the Internet. With these, or any other number of users, the more read or write access to the data, the more cost-effective CCs become, due to the large difference of costs for local vs remote data transfer. Beyond that number of users, the CC solution

³⁶The pricing of commercial services depends on a detailed profile of resource consumption. For instance, there are web applications to estimate costs for Microsoft Azure <https://azure.microsoft.com/en-us/pricing/calculator/> or Amazon AWS <http://calculator.s3.amazonaws.com/index.html>.

has cheaper prices. In terms of costs, the local CC computing servers (virtualised servers or dedicated hosts) are clearly cheaper than Internet cloud providers. We have not considered the cost of software development and maintenance because there are several mature open-source software products for these functions. In our study, we assumed dedicated hardware, but slight additional economies of sharing can be achieved through IaaS CPR. Considering locality in terms of latency, performance, independence from Internet connectivity when it is fragile, and data protection, we can conclude that the CC is reasonably attractive for end users, and it appears to be financially viable, advantageous in cost, and therefore sustainable. There is the additional benefit of promoting local money flows, that is, contributing to local socio-economic development. Furthermore there is a preference in communities to keep data nearby, for privacy and availability reasons, away from excessive data retention and potential exploitation from cloud providers, blockade from attackers or censorship from governments. These results, combined with the discussions of other aspects that emerged in the pilot, allow to qualify the levels of feasibility and sustainability reached.

8. Discussion

In previous sections, we elaborated on key aspects of the governance and implementation of CCs, which determine the degree of technological readiness (feasibility) and (organisational, governance, and economic) sustainability of CCs as commons, which relate to our main research questions. Here we discuss several reflections about factors and artefacts that appeared during the collaboration and discussions, as part of the action research, with the guifi.net community and their participants directly involved: 1) the nature of CCs as CPR, in Section 8.1; 2) the development of the software implementation, in Section 8.2; 3) the experience gained with the deployed system and prototype storage service as it relates to other user-oriented services, in Section 8.3. For each topic, we first describe the idea and its relevance, and then we discuss the analysis and lessons learned.

8.1. Community Cloud Infrastructure as a Common-Pool Resource

Despite the lack of benefits from economies of scale in unit costs compared to large remote cloud data centres, local infrastructures have their own technological advantages in terms of latency, better connectivity with end users, partition tolerance, or faster and more regular transfer rates. As a result of local cooperation and sharing, users enjoy cheaper or even free local network traffic, cheaper local cloud infrastructure, and socio-economic advantages from local interactions, investing and contributing to the local economy and promoting local development. The most relevant factors that emerged during the pilot are detailed next.

Participation with contributions Users must be willing to participate with contributions to the CC. These contributions can be in terms of hardware, locations, maintenance, etc. The entry cost is relatively small: a low-capacity and low-power device³⁷ is enough to start. The installation process is relatively simple, and the devices require little maintenance. These tasks can even be delegated to a friend or trusted member of the community. A large body of research and practical experience in the past has already shown that users are often willing to donate spare computing resources to third-party propositions in exchange for added value [61]. Economic or social incentive mechanisms building on that have been proposed in CCs [62] and CNs [50] research.

Access to nearby strategic locations (e.g., homes, service rooms in buildings, street cabinets, warehouses, and public data centres) is another beneficial factor from CPR clouds. Many key local public and private stakeholders can offer these locations to voluntary and commercial service providers for the local interest. The locations of the resources are valuable assets for enabling innovative locality-critical services without relying on Internet connectivity. In addition, privacy regulations and privacy protection can be more easily fulfilled if a commercial service runs on local CPR-provided cloud infrastructures due to lower costs and more opportunities to audit and control the collection of personal data. Such an approach has enormous potential to address use cases in the smart-city and IoT domains.

³⁷e.g., mini-computers or single-board computers, such as the Raspberry Pi or a NUC device.

Locality and cooperative cost sharing Therefore, there is a local opportunity to beat or complement traditional Internet cloud infrastructure and service providers in terms of pricing or features that stem from locality and the cooperative model of cost sharing in CNs [63], taking advantage of the shared collective network and cloud infrastructure. In such a model, local entrepreneurs can venture in setting up garage or warehouse data centres, or established organisations, like companies, governments, schools, farms, or factories, can lower their costs by venturing into the IaaS business for local cloud users.

