Case definition and evaluation criteria

Deliverable number: D4.1

Version 0.2
**Project Acronym:** Clommunity  
**Project Full Title:** A Community networking Cloud in a box  
**Type of contract:** Small or medium-scale focused research project (STREP)  
**contract No:** 317879  
**Project URL:** http://clommunity-project.eu

<table>
<thead>
<tr>
<th>Editor:</th>
<th>Guifi.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deliverable nature:</td>
<td>Report (R)</td>
</tr>
<tr>
<td>Dissemination level:</td>
<td>Public (PU)</td>
</tr>
<tr>
<td>Contractual Delivery Date:</td>
<td>April 30, 2014</td>
</tr>
<tr>
<td>Actual Delivery Date:</td>
<td>July 15, 2014</td>
</tr>
<tr>
<td>Suggested Readers:</td>
<td>Project partners</td>
</tr>
<tr>
<td>Number of pages:</td>
<td>35</td>
</tr>
<tr>
<td>Keywords:</td>
<td>WP4, testbed design, requirement analysis, experimental research, community networks, testbed</td>
</tr>
</tbody>
</table>
| Authors:         | Agustí Moll, Pau Escrich, Roger Baig, Roger Pueyo (Guifi.net)  
|                  | Felix Freitag, Mennan Selemi (UPC) |
| Peer review:     | Amin Khan (UPC)        |

**Abstract**

This document comprises the case definition, performance objectives, evaluation criteria and metrics for the CLOMMUNITY community networking testbed.

Use cases are divided into four domains. The first domain for use cases focuses on the setup, usage and maintenance of Infrastructure Services for Community Clouds (ISCC). The second one looks at Networking Services for Community Clouds (NSCC) that provide network layers for interconnecting cloud services on top. The third domain, about Platform Services for Community Clouds (PSCC), discusses the platform services needed to run applications on top. Finally, the fourth domain is centred on user-oriented Software Services for Community Clouds (SSCC), containing the applications that provide cloud services to community network end users. After defining the use cases, the state of the art of Community Networking Clouds is analysed, focusing on the status of the Guifi.net community. This considered, the performance objectives and their evaluation criteria and metrics are established.
# Contents

1 Introduction .................................................. 5
   1.1 Contents of this deliverable .............................. 5
   1.2 Terminology ............................................. 5
   1.3 Relations of this document with other CLOMMUNITY tasks and deliverables .......................... 6

2 Definition of use cases ........................................ 7
   2.1 Infrastructural services for community clouds ............... 7
      2.1.1 Virtual machines .................................... 7
      2.1.2 A community distribution to provide infrastructural services ................................ 8
   2.2 Networking services for community clouds .................. 9
      2.2.1 Overlay network (L2 over L3) .......................... 9
   2.3 Platform services for community clouds .................... 10
      2.3.1 Service monitoring .................................. 10
      2.3.2 User authentication .................................. 11
      2.3.3 Service announcement ................................ 11
      2.3.4 Service discovery .................................... 12
      2.3.5 Database ............................................. 12
   2.4 Software services for community clouds ................... 12
      2.4.1 Data storage service ................................ 13
      2.4.2 Media streaming service .............................. 14
   2.5 Non-technical aspects ..................................... 14
      2.5.1 User engagement and participation .................... 14
      2.5.2 Privacy awareness .................................... 15
   2.6 General considerations ..................................... 15
      2.6.1 Free, Libre, Open Source Software (FLOSS) ............... 15
      2.6.2 Distribution and decentralisation ..................... 15

3 State of the art of Community Networking Clouds ............... 16
   3.1 Infrastructural services for community clouds ................ 17
      3.1.1 Virtual machines .................................... 18
      3.1.2 Legacy Guifi.net distribution (Guinux) ................. 18
   3.2 Networking services for community clouds .................. 20
      3.2.1 Overlay network (L2 over L3) .......................... 20
   3.3 Platform services for community clouds .................... 20
      3.3.1 Service monitoring .................................. 21
      3.3.2 User authentication .................................. 21
      3.3.3 Service announcement ................................ 22
      3.3.4 Service discovery .................................... 22
      3.3.5 Database ............................................. 22
   3.4 User-oriented services for community clouds ................ 22
      3.4.1 Data storage service ................................ 22


List of Figures

3.1 Guinux screenshots ........................................... 19
List of Tables

1.1 List of terminology equivalences between generic clouds and Community Network (CN) clouds ................................................................. 6
3.1 List of network-related Guifi.net services in Catalonia and Barcelonès areas ........ 16
3.2 List of user-focused Guifi.net services in Catalonia and Barcelonès areas ........... 17
4.1 ISCC cases performance objectives ............................................. 26
4.2 NSCC cases performance objectives ........................................... 26
4.3 PSCC cases performance objectives .......................................... 27
4.4 SSCC cases performance objectives .......................................... 27
4.5 Non-technical cases performance objectives ............................... 27
4.6 Common performance objectives .............................................. 28
5.1 ISCC cases metrics and evaluation ........................................... 29
5.2 NSCC cases metrics and evaluation .......................................... 29
5.3 PSCC cases metrics and evaluation .......................................... 30
5.4 SSCC cases metrics and evaluation .......................................... 30
5.5 Non-technical cases metrics and evaluation ............................... 31
1 Introduction

Community networks (CNs) are a particular type of telecommunication infrastructures in which communities of citizens build, own and operate IP-based networks, open for individual and collective participation. In these networks, stakeholders established in a geographic area or community (individuals, schools, small and medium enterprises (SMEs), professional associations, municipalities, etc.) team up to plan, create, invest on and run an open network infrastructure based on the concepts of self-service and self-management. These efforts are aimed at satisfying the community’s demands such as broadband Internet access, Voice over IP (VoIP), and many other user-oriented services, helping digital inclusion in domains outside the commercial interest of traditional Internet Service Providers (ISPs) and Application Service Providers (ASPs).