Cost-oriented pricing for local sustainability From a business perspective, CCs offer resources and basic services according to a cooperative commons model with a cost-oriented pricing. The costs should include fair remuneration and local reinvestment to preserve the sustainability of the commons. For SMEs and entrepreneurs, the CPR is a suitable context for experimentation and learning and an opportunity to explore commercial services without a strong initial risk or capital investment. Research about cost sharing in CNs [63] explores this factor.

Voluntary and professional effort In the commons, we find coexistence and cooperation between voluntary and professional schemes. The first usually comes with no service commitment (best effort or less), while the second involves specific commitments (SLO) in exchange for service fees. Ideally, both schemes can complement each other. Volunteers can buy and include professionally maintained resources in their voluntary efforts, and professionals can occasionally leverage from voluntary resources or efforts in their services (e.g., software tools and peer-assisted cloud services) in exchange for contributions back to the commons. This close collaboration allows pooling and growth at smaller steps with smaller upfront costs. The BFS service, for instance, can be deployed on a hybrid schema combining a stable server infrastructure with less stable resources from volunteers to handle peaks or growth at the expense of a compensation fee. Local currencies have been explored [51] to regulate these transfers without involving fiat money.

Influence of underlying conditions Community clouds can, in principle, develop over any kind of network infrastructure. While the commoditisation of hardware (low cost), software (free and open), networking (flat rates), and access to locations (facilities for deployment and right of pass) is a major enabling factor for sharing, the major barriers come from differentiation, such as with traffic-based (instead of capacity-based) charging or lack of traffic neutrality (when ISPs do traffic discrimination by throttling or blocking local servers and promote paid fast lanes for major Internet cloud and content providers), or provide asymmetric access (a TV-like Internet).

Scalability Local community commons can also benefit from economies of scale through the local aggregation of resources (e.g., pooling of needs for data stores, remote backups, server consolidation for public websites, databases, hosts, and containers). This way, a group of small and medium local cloud users can act as wholesale intermediaries and benefit from volume pricing from large Internet cloud providers. This requires interoperability and federation mechanisms to group resource needs and to transparently integrate Internet cloud providers as a backend. The wholesale access to Internet carriers or Voice over IP providers by ISPs involved in the guifi.net CN or the digital exchanges discussed in Section 4.1 are good real examples of that.

Second-layer organisations A cloud commons can act together as a larger organisational umbrella representing and protecting their collective interests in the face of external agents, like governments, companies, users, regulators, research organisations, investors, and standardisation bodies. Guifi.net Foundation, which has played that role for the CN, sees similar challenges for CCs, and an umbrella organisation can represent and more effectively protect a group of emerging local CCs better than CCs can individually.

Validation of the CC model Regarding the validation of the CC commons model, our experience with the guifi.net CC shows the value of the scenario of ‘platform commons’ (scenario *c* in Figure 3) but also its difficulties. We managed to work with different stakeholders, for example, many volunteers and

two professionals interested in the feasibility and sustainability analysis of the file storage service. We worked with software developers. Some were supported by research projects, and others as volunteers, and we counted on the support of Guifi.net Foundation. In the context of the participation framework, we started the dialogues for the CCCL and the compensation system, and although such social processes of deliberation take time to deliver results, the interim results are positive. Therefore, we can conclude we have developed a functional organisational and governance model for a CC infrastructure as an open commons (CPR) reaching technology readiness level 9 ‘actual system proven in operational environment’ [35], in the specific environment of the guifi.net community. Applied to other different environments, that may require adaptation, it can be considered at least a ‘field demonstration’ (TRL 6) or ‘system prototype demonstration in operational environment’(TRL 7). From the perspective of demand-readiness levels [36], we have moved from the feeling of ‘something is missing’ in level 1 to ‘building the adapted answer to the expressed need in the market’ in level 9, in the specific environment of the guifi.net community. Furthermore, the model and its discussion can be applied to other communities, as the only requirements are the feasibility factors described in Section 3.6, and then the pooling by the CC community of networking, computing, or storage resources.

8.2. Community Cloud Software Platform

Throughout our research and interaction with the pilot community and other interested CNs, we identified several factors and artefacts related to the software platform that are detailed next.

Emphasis on digital self-reliance or self-determination This translates in a preference for resilient decentralised CC infrastructures, capable of continuing to deliver their services when facing functional, environmental, or technological changes to a degree that they can support and enhance other critical local infrastructures, services, and applications. We have also perceived a growing concern about the risks of extraction and exploitation of personal data. Some of the main domains we have identified, where these aspects are particularly relevant, include smart infrastructures, such as energy, transportation, waste and sewage, and communications; services for citizens, governments, and industry based on smart monitoring and control; and applications for smart communities to improve the efficiency of services and meet residents’ needs.

Software tools and services for the deployment and operation of a CC infrastructure We observe that, similar to the operation of the network infrastructure, a set of software tools and services are needed to ease the tasks of deploying the cloud infrastructure and coordinating its operation and usage. The Cloudy software stack we developed materialises the most urgent services we have identified and facilitates the adoption of the required software components by the users, such as the common user authentication and the DADS services. Therefore, Cloudy is expected to help standardise and unify the cloud infrastructure and service level in CNs, similarly as guifi.net has achieved through the website and other tools at the network infrastructure level.

The complexity of software However, this results in software complexity that requires a committed community of developers, maintainers, and users. The guifi.net experience shows that an effective approach to face this challenge is through a combination of volunteers (for testing, reporting bugs, making suggestions, contributing code, documentation, etc.) and professionals (for delivering professional code). To this end, it is necessary to develop a value chain that ensures a minimum revenue stream to pay the professional developers and volunteers. Software that can be customised to work for multiple CCs creates opportunities for more sustainable crowdfunding of these developments. Moreover, this surplus must also allow maintaining, updating, and expanding the existing hardware to keep the value chain alive and avoid a spiral of decline, which is more pronounced in low-cost devices (e.g., we have experienced cases of nodes suffering from cell burnout in compact flash storage and power supply burnout).

Resilience Compared to proprietary cloud environments, community-owned clouds are prone to node failures given their more fragile environment. Replication techniques are therefore essential for resilience

so that users will not lose their data. For instance, the Tahoe-LAFS data objects are typically stored on 10 nodes of the storage server pool, but only any three of these are required to retrieve the data. Obviously, this resilience has a cost in terms of network traffic and storage capacity.

Validation of the CC software platform We can conclude that we have reached TRL 9 ‘actual system proven in operational environment’ [35], or at least TRL 7 in other different environments. Although being in continuous evolution and improvement, the software platform is applicable to other communities. Adoption by other communities will define new requirements but will also bring in new contributions. This is the case with a new community in Italy that is preparing a pilot trial of Cloudy.

8.3. Tailored Services at the Network Edge and Effects on Users

Our exercise of working with a community is consistent with other experiences of collaborative system design. Several factors that emerged during the pilot are detailed next.

Collaborative system design Identifying requirements and proposing useful applications is a community building exercise that creates social bonds between experimenters, early adopters, and developers. Sharing knowledge with the rest of the community helps motivate developers to create new services of interest even outside the initial community.

Crowdfunding model of local services In an open environment for innovation, crowdfunding campaigns can sponsor the development of new applications of local interest or of a specialised sector. Some cases from discussions with the volunteers include services required by and for local governments, SMEs, farms, factories, schools, and citizens, regarding environmental monitoring, control of public and private infrastructures, and security or emergencies. A model based on several simple but adapted local solutions can flourish in CCs, opening new markets for the local economy. The idea of micro-services follows this trend [64].

Complementarity to traditional cloud services The presented CC model is complementary to traditional cloud service provisions and can benefit from the integration with it. The inter-cloud integration [65] has been reported to be beneficial in this regard. However, it must be considered that the resources and services at the edge are also more fragile. Thus, the SLAs and guarantees are difficult to achieve for high-demand services [39]. Therefore, interoperability with industry standards like Open-Stack or APIs from Internet cloud providers are desirable to merge the benefits of services running on ‘big iron’ clusters and data centres with the local decentralised structures that Cloudy supports. The resulting hybrid clouds pose interesting research and innovation challenges like the federation of very diverse computing, storage, and networking resources.

Validation of the file storage SaaS Regarding this storage service, we can conclude that we also have reached not only TRL 7 but also reached TRL 9, which implies identifying and implementing the social and technological factors to bootstrap (feasibility) but also to sustain the service (sustainability). The results are relevant and applicable to other end-user oriented applications and other communities adopting the CC model. As Ostrom said [30], regimes should adapt to local conditions, not the other way around.