Nowadays CNs resources are mainly used as a mere transport network to access foreign, distant cloud services for their daily purposes (e-mail, data storage, etc.), much like with traditional ISPs. This means that the potential of CNs is not fully exploited. For instance, one of the differential traits in CNs is the bandwidth symmetry of the links, as opposed to most commercial ISPs. This makes CNs specially suitable for hosting local services and applications. Bringing the cloud to CNs is a way to take advantage of their special characteristics. CLOMMUNITY is aimed at improving cloud services in the context of CNs, by providing a set of tools and resources for end users to facilitate their transition and adoption of cloud services inside CNs.

1.1 Contents of this deliverable

In this deliverable, the first one from WP4 Experimentation and evaluation, the cases of the pilot studies are first identified and defined, divided into four domains. The first domain for use cases focuses on the set-up, usage and maintenance of Infrastructure Services for Community Clouds (ISCC). The second one looks at Networking Services for Community Clouds (NSCC) that provide network layers for interconnecting cloud services on top. The third domain, about Platform Services for Community Clouds (PSCC), discusses the platform services needed to run applications on top. Finally, the fourth domain is centred on user-oriented Software Services for Community Clouds (SSCC), containing the applications that provide cloud services to community network end users. After defining the use cases, the state of the art of Community Networking Clouds is analysed, focusing on the status of the Guifi.net community. This considered, the performance objectives and their evaluation criteria and metrics are established.

1.2 Terminology

In the context of cloud computing, services are typically divided into three groups: infrastructure, platform and software. Albeit less usual, networking services are commonly considered as another for a total number of four. From the point of view of the providers, the delivery models of their services offer solutions to satisfy the needs on these categories. The four service models considered in this document are commonly referred as Infrastructure as a Service (IaaS), Network as a Service (NaaS),
1.3 Relations of this document with other CLOMMUNITY tasks and deliverables

Platform as a Service (PaaS) and Software as a Service (SaaS). The taxonomy and terminology for the cloud software delivery models are well known and standardised[1] [2].

In all the sections of this document (definition of use cases, state of the art discussion, etc.), however, a slightly different terminology has been used. This has been intentionally done to explicitly state that the subjects of analysis of a particular section are cloud services in the context of CNs, rather than generic cloud services in a more general perspective. Table 1.1 shows the equivalence between generic and cloud-specific terminology.

<table>
<thead>
<tr>
<th>Generic cloud deployments</th>
<th>CN cloud deployments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure as a Service (IaaS)</td>
<td>Infrastructure Services for Community Clouds (ISCC)</td>
</tr>
<tr>
<td>Network as a Service (NaaS)</td>
<td>Networking Services for Community Clouds (NSCC)</td>
</tr>
<tr>
<td>Platform as a Service (PaaS)</td>
<td>Platform Services for Community Clouds (PSCC)</td>
</tr>
<tr>
<td>Software as a Service (SaaS)</td>
<td>Software Services for Community Clouds (SSCC)</td>
</tr>
</tbody>
</table>

Table 1.1: List of terminology equivalences between generic clouds and CN clouds

1.3 Relations of this document with other CLOMMUNITY tasks and deliverables

This deliverable is included in the context of WP4, which receives inputs from the following deliverables:

- **D2.1** System requirements and software architecture (initial)
- **D2.2** System software (year 1)
- **D3.1** Requirements for a holistic abstract network and service architecture
- **D3.2** Experimental research on community clouds (year 1)
- **D3.3** Experimental research on community clouds (year 2)
- **D5.1** IT and knowledge management tool
- **D5.2** Dissemination activities in year 1

This deliverable serves input to the following work packages:

- **WP2** System development
- **WP3** Experimentally-driven research
- **WP5** Promotion and sustainability
2 Definition of use cases

This chapter presents the use cases that have been identified to be of the highest importance to ensure the adoption of cloud services in CNs. They are presented divided into the following four domains: Infrastructure Services for Community Clouds (ISCC), Networking Services for Community Clouds (NSCC), Platform Services for Community Clouds (PSCC) and Software Services for Community Clouds (SSCC). Due to the binding nature of CNs, two additional sections have been added, taking non-technical considerations into account and establishing a common set of requirements, in order to better cope with the CNs’ particularities.

The case selection is the result of the combination of two main criteria, key components of the current cloud solutions and the CNs’ characteristics. Likewise, CNs-specific traits are addressed from the technical point of view (implicitly needed, technical restrictions, etc., frequently resulting into PSCC cases) and from the point of view of useful applications and contents for end users (SSCC).

2.1 Infrastructural services for community clouds

In the ISCC domain two cases have been identified. On one hand, virtual machines as an ISCC are key to any cloud computing environment, and on the other hand, tailored software distributions are a very effective mean\(^1\) to reduce the level of skills and the learning curve needed to install and maintain software applications.

2.1.1 Virtual machines

In the last years, the rapid emergence and evolution of the cloud ecosystem has been brought about – at least in part – by the development and popularisation of virtualisation technologies. By means of software emulation techniques, Virtual Machines (VMs) provide a set of abstraction layers between the hardware and the applications that allow fast, flexible and dynamic deployments of services on top. VMs can emulate anything: Network infrastructure device (a router, a firewall, etc.), full server on which to run any end-user service, dedicated storage service, database server, etc.

A good practice in the use of VMs is to assign a dedicated machine (virtual) to each application or service. At the expense of a slight overhead in resources usage, every service is independent from the other ones, and works transparently to the Infrastructure. This practice eases management control and maintenance, and at the same time can provide better systems security as the VMs operate isolated from each other.

Techniques for deploying VMs can be classified in several different ways, but the most representative criterion is the level of virtualisation. Two different approaches can be distinguished here:

---

\(^1\)Some examples are: 64 Studio, specialised in audio and video production on x86-64 workstations, Emdebian Grip, intended for use on resource-limited embedded systems, Skolelinux, provided as a thin client distribution for schools, etc.
full, hardware-level virtualisation\(^2\) and operating system-level virtualisation\(^1\). In the first category, a complete hardware infrastructure is emulated and provided to the upper software layers, where the operating system of the guest machine is independent of the one in the host. In the second category, the kernel of the host operating system allows for multiple isolated user-space instances, instead of just one. These instances, usually called containers, have the look and feel of a real server from the point of view of the upper layers, but share common infrastructure with the host machine (like the kernel of the operating system).