9. Conclusion

Community clouds (CCs) at the network edge are motivated by their disruptive potential for changing the future cloud service landscape by expanding the current cloud service offerings with local cloud resources and service infrastructures open for access (usage) and open for participation (construction, operation, and governance). The paper presents an analysis, design, and evaluation of the artefacts and enabling factors for the feasibility and sustainability of local CCs, through the development of an organisational and governance model, a system architecture, its hardware/software implementation in Cloudy, and the technical

and economic experimental evaluation of a storage application. The research combines action research with a community of users from the guifi.net CN and experimentally driven research on a real deployment.

The first key aspect is about organisational models. To this end, the paper first reviews the mechanisms that have led the guifi.net CN, a case of network infrastructure as open commons, to start up and become sustainable. Then, specific issues for the applicability of these mechanisms in CC-based services are discussed. The model we propose, inspired by the experience of guifi.net but applicable to other communities, implements a CC with the IaaS and PaaS layers organised as an open CPR. A framework of tools (artefacts) to govern such a CC is presented. Some of these components have been already implemented in the Cloudy software stack.

The software implementation and its deployment in the field is the second key aspect. The overall CC model, implementation, pilot deployment, and services have been put into practice in guifi.net for more than two years of operation with about 60 end-user devices and 64 participants from the community. The results of that experimentation confirm the feasibility of CCs as open commons in a real setting (TRL 9 in the guifi.net environment, TRL 7 for others), from the concept to the implementation and its governance.

The Cloudy software stack can be extended by its users with new application-oriented services (SaaS). As such, the third key aspect is the evaluation of a storage application service from the technological and socio-economic viewpoints to assess the technical feasibility and the economic sustainability of CC application services in comparison to commercial cloud solutions. The technical analysis shows the application can be implemented and integrated in Cloudy and it operates correctly. The economic analysis for the storage application service shows competitive long-term costs from a rather small number of users. The extended period of experimentation with the guifi.net CC, combined with the technological and socio-economic analysis of a representative SaaS, confirms the potential for the sustainability of CCs as open commons.

In general, we identify a set of key enabling factors for feasibility (demand, early adopters, technology availability, ease of participation, and licence), and a set of governance artefacts/tools (governance: licence and collaboration agreements; socio-economic: conflict resolution, sanctioning system, economic compensation; and coordination: management and provisioning, and communication) for sustainable CPR CCs.

Although we provide compelling findings for an ‘adapted answer to the expressed need’ (DRL 9), further studies are required to extend the results to other services, and evaluate replication in other regions, other socio-economic environments, other CNs, and on a larger scale. It is interesting to note that CCs as open commons, despite being inspired by CNs, can develop in diverse environments, including Internet access networks provided by commercial ISPs.

The next steps will include the development of tools (artefacts) to measure and account for the contribution to and usage of the cloud CPR to be able to share the costs fairly and promote reinvestment. The introduction of additional close-to-market and locality-sensitive services using the CC CPR help further identify and mitigate the risks, understand the factors that enable or facilitate the replication and growth of the model elsewhere, and contribute to optimising the business model for CCs as commons.

Acknowledgments

The authors want to thank the valuable comments of Agustí Moll, Roger Pueyo, Emmanouil Dimogerontakis, Andrés Arcia-Moret, Jon Crowcroft and Marc Perea. We also thank the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions. This work was supported by the European Community Framework Programme 7, FIRE Initiative projects “Community Networks Testbed for the Future Internet” (CONFINE) FP7-288535 and “A Community Networking Cloud in a Box” (CLOMMUNITY) FP7-317879 (2013–2016), the Spanish government TIN2016-77836-C2-2-R, and the European Community H2020 Programme netCommons H2020-688768 (2016–2018) and “architectuRe for an Internet For Everybody” (RIFE) H2020-644663 (2015–2018) projects.

References

- [1] D. Bollier, *Think Like a Commoner: A Short Introduction to the Life of the Commons*, New Society Publishers, 2014.
URL <https://books.google.co.uk/books?id=C-DOAgAAQBAJ>

- [2] Y. Benkler, *The Penguin and the Leviathan: The Triumph of Cooperation Over Self-interest*, Crown Business, 2011.
URL <https://books.google.co.uk/books?id=0jiJm41QW8YC>
- [3] 2017 affordability report, alliance for Affordable Internet (2017).
URL <http://a4ai.org/affordability-report/report/2017/>
- [4] S. Murugesan, I. Bojanova, *Encyclopedia of Cloud Computing*, John Wiley & Sons, 2016.
- [5] L. Navarro, R. Baig, F. Freitag, E. Dimogerontakis, F. Treguer, M. Dulong de Rosnay, L. Maccari, P. Micholia, P. Antoniadis, Report on the Existing CNs and their Organization (v2), netCommons Deliverable D1.2, Sept. 2016.
URL <http://netcommons.eu/?q=content/report-existing-cn-and-their-organization-v2>
- [6] Y. Benkler, H. Nissenbaum, Commons-based peer production and virtue, *Journal of Political Philosophy* 14 (4) (2006) 394–419.
- [7] J. A. Tacchi, D. Slater, G. N. Hearn, *Ethnographic Action Research: A User’s Handbook*, UNESCO, New Delhi, India, 2003.
URL <https://eprints.qut.edu.au/4399/>
- [8] A. Gavras, A. Karila, S. Fdida, M. May, M. Potts, Future internet research and experimentation: The fire initiative, *SIGCOMM Comput. Commun. Rev.* 37 (3) (2007) 89–92. doi:10.1145/1273445.1273460.
URL <http://doi.acm.org/10.1145/1273445.1273460>
- [9] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC ’12, ACM, New York, NY, USA, 2012, pp. 13–16. doi:10.1145/2342509.2342513.
URL <http://doi.acm.org/10.1145/2342509.2342513>
- [10] P. Garcia, et al., Edge-centric Computing: Vision and Challenges, *ACM SIGCOMM Computer Communication Review* 45 (5) (2015) 37–42.
- [11] L. M. Vaquero, L. Rodero-Merino, Finding your way in the fog: Towards a comprehensive definition of fog computing, *SIGCOMM Comput. Commun. Rev.* 44 (5) (2014) 27–32. doi:10.1145/2677046.2677052.
URL <http://doi.acm.org/10.1145/2677046.2677052>
- [12] W. Shi, S. Dustdar, The promise of edge computing, *IEEE Computer* 49 (5) (2016) 78–81. doi:10.1109/MC.2016.145.
URL <http://dx.doi.org/10.1109/MC.2016.145>
- [13] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, E. Riviere, Edge-centric computing: Vision and challenges, *SIGCOMM Comput. Commun. Rev.* 45 (5) (2015) 37–42. doi:10.1145/2831347.2831354.
URL <http://doi.acm.org/10.1145/2831347.2831354>
- [14] A. Marinos, G. Briscoe, Community Cloud Computing, *Cloud Computing* 5931 (December) (2009) 472–484. arXiv: 0907.2485, doi:10.1007/978-3-642-10665-1_43.
- [15] NIST, U. S. Department of Commerce, The NIST Definition of Cloud Computing.
URL <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [16] A. Li, X. Yang, S. Kandula, M. Zhang, Cloudcmp: comparing public cloud providers, in: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pp. 1–14.
- [17] X. Wen, G. Gu, Q. Li, Y. Gao, X.-J. Zhang, Comparison of open-source cloud management platforms: Openstack and opennebula., in: FSKD, IEEE, 2012, pp. 2457–2461.
URL <http://dblp.uni-trier.de/db/conf/fskd/fskd2012.html#WenGLGZ12>
- [18] NYSE Technologies Introduces the World’s First Capital Markets Community Platform.
URL <http://www1.nyse.com/press/1306838249812.html>
- [19] Optum Introduces Health Care Cloud Environment that Enables Secure Collaboration, Enhances Patient Care and Speeds Innovation.
URL http://www.unitedhealthgroup.com/newsroom/articles/news/optum/2012/0214cloud.aspx?sc_lang=en
- [20] E. Brewer, Cap twelve years later: How the ”rules” have changed, *Computer* 45 (2) (2012) 23–29. doi:10.1109/MC.2012.37.
- [21] P. M. Mell, T. Grance, Sp 800-145. the nist definition of cloud computing, Tech. rep., National Institute of Standards & Technology, Gaithersburg, MD, United States (2011).
URL <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>
- [22] S. K. Patel, V. R. Rathod, S. Parikh, Joomla, drupal and wordpress-a statistical comparison of open source cms, in: *Trendz in Information Sciences and Computing (TISC)*, 2011 3rd International Conference on, IEEE, 2011, pp. 182–187.
- [23] P. G. López, M. S. Artigas, C. Cotes, G. Guerrero, A. Moreno, S. Toda, Stacksync: Architecturing the personal cloud to be in sync, Tech. rep., Universitat Rovira i Virgili (2013).
- [24] F. E. Gillett, C. Mines, T. Schadler, M. Yamnitsky, H. Shey, A. Martland, R. Iqbal, The personal cloud: transforming personal computing, mobile, and web markets, Tech. rep., Forrester (2011).
- [25] G. Banga, P. Druschel, J. C. Mogul, Resource containers: A new facility for resource management in server systems, in: *OSDI*, Vol. 99, 1999, pp. 45–58.
- [26] D. Merkel, Docker: lightweight linux containers for consistent development and deployment, *Linux Journal* 2014 (239) (2014) 2.
- [27] D. Bernstein, Containers and cloud: From lxc to docker to kubernetes, *IEEE Cloud Computing* 1 (3) (2014) 81–84. doi:10.1109/MCC.2014.51.
- [28] N. Rameshan, L. Navarro, E. Monte, V. Vlassov, Stay-away, protecting sensitive applications from performance interference, in: Proceedings of the 15th International Middleware Conference, Middleware ’14, ACM, New York, NY, USA, 2014, pp. 301–312. doi:10.1145/2663165.2663327.
URL <http://doi.acm.org/10.1145/2663165.2663327>