There are plenty of software solutions that provide full hardware-level, operating system-level or both kinds of virtualization. Among all the available options, free and open source solutions are preferred to proprietary ones (see 2.6 for more details on this). Some of the virtualisation software that can be tested are OpenStack\(^4\), OpenNebula\(^5\) and Proxmox\(^6\) for the hardware-level virtualization approach and LXC\(^7\) and Docker\(^8\) for operating system-level virtualisation. The list is not exhaustive nor exclusionary, since many other virtualisation platforms that meet the requirements exist.

Chapter 3 discusses the organic growth of community networks like Guifi.net, in which the expansion of the network infrastructure goes on par with the involvement of new members and users. Regarding the deployment community cloud services, growth perspectives must be defined in these same terms, so that they scale with their adoption by end users. To achieve this, virtual machines can play an important role, since they can be flexibly and dynamically deployed. This schema also allows non-tech-savvy users to experiment with new services without needing to care for the administration of the underlying hardware infrastructure.

### 2.1.2 A community distribution to provide infrastructural services

When analysing community networks, one of topics discussed is the ratio of users that provide contents or services useful for the rest of the community. Statistics show around 1 in 100 users providing these contents or services\(^9\). One of the reason for such a low ratio can be found in the technical entry barrier. For instance, people willing to provide a certain service in the community network context will need to have, at least, some knowledge about networking and server administration. Therefore, users willing to share any kind of information to the community, must first take care of the technical aspects, then provide the contents.

As an example of this, the release in 2010 of the Guinux distribution generated a lot of interest (in mailing lists and GuifiLabs), specially from people who were afraid before of deploying services for the community because of the needed technical skills. With the apparition of this distribution, the steps for installing and configuring the services was greatly simplified, leading to the proliferation of many of them. Regarding the deployment of cloud services, it can be inferred that every effort put into lowering (or removing) any entry barrier will help users to provide more services and produce more contents. Additionally, the availability of tools to facilitate the deployment of services by people without advanced technical knowledge generates, in turn, an empowerment \([3]\) effect. As a

\(^3\)http://en.wikipedia.org/w/index.php?title=Operating_system%E2%80%93level_virtualization&oldid=601062988
\(^4\)https://www.openstack.org/
\(^5\)http://opennebula.org/
\(^6\)https://www.proxmox.com/
\(^7\)https://linuxcontainers.org/
\(^8\)https://www.docker.io/
\(^9\)http://ww2.grn.es/merce/2008/ramonroca.html
result, a subset of people from inside the community itself appears and takes care of maintaining and improving these tools, ending up in a benefit for the community.

GNU/Linux-based distributions may be installed on very heterogeneous machines, from low-end virtual servers to machines with lots of memory and CPU power. This wide range of hardware characteristics might be seen as a problem, for instance, when defining the minimum required hardware resources. However, a potential feature can be to offer different flavours of the distribution, adapting them to different hardware. In this sense, a standalone version, a container-based one, and a virtual machine can be provided.

- stand-alone version for physical devices (real hardware)
- virtualised version for virtual machines (virtualised hardware)
- isolated version for containers (virtualised operating system)

In order to make the community cloud distribution reach more potential users, it is important to migrate or integrate the tools offered by the old one. This involves that these already available tools will be updated and maintained and the users only interested in them can deploy secure and up-to-date installations. Additionally, user awareness and diffusion will be improved, as these users will be able to activate the cloud services at any given time.

2.2 Networking services for community clouds

Typical cloud deployments (i.e. non community networking-based clouds) are commonly built on top of dedicated, specific networking infrastructures. These networks are designed and operated to fit with the upper running layers of cloud services, accomplishing certain performance and capability requirements that community networks do not provide, or cannot even provide. To name a few, such requirements could be established in terms of bandwidth, latency, reliability, redundancy, but also in terms of addressing and routing policies, traffic shaping, packet prioritisation, multicast traffic support, etc.

Deployments of community clouds are constrained by the underlying community network in terms of resources allocation, performance and capabilities. Since the cloud infrastructure is to be built practically independently from the physical network itself, it might be needed to provide networking as a service tools if upper cloud layers require specific performance conditions. For example, if one of the cloud’s software as a service can not operate adequately when packet losses occur (which is likely to happen in community networking), a possible NSCC demand would be an overlay network that would handle these packet losses and provide a loss-free network to the upper layer.

Another issue with community networking clouds is the fact that the hardware on which the cloud is actually deployed can be spread over different network nodes. These nodes will typically be distributed over different geographic locations and, most probably, on different physical network segments. This imposes some limitations to the cloud services running on top. For instance, services relaying on multicast or broadcast capabilities, like a Service Discovery daemon, would be unaware of the services running on the next node’s cloud.

2.2.1 Overlay network (L2 over L3)

According to the previously analysed case study, in order to overcome the lack of a layer-two network connecting all the devices taking part in the community networking cloud, two options appear. The first one is to limit the existence of community networking clouds to the boundaries of the physical
network connecting the hardware resources between them. While this would be the simplest option, it would either effectively limit the potential growth and scalability or require network administrators to adapt the current network infrastructure to adopt new policies to join different physical network segments. To some extent, VLAN techniques, bridging or other strategies could prove useful. However, they would require the involvement of many actors, which would most possibly end up stopping the deployment of cloud services.

The alternative devised to overcome this limitation is to provide a virtual, overlay layer-two network on top of the IP-based layer-three network. In addition, this overlay network must be created automatically, with minimum user intervention and without requiring any reconfiguration of the community networking infrastructure. Then, on top of this overlay network, cloud services requiring a common physical network, multicast or broadcast functionalities can operate even when located at geographically distant positions.