- [29] N. Rameshan, Y. Liu, L. Navarro, V. Vlassov, Hubbub-scale: Towards reliable elastic scaling under multi-tenancy, in: IEEE/ACM 16th International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2016, Cartagena, Colombia, May 16-19, 2016, IEEE Computer Society, 2016, pp. 233–244. doi:10.1109/CCGrid.2016.71.
URL <https://doi.org/10.1109/CCGrid.2016.71>
- [30] E. Ostrom, *Governing the commons: the evolution of institutions for collective action*, Cambridge University Press, 1990.
URL <http://www.cambridge.org/us/academic/subjects/politics-international-relations/political-economy/governing-commons-evolution-institutions-collective-action>
- [31] B. M. Frischmann, *Infrastructure commons in economic perspective*, First Monday 12 (6).
URL <http://firstmonday.org/ojs/index.php/fm/article/view/1901>
- [32] B. M. Frischmann, *Infrastructure: The social value of shared resources*, Oxford University Press, 2012.
- [33] Y. Benkler, Some economics of wireless communications, *Harv. JL & Tech.* 16 (2002) 25.
URL http://heionlinebackup.com/hol/cgi-bin/get_pdf.cgi?handle=hein.journals/hjlt16§ion=7
- [34] R. Baig, R. Roca, F. Freitag, L. Navarro, guifi.net, a crowdsourced network infrastructure held in common, *Computer Networks* 90 (2015) 150 – 165, crowdsourcing. doi:<http://dx.doi.org/10.1016/j.comnet.2015.07.009>.
URL <http://www.sciencedirect.com/science/article/pii/S1389128615002327>
- [35] E. Commission, *Technology readiness levels (trl)*, G. Technology readiness levels (TRL), HORIZON 2020 – WORK PROGRAMME 2014-2015 General Annexes, Commission Decision C(2014)4995.
- [36] W. P. 11-01, *Technology and market readiness levels*, Tech. rep., Dent Associates (2011).
URL <https://www.dentassociates.co.uk/wpdivi/wp-content/uploads/2015/09/Technology-and-Market-Readiness-Levels.pdf>
- [37] N. Haile, J. Altmann, Value creation in it service platforms through two-sided network effects, in: *International Conference on Grid Economics and Business Models*, Springer, 2012, pp. 139–153.
- [38] N. Haile, J. Altmann, Value creation in software service platforms, *Future Generation Computer Systems* 55 (2016) 495 – 509. doi:<http://dx.doi.org/10.1016/j.future.2015.09.029>.
URL <http://www.sciencedirect.com/science/article/pii/S0167739X1500309X>
- [39] I. Petri, O. F. Rana, G. C. Silaghi, Y. Rezgui, Risk assessment in service provider communities, *Future Generation Computer Systems* 41 (2014) 32 – 43. doi:<http://dx.doi.org/10.1016/j.future.2014.08.013>.
URL <http://www.sciencedirect.com/science/article/pii/S0167739X14001617>
- [40] K. Danniswara, H. P. Sajjad, A. Al-Shishtawy, V. Vlassov, Stream processing in community network clouds, in: *Future Internet of Things and Cloud (FiCloud)*, 2015 3rd International Conference on, 2015, pp. 800–805. doi:10.1109/FiCloud.2015.95.
- [41] M. Henze, J. Hiller, O. Hohlfeld, K. Wehrle, Moving Privacy-Sensitive Services from Public Clouds to Decentralized Private Clouds, in: *2016 IEEE International Conference on Cloud Engineering Workshops*, 2016, pp. 130–135. doi:10.1109/IC2EW.2016.24.