2.3 Platform services for community clouds

Platform services is the group of software and applications providing basic services and resources to enable the end-user software services on top. Platform services are aimed at offering transparent and uniform procedures to the software services layer, to avoid duplication of functions and simplification of common mechanisms.

An example of a typical platform service are the authentication services. By using the same platform service, software services can provide a common validation method to end users. This way users must manage a single user name and password pair instead of having a pair for each their services.

In the context of CNs, service discovery services, in conjunction with service announcement services, are a key component in the local contents ecosystem and play a similar role of the Internet search engines\textsuperscript{10}.

2.3.1 Service monitoring

A basic platform service in cloud deployments is monitoring. Generally, cloud services are meant to be run unattended for long periods of time (in the months order of magnitude). However, there is always a non-zero probability for a service to stop. The causes for a service to stop can be many, and can be attributed to many factors. To name a few, software bugs or misconfiguration, hardware failures (hard drives, memory, etc.), power shortages, overload, attacks, etc. are among the most common ones. While addressing all the potential causes for downtime is out of the scope of this project, providing platform services such as service monitoring can prove very useful to verify the correct operation of the applications and, when needed, perform any required actions.

We foresee the following types of monitoring services.

- **Network monitoring** Monitoring of network assets such as nodes, links, network interfaces, etc. Metrics: availability, bandwidth used, etc.

- **Servers monitoring** Monitoring of hosts and virtual machines. Metrics: availability, CPU, RAM, storage space, etc.

- **Software monitoring** Monitoring of end-user applications and services. Metrics: tailored tests and metrics.

\textsuperscript{10}Internet search engines do not cope well with CNs because CNs are frequently private networks. Moreover, service discovery/announcement services expected features exceed search engines design functionality.
2.3.2 User authentication

Many end-user applications and services require user authentication service, and many of them allow this service to be covered by an external application. A user authentication platform service is essential for cloud computing in CNs to enable the reuse of credentials by the users. Additionally, if this service is offered by well-known entities that service operators trust, service operators can externalise this service.

2.3.3 Service announcement

One important aspect of community networks is the announcement of services and resources to the community. From the social perspective of the network, good communication practices between service providers, network operators, and end users result in better engagement and usage dynamics. However, this communication is often performed at a close and informal level. For instance, a user may be running a service for self-provision purposes and eventually let other users (i.e., neighbours, relatives, and friends) know about it during an informal meeting, but never announce it openly to all the community. These open, whole-community wide announcements are to be made manually, usually via mailing lists or by registering them on a web page featuring a list of services. This is the case of the Guifi.net community: service providers and users can let other people know about their deployments either writing to the mailing lists, adding them to the Services \(^\text{11}\) section of Guifi.net web page or in more informal meetings (Guifilabs \(^\text{12}\) etc.)

From the technical point of view, two approaches can be followed to facilitate the publication of available services to end users. The first one is the implementation of a service announcement mechanism that automatically publishes available services of a cloud node to all the other nodes. In this sense, it would be a machine-to-machine mechanism rather than a user-targeted one. This way, software services that work in a distributed manner can share their resources or allocate resources on other nodes. For instance, services like distributed storage or voice calling would benefit from that, by facilitating federation of services and clusters.

The second approach is to announce deployed cloud services on user-friendly, searchable repositories, allowing end users to browse them and get information of the available cloud services. This can be done by means of a tool that, after a particular cloud service is deployed, publishes this information to a database or a web page. This tool could also publish the status of the service (in terms of uptime, load, resources usage, etc.) where applicable and, ideally, unpublish it if for some reason the service becomes unavailable.

As a final consideration, public, automatic service announcement should be a mandatory feature of all the cloud services provided the community. The easier the end users can get to know about available cloud services, the higher the user engagement and the services usage will be. Subsequently, this feature should be enabled or disabled at the provider’s will if, for any reason the service is not meant to be published.

\(^{11}\)https://Guifi.net/node/3671/view/services

\(^{12}\)Guifi.net users’ regular meetings where topics of interest such as networking, IT or other aspects of the organisation are discussed.

Deliverable D4.1
2.3.4 Service discovery

Service discovery is closely related to service announcement since they are counterparts to a great extent. In order to link cloud services sharing resources between different nodes (for example, distributed file systems for data storage), tools that automate or reduce manual intervention can prove to be helpful. On one hand, during the cloud services deployment, certain settings and parameters can be automatically configured from the information announced, for instance federation of clouds or services could also be enhanced with these tools. On the other hand, end users can easily locate these cloud services in their vicinities, with real time information regarding the status and availability of resources.

Following a reasoning analogous to the previous subsection 2.3.3, a tool for obtaining information from a repository listing cloud services could be proposed. Strictly, it would not be a service discovery tool, but an information fetching tool. However, the difficulty of developing this piece of software would lay on the fact that the two community clouds announcing services on a common repository could not be directly interconnected (because of the lack of community networking infrastructure linking them). This could be confusing, specially for the end users who would see certain services as available but would be unable to reach them. For these reasons, service discovery tools should focus on discovery on their own rather than on fetching lists of services from repositories.

Contrary to the case of service of the previous subsection 2.3.3, where every cloud service should have the automatic announcement feature, automatic discovery should only find and display the services meant to be public. For instance, a mechanism that exhaustively scans all the cloud nodes for services could, from the technical point of view, finds all the announced services plus the unannounced ones. However, this would most likely be considered too invasive an action. Additionally, nodes not belonging to the cloud deployment could also be scanned and non-cloud services could be mistakenly detected. In short, then, only announced services should be discovered.

2.3.5 Database

Taking into account that database service is a common platform service and that in the specific context of CNs there are already many applications that have one in the back-end, such a service appears to be imperative. Nonetheless, it is worth to mention, already in the case definition stage, that taking into account the diversity of requirements of the applications and of the database types available it will be difficult to find a single solution suitable for all the scenarios.