- [42] N. Powers, T. Soyata, Axaas (acceleration as a service): Can the telecom service provider rent a cloudlet?, in: *Cloud Networking (CloudNet)*, 2015 IEEE 4th International Conference on, 2015, pp. 232–238. doi:10.1109/CloudNet.2015.7335314.
- [43] K. Yoi, H. Yamaguchi, A. Hiromori, A. Uchiyama, T. Higashino, N. Yanagiya, T. Nakatani, A. Tachibana, T. Hasegawa, Multi-dimensional sensor data aggregator for adaptive network management in m2m communications, in: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 1047–1052. doi:10.1109/INM.2015.7140431.
- [44] R. Tolosana-Calasan, J. A. Bañares, C. Pham, O. F. Rana, Revenue-based resource management on shared clouds for heterogenous bursty data streams, in: *Proceedings of the 9th International Conference on Economics of Grids, Clouds, Systems, and Services, GECON’12*, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 61–75. doi:10.1007/978-3-642-35194-5_5.
URL http://dx.doi.org/10.1007/978-3-642-35194-5_5
- [45] R. Tolosana-Calasan, J. A. Bañares, C. Pham, O. F. Rana, Resource management for bursty streams on multi-tenancy cloud environments, *Future Gener. Comput. Syst.* 55 (C) (2016) 444–459. doi:10.1016/j.future.2015.03.012.
URL <http://dx.doi.org/10.1016/j.future.2015.03.012>
- [46] R. Baig, L. Dalmau, R. Roca, L. Navarro, F. Freitag, A. Sathiseelan, Making community networks economically sustainable: The guifi.net experience, *ACM SIGCOMM Global Access to the Internet for All (GAIA) Workshop 1*. doi:10.3200/ENVT.50.4.8-21.
- [47] V. Vlassov, J. Dowling, A. Khan, M. Selimi, F. Freitag, L. Navarro, D. e. a. Vega, Deliverable d3.4: Holistic abstract network and service architecture (2015).
URL http://felix.site.ac.upc.edu/Clommunity_deliverables/D3_4.pdf
- [48] M. Forzati, C. P. Larsen, C. Mattsson, Open access networks, the swedish experience, in: *12th International Conference on Transparent Optical Networks (ICTON)*, 2010, pp. 1–4. doi:10.1109/ICTON.2010.5549139.
- [49] N. Facchi, F. Freitag, L. Maccari, P. Micholia, F. Zanini, Release of All Open Source Software for all Applications (v1), netCommons Deliverable D3.2, Dec. 2016.
URL <http://netcommons.eu/?q=content/release-new-open-source-software-all-applications-v1>
- [50] P. Antoniadis, M. Karaliopoulos, G. Klissiaris, I. Koutsopoulos, P. Micholia, Incentives for Participation and Active Collaboration in CNs (v1), netCommons Deliverable D2.3, Jan. 2017.
URL <http://netcommons.eu/?q=content/incentives-participation-cns-v1>
- [51] P. Antoniadis, J. Martignoni, L. Navarro, Economic Sustainability of CNs (v1) Introducing Community Currencies, netCommons deliverable D2.4, Dec. 2016.
URL <http://netcommons.eu/?q=content/economic-sustainability-cns-v1>
- [52] U. Gneiting, T. Bruno-van Vijfeijken, H. P. Schmitz, R. Gomez, Setting higher goals: Rights and development, trade-offs