2.4 Software services for community clouds

Among all the user-oriented services currently available in the Internet cloud, this section discusses three examples that can be imported in the community cloud. The reasons affecting this selection are:

- Adoption rate by end users
- Application area diversity
- Technological diversity
- Technological challenge
- Distribution
- Decentralisation
- Federation
2. Definition of use cases

The first selection criteria are the current and the potential adoption rate of cloud services by end users. When importing these services in the community cloud, they must prove useful for the users. In this sense, if end users are already familiar with a particular cloud service (be it a generic one or branded with a commercial name), they are more likely to adopt the community cloud-based alternative than with ground-breaking, unexplored services.

Regarding the application area of the services, exploring different fields and proposing a limited scope of solutions for each one can raise more interest from end users than offering a wide variety of solutions for very few areas. For example, providing single solutions for services like data storage, video streaming and document edition can cover more users’ needs than providing five different instant messaging solutions. In addition to this, when proposing solutions for a cloud service, using diverse technological approaches lead to more user engagement than offering many similar alternatives. For instance, one solution for a cloud service could focus on data privacy and security, while another could focus on data availability and reliability. At the same time, favouring adoption of innovative technologies that, perhaps, are not yet common on community networking deployments is a way of pushing the limits of such solutions and of the community network itself.

Last but not least, one of the characteristics of community networks is the great distribution over wide geographical areas. From this point of view, solutions that can be distributed around different locations can better fit the underlying community than other solutions unaware of this fact. Very much in this direction, decentralised solutions can prove very valuable in the context of a decentralised networking environment like community clouds. This way, the need for a central server on which the cloud services depend is reduced or avoided at all, giving rise to more flexible solutions. Then, to coordinate this distributed and decentralised cloud services, federation mechanisms can be explored and adopted.

2.4.1 Data storage service

Data storage is one of the many areas on which cloud services operate. In this wide context, many different solutions are available, sometimes focusing on very different aspects of the service. To name a few: data storage cloud services can be used for off-site backup and recovery purposes, for on-line synchronisation from different machines or locations, for collaborative sharing and edition of information. Furthermore, some solutions focus on providing cryptographically secure services, others put their emphasis on offering huge amounts of storage space, etc.

In the context of community cloud, data storage is also needed by end users, and the use cases can be as diverse as with Internet cloud. While trying to address all of them can be too ambitious a goal, providing at least two very differentiated solutions can be a first step for gaining end-user adoption. Additionally, users of one of the solutions can get to know the different approach offered by the other.

The first proposed data storage cloud service is the one that focuses on the ease of use, providing convenient ways for end users to upload, access, manage and share their data. This can be accomplished by means of a web interface, WebDav integration, FTP/SFTP/Rsync access, a client to sync local and remote folders, etc. This solution should allow storing big amounts of data (in the GB order) and should be comparable to the existing free services.

On the other hand, a second proposal for a data storage cloud service could focus on security, by means of the adequate cryptographic techniques. This service could provide end users concerned about moving their contents to a cloud storage warranties about the privacy, integrity, security and

---

13free as in gratis -not as in freedom, or commercial services that offer freemium services or plans
2.5. Non-technical aspects

Community networks are, first and foremost, social projects. Therefore, non-technical aspects play a critical role in the development and maintenance of the network and the community itself. User engagement, awareness of rights, quality of experience or any combination thereof are aspects that must be considered in any action taken or contribution made.

2.5.1 User engagement and participation

Community members are the ultimate reason and the driving force of these projects. Promoting users engagement must be a key factor for use case design, selection and evaluation. Additionally, apart from the natural dissemination activities of the project (discussed in the specific deliverables), to spur the user engagement it is foreseen to contribute hardware (computers, where the software developed and integrated by the project will be installed and tested) to the community.

In addition to these proposed requirements, services that operate in a distributed manner, spreading data and resource usage among different nodes and locations would also be of interest to the end users.

2.4.2 Media streaming service

As with data storage (see 2.4.1), most Internet media streaming services are commonly operated on a cloud basis. However, with few user content producers and many users-consumers, information mostly flows vertically in a top-down fashion.

In a cloud deployed on top of community networking, where the architecture and operation of the network is flattened and follows a more horizontal scheme, it is important to provide end users with tools and services that allow them to provide, share and consume media content easily with other users. Since provision of community cloud-based media streaming services can be addressed in many ways, some general guidelines are suggested to identify the use cases and implement their solutions.

An approach to media streaming services could focus on live streaming of real-time media content. This could be generated on a continuous basis (a video stream obtained from a webcam), or periodically (a scheduled event retransmitted live like, for instance, a radio show) or in a sporadic way (live retransmission of conferences, talks, speeches, social events, etc.). This media streaming service should allow end users to broadcast such content easily, requiring them to focus on just making the media stream available to the cloud, without constraining them by potential issues like network bandwidth problems, computational power for encoding the content to adapt it to other users’ requirements, etc.

As a complement to the previously suggested services, another possible media streaming solution could be to provide on-demand media content. From the community networking point of view, this would only come as a replacement of current Internet cloud-based video-on-demand streaming services. To go one step beyond current media streaming services, features in the user client software like p2p co-streaming, chat, etc. would enhance the service to fit it better into the community networking infrastructures.

2.5 Non-technical aspects

Community networks are, first and foremost, social projects. Therefore, non-technical aspects play a critical role in the development and maintenance of the network and the community itself. User engagement, awareness of rights, quality of experience or any combination thereof are aspects that must be considered in any action taken or contribution made.
2.5.2 Privacy awareness

The participants in the social projects are on the most part more concerned than general populace with regard to the awareness and willingness to commit to new causes. Guifi.net, as a telecommunications project, is not indifferent to privacy and security aspects. CLOMMUNITY project can contribute to strengthen these concerns by offering encryption services as part of its services set. Similarly, CLOMMUNITY project must contribute to highlight the value of local content, and specially, self-managed local contents, as an additional mean to ensure privacy and confidentiality.

2.6 General considerations

Community networks usually define themselves as free, open and neutral. This means, in short, that everybody is able to join, explore, use, enjoy and expand the network with no restrictions other than allowing others to do the same in equal terms. To be coherent with this spirit of openness and transparency, cloud services provided on top of community networks should be based on these same principles.

2.6.1 Free, Libre, Open Source Software (FLOSS)

End users in community networks do not play a mere consumer role, but are an active part of the community's decision making and organisation. Accordingly, in order to allow these users to have full control over the cloud and its services, they must be able to obtain the source code of the software running, analyse it, modify it and redistribute it freely. Using FLOSS means that users are empowered with the aforementioned opportunities to truly control their resources. Obviously this does not mean that all the end users will rush to contribute to the project with their coding, but they will be empowered with the freedom to use, analyse, report and share the software and, eventually, improve it with their code.

Apart from using existing FLOSS in the deployment of community clouds, all the source code developed in the context of the CLOMMUNITY project (e.g. specific applications to announce and discover services) should be made available with the same terms, so that end users have the possibility to see, analyse and modify any software they may receive and use.

2.6.2 Distribution and decentralisation

Regarding deployment and organisation, community networks tend to work in a distributed and decentralised manner, with few centralised features and services. When deploying community cloud services, this model should be observed whenever possible. First, from the technical point of view, decentralised and distributed services can better fit onto and benefit from the underlying network infrastructure. Second, from the social point of view, spreading services and resources among different community networking nodes also gives end users a sense of proximity that turns into a stronger commitment in terms of engagement, participation and usage responsibility.
In community networks, a relevant issue currently observed, in contradiction to their spirit, is that despite the fact that internal traffic is free and Internet access is charged or restricted (bandwidth limitations apply or packets exit through proxies), the predominant trend is to use the available resources only as a means to access external services provided elsewhere in the Internet. Ubiquitous cloud services, such as private data storage and backup, instant messaging, media sharing, social networking, etc. are operated by some well-known Internet services vendors. Community network participants are thus being increasingly affected by the problems and disadvantages of this model (privacy, security, property, legislation, dependency, etc.).

In many cases, Internet cloud services have equivalent alternatives that are owned and operated at the community level, while in other cases there exist no locally-driven alternatives yet. The reasons for the absence of these community-owned services can be found in the difficulty to deploy (find, evaluate, choose, install, operate and maintain) such services and the shortage or lack of individuals, organisations or companies interested in the commercial operation of these services.

To get a dimension of the current situation, we analyse the list of services published (i.e. publicly announced) by the Guifi.net community network. We do so by means of the list of services available on the Guifi.net web page for Catalonia\(^1\) and also for the smaller Barcelonès\(^2\) area inside, both comprising the city of Barcelona.

<table>
<thead>
<tr>
<th>Services</th>
<th>Catalonia</th>
<th>Barcelonès</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy server (Internet access)</td>
<td>275</td>
<td>32,62%</td>
</tr>
<tr>
<td>SNP graphs server</td>
<td>219</td>
<td>25,98%</td>
</tr>
<tr>
<td>DNS server</td>
<td>198</td>
<td>23,49%</td>
</tr>
<tr>
<td>NTP server</td>
<td>96</td>
<td>11,39%</td>
</tr>
<tr>
<td>Bandwidth measurement</td>
<td>36</td>
<td>4,27%</td>
</tr>
<tr>
<td>Internet access for end users</td>
<td>10</td>
<td>1,19%</td>
</tr>
<tr>
<td>Logs server</td>
<td>4</td>
<td>0,48%</td>
</tr>
<tr>
<td>LDAP server</td>
<td>3</td>
<td>0,36%</td>
</tr>
<tr>
<td>Wake on lan</td>
<td>2</td>
<td>0,24%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>843</strong></td>
<td><strong>84</strong></td>
</tr>
</tbody>
</table>

Table 3.1: List of network-related Guifi.net services in Catalonia and Barcelonès areas

---

\(^1\)http://Guifi.net/node/2413/view/services

\(^2\)http://Guifi.net/node/2435/view/services
3. State of the art of Community Networking Clouds

3.1. Infrastructural services for community clouds

<table>
<thead>
<tr>
<th>Services</th>
<th>Catalonia</th>
<th>Barcelonès</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web pages</td>
<td>57</td>
<td>15</td>
</tr>
<tr>
<td>VoIP / audio / video / chat / IM</td>
<td>48</td>
<td>16</td>
</tr>
<tr>
<td>Data storage server</td>
<td>41</td>
<td>7</td>
</tr>
<tr>
<td>Radio / TV stations</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>P2P server</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>Linux mirrors</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>Webcam</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Mail server</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Weather station</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Games server</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>CVS repository</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Server virtualization (VPS)</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>229</strong></td>
<td><strong>56</strong></td>
</tr>
</tbody>
</table>

Table 3.2: List of user-focused Guifi.net services in Catalonia and Barcelonès areas

Comparing tables 3.1 and 3.2, it can be noticed that the services related to the network operation itself outnumber the services intended for end-users. In Catalonia, the ratio of network services compared to user services is 843 to 229, this is almost 3.5 to 1\(^3\). Focusing on end-user services, the most relevant fact is that the number of proxies is higher than the entire set of services aimed at providing content for and by the end users, that is, 275 to 229.

The relationships highlighted in the previous paragraph are also observed in the Barcelonès area. However, this area has particularly very low network degree (ratio of core nodes to end-user nodes) compared to Guifi.net average [4, 5]. These two facts indicate that services are currently offered by the owners core nodes.

This chapter looks at the state-of-the-art of the use cases presented in the previous section. For each use case the current solutions and the lessons learnt are discussed.

3.1 Infrastructural services for community clouds

Two cases have been identified as critical at infrastructure level: virtual machines, a key component of any cloud architecture, and a software framework, in the form of GNU/Linux distribution, specially meant for non-technical people, to ease and harmonise the process of integrating new resources to the cloud.

\(^3\)Frequently the same machine runs more than one service (the proxy+SNMP+DNS pack is a typical case, as discussed later on in section 3.1.2)
3.1. Virtual machines

Following the global trend, popularity of virtualisation techniques is increasing in Guifi.net. To the best of our knowledge, after overcoming the initial resistance, and once the efficiency benefits have been proved, currently almost all critical services are run on virtualised environments.