- and challenges in implementing a rights-based approach to development, Monday Developments.
- [53] Y. Liu, N. Rameshan, E. Monte, V. Vlassov, L. Navarro, Prorenata: Proactive and reactive tuning to scale a distributed storage system, in: Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on, IEEE, 2015, pp. 453–464.
 - [54] HashiCorp, Serf: a decentralized solution for cluster membership, failure detection, and orchestration., [Online; accessed 22-July-2016] (2016).
URL <https://www.serfdom.io/>
 - [55] M. Selimi, A. M. Khan, E. Dimogerontakis, F. Freitag, R. P. Centelles, Cloud services in the guifi.net community network, *Comput. Netw.* 93 (P2) (2015) 373–388. doi:10.1016/j.comnet.2015.09.007.
URL <http://dx.doi.org/10.1016/j.comnet.2015.09.007>
 - [56] M. Selimi, F. Freitag, L. Cerdà-Alabern, L. Veiga, Performance evaluation of a distributed storage service in community network clouds, *Concurrency and Computation: Practice and Experience* 28 (11) (2015) 3131–3148, cpe.3658. doi:10.1002/cpe.3658.
URL <http://dx.doi.org/10.1002/cpe.3658>
 - [57] M. Selimi, N. Apollonia, F. Olid, F. Freitag, L. Navarro, A. Moll, R. Pueyo, L. Veiga, Integration of assisted p2p live streaming service in community network clouds, in: to be published in 7th IEEE International Conference on Cloud Computing Technology and Science CloudCom (2015)), IEEE, 2015, pp. 202–209.
 - [58] M. Selimi, F. Freitag, R. Pueyo Centelles, A. Moll, L. Veiga, Trobador: Service discovery for distributed community network micro-clouds, in: *Advanced Information Networking and Applications (AINA)*, 2015 IEEE 29th International Conference on, 2015, pp. 642–649. doi:10.1109/AINA.2015.248.
 - [59] M. Selimi, A. M. Khan, E. Dimogerontakis, F. Freitag, R. P. Centelles, Cloud services in the guifi.net community network, *Computer Networks* 93, Part 2 (2015) 373 – 388, community Networks. doi:<http://dx.doi.org/10.1016/j.comnet.2015.09.007>.
URL <http://www.sciencedirect.com/science/article/pii/S1389128615003175>
 - [60] E. Ostrom, The challenge of common-pool resources, *Environment: Science and Policy for Sustainable Development* 50 (4) (2008) 8–21. doi:10.3200/ENVT.50.4.8-21.
 - [61] D. P. Anderson, E. Korpela, R. Walton, High-performance task distribution for volunteer computing, in: *Proceedings of the First International Conference on e-Science and Grid Computing, E-SCIENCE '05*, IEEE Computer Society, Washington, DC, USA, 2005, pp. 196–203. doi:10.1109/E-SCIENCE.2005.51.
URL <http://dx.doi.org/10.1109/E-SCIENCE.2005.51>
 - [62] A. M. Khan, Ümit C. Büyüksahin, F. Freitag, Incentive-based resource assignment and regulation for collaborative cloud services in community networks, *Journal of Computer and System Sciences* 81 (8) (2015) 1479 – 1495. doi:<http://dx.doi.org/10.1016/j.jcss.2014.12.023>.
URL <http://www.sciencedirect.com/science/article/pii/S0022000014001871>
 - [63] R. R. L. N. F. F. A. S. R. Baig, L. Dalmau, Making community networks economically sustainable, the guifi.net experience, in: *ACM SIGCOMM Global Access to the Internet for All (GAIA) Workshop, GAIA '16*, 2016, pp. 31–36.
 - [64] M. Selimi, L. Cerdà-Alabern, M. Sánchez-Artigas, F. Freitag, L. Veiga, Practical service placement approach for microservices architecture, in: *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, IEEE Press, 2017, pp. 401–410.
 - [65] R. Buyya, R. Ranjan, R. N. Calheiros, Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services, in: *Algorithms and Architectures for Parallel Processing, LNCS*, 2010, pp. 20–31.