The common solution is Proxmox\(^4\). Proxmox is a GNU/Linux virtualisation distribution based on two different hypervisor technologies: KVM\(^5\) and OpenVZ\(^6\). The first one is used to provision hardware-level virtualisation, the second to provide Linux container virtualisation. Proxmox is based on the latest Debian Linux distribution and uses the main Debian public repositories to keep the operating system up to date. The web based GUI is used to administer Proxmox and gives the admin the ability to create new virtual machines and manage storage.

The Guifi.net Foundation operates an 8-nodes 3-areas Proxmox cluster. Proxmox is also used by UPC in the context of the CONFINE project\(^7\) [6] and by several Guifi.net’s ISPs. Nonetheless, the clusters are operated independently and we are not aware of any coordination protocol or effort.

Proxmox is developed and maintained by Proxmox Server Solutions. It is available mainly in the forms of a free, limited edition and an enterprise edition with full capability and support. Community projects should be aware of the risks of depending on a product offered by a private company in favour of community projects whenever possible, especially when the company is closing the software gradually, as Proxmox Server Solutions is doing.

3.1.2 Legacy Guifi.net distribution (Guinux)

In the context of the Guifi.net environment, some efforts had been made in the past to provide the end users with tools to help them with the deployment and expansion of the community network from the software and services point of view. For instance, in April 2010, Guinux\(^8\) (a GNU/Linux distribution based on Fedora\(^9\) 12\(^10\)) was made available to end users so that they could deploy servers with services useful for community networking. The distribution was shipped with three scripts that could be run from the desktop (see Fig. 3.1\(^11\)) and allowed quasi-automatic configuration of the most important network services (proxy, DNS and SNMP graphs servers). After this, development of the Guinux distribution was discontinued for more than one year. On October 2011, a new release of Guinux was published\(^12\), this time based on Ubuntu\(^13\) 11.04\(^14\). Finally, in March 2012 a repository\(^15\) for Debian-based systems was published\(^16\), containing the server configuration scripts and other Guifi-related tools.

Favouring the repository-based publication of Guifi-related network software tools provides some

\(^4\)https://www.proxmox.com/  
\(^5\)http://www.linux-kvm.org  
\(^6\)http://openvz.org  
\(^7\)http://confine-project.eu/  
\(^8\)http://Guifi.net/node/29320  
\(^9\)http://fedoraproject.org  
\(^10\)https://fedoraproject.org/wiki/Releases/12/Schedule  
\(^11\)FC12 screenshots by Ramon Roca, published under Creative Commons license on http://Guifi.net/node/29320  
\(^12\)https://lists.Guifi.net/pipermail/Guifi-rdes/2011-October/021545.html  
\(^13\)http://www.ubuntu.com  
\(^14\)http://old-releases.ubuntu.com/releases/11.04/  
\(^15\)http://serveis.Guifi.net/debianGuifi/  
\(^16\)http://ca.wiki.Guifi.net/wiki/Configurar_Repositori_apt_Guifi
advantages in comparison to providing a full-blown distribution. First, the repository can be added to a host with an operating system already installed, without needing to perform a complete operating system installation from scratch. This means too that the system can be kept to a minimum number of packages, avoiding those that are not needed to operate network-related services, which were installed by default with the full-blown Guinux distribution (the desktop environment, an e-mail client or a pictures viewer, to name a few). Second, from the technical workload point of view, maintenance and updating of a repository is much simpler than including the tools shipped in a complete, up-to-date distribution. This way only minor changes, if any, need to be done when new releases of the distributions appear.

However, making community networking tools available only via a repository implies several drawbacks, specially from the social point of view. The most remarkable one is the entry barrier that non-technically skilled users find. While installing an operating system by following a click-by-click process is within the reach of most Guifi.net users, adding custom repositories to the system and manually configuring the services is not such a straightforward task. This means that non-advanced users no longer have the option of deploying a system that works out-of-the-box, knowing what this system...
3.2 Networking services for community clouds

Guifi.net is an IPv4 RFC1918\(^{17}\) routed network. In this network, tunnelling techniques are used to deliver Internet IPs. The Guifi.net Foundation is a RIPE-NCC\(^{18}\) member. It is operating two IPv4 ranges, /21 and a /24, and one IPv6 range, /32, and has peer agreements with 22 other ISPs. These IPs blocks are subdivided by the Guifi.net Foundation and are allocated to the ISPs operating inside Guifi.net. ISPs are free to use the tunnel technologies they prefer. The most common ones are PPPoE, IP-IP, OpenVPN, etc. Tunnels are usually managed by specific hardware, i.e. multicore routers.

There are some groups within the network using tunnels for different purposes:

The most common:

- **Virtual Private Networks (VPNs):** As a service among people in Guifi.net
- **Quick Mesh Project (qMp):** Tunnels IP-IP in BatMan-eXperimental version 6 (BMX6), because loops may get created loops when accessing outside the qMp network.
- **Confine:** Uses tinc\(^{19}\) as a management network
- **Ethernet over Internet Protocol (EoIP):** In some cases create tunnels at Ethernet level, for example RouterOS\(^{20}\) has this feature enabled by default.

3.2.1 Overlay network (L2 over L3)

To the best of our knowledge there are no significant implementation of this service in Guifi.net.

3.3 Platform services for community clouds

Platform services for CNs can be classified into network services and end-user services. Network services include all those services aimed at easing the design, expansion, management and monitoring, etc. of the network. End-user services include the services aimed at improving the User eXperience (UX).

Guifi.net network services have become a reference for CNs because such tools are an essential part of the plan to achieve the complete self-service model of Guifi.net. To this end, and in addition to the ones described in the following subsections,\(^{21}\) **unsoclic** must be mentioned. **unsoclic** is a tool integrated as part of the Guifi.net website that provides the complete configuration of network nodes.

---


\(^{18}\) [http://www.ripe.net/](http://www.ripe.net/)

\(^{19}\) [http://www.tinc-vpn.org](http://www.tinc-vpn.org)


\(^{21}\) **unsoclic** means a single click spelt altogether in Catalan language.
Almost all CNs have developed their own network services framework. Some of the most relevant are: Wlansolvenia\(^{22}\) (Nodewatcher\(^{23}\)), AWMN\(^{24}\), FunkFeuer\(^{25}\), Free Network Foundation\(^{26}\).

Nonetheless, end-user services are just partially successfully resolved in CNs. For instance, in Guifi.net, the user authentication solution, based on Lightweight Directory Access Protocol (LDAP) is fairly solid and can be reused almost as it is in CLOMMUNITY. On the contrary, services announcement service, a solution that hasn’t almost evolved since the beginning, due to the number of services listed has become almost useless.

### 3.3.1 Service monitoring

**Network monitoring** SNPservices\(^{27}\) is an advanced network monitoring tool developed by Guifi.net. It is a PHP\(^{28}\) client-server application that uses RRDTool\(^{29}\) for data logging and graphing system. It is distributed as a software package available at Guifi.net repository\(^{30}\). As shown in Table 3.1, there are more than 200 monitoring servers (graph servers) in Guifi.net which monitor almost the whole network. The data from these servers is integrated in the Guifi.net WEB site. The service can be integrated in CLOMMUNITY almost as it is.

**Servers monitoring** Server monitoring is already included as part of the SNPservices. With minor modifications, it can be extended to monitor VMs.

**Software monitoring** To the best of our knowledge there is no significant implementation of this service in Guifi.net, probably due to the amount of tailored work required for each service to be monitored. We foresee that the modular design will be the most appropriate for this.

### 3.3.2 User authentication

Currently the Guifi.net Foundation operates and maintains a centralised authentication service based on LDAP. This service is used by the Guifi.net and the social.Guifi.net websites, by the federated proxies \(^{31}\), etc. The service is run in four servers, using one as master and three as slaves. High availability and load balancing is achieved by DNS and HAproxy\(^{32}\). The service is also available as OpenID\(^{33}\).

The current Guifi.net’s solution seems suitable for CLOMMUNITY as it is.

---

\(^{22}\)https://wlan-si.net/
\(^{23}\)https://dev.wlan-si.net/wiki/Nodewatcher
\(^{24}\)http://www.awmn.net
\(^{25}\)http://www.funkfeuer.at/
\(^{26}\)http://thefnf.org/
\(^{27}\)http://es.wiki.Guifi.net/wiki/SNPservices
\(^{28}\)http://www.php.net/
\(^{29}\)http://oss.oetiker.ch/rrdtool/
\(^{30}\)http://serveis.Guifi.net/debian/Guifi/
\(^{31}\)In Guifi.net there are about 300 federated web proxies. Federated, in terms of credentials, means that once a user name and password pair is approved in one of the proxies, it is recognised as valid by the rest. User approval rules are the same for all the federated proxies.
\(^{32}\)http://haproxy.1wt.eu/
\(^{33}\)http://openid.net/
3.3.3 Service announcement

Much like in the software development paradigm described in The Cathedral and the Bazaar[7], the Guifi.net community network could be identified as a small “bazaar” that grew organically to service the community, and Guifi.net website[34] that exposes new services can be termed as “bazaar location”. The problem with this is that many of organically growing the services are not ordered and in many cases are outdated. Moreover no one (person or machine) that constantly monitors the services can causing wrong information or does not upgraded. And this situation can do Community does not trust web information. And when you get to this point the community can stop using the system.

3.3.4 Service discovery

To the best of our knowledge there is any significant implementation of this service in Guifi.net.

3.3.5 Database

Apart from the aforementioned LDAP database (Subsection 3.3.2), other databases are also used in Guifi.net. A non-exhaustive list follows.

- **Network description (“CNML” database)** One of the key components of Guifi.net framework back-end is the database that contains the description of the whole network (nodes, links, distances, devices, neighbours, etc.). This information is publicly available in Community Network Mark Up Language (CNML) format[35]. It is a MySQL database.
- **Guifi.net website back-end** Guifi.net website is Drupal[36] based that uses a MySQL database to store all the contents.
- **”glir” database** A MySQL database used by a GLPI[37] based website, the Network Operations Center (NOC) website.
- **“glir” tree** A LDAP tree, complementary to the Guifi.net users’ one, containing the profile of NOC members.

The following list contains well-know databases of other CNs.

- **WindDB** The AWMN network database.
- **Redeemer** The Funkfeuer network database.
- **NodeDB** A general CNs network database developed by FunkFeuer.

3.4 User-oriented services for community clouds

3.4.1 Data storage service

In the context of community networks, data storage services have been sporadically deployed by enthusiastic users who wanted to share some of their contents (pictures, documents, etc.) with the rest of the community. Typically the contents have been accessible from an FTP server or, sometimes, via

---

[34]http://Guifi.net/en/node/3671/view/services
[36]https://drupal.org/
[37]http://www.glpi-project.org/
HTTP. In some cases, users have also enabled uploading to folders to let other users put their files for share with the community. Despite this, it should not be considered a data storage service for end users, specially in the terms described in 2.4.1.

Leaving apart the cases of the previous paragraph, very few data storage services are known to be running in the Guifi.net community network. According to the list of available services in the web page\[38\], only a couple of ownCloud\[39\] setups are published. It is informally known that some users have deployed ownCloud, Seafile\[40\] and similar storage services for private usage (family, local groups of users, work groups, etc.).

### 3.4.2 Media streaming service

Guifi.net initial streaming experiences dates back to 2009 when a community member (Eloi Rebes\[41\]) started recording most of the dissemination talks and workshops and uploading them at Blip.tv\[42\]. Sometime later other members also started uploading their videos.

In 2010, Eloi started the GuifiTV\[43\] project to harmonise the video formats and the contents. Also that year he made the first live stream with great expectation using Flumotion\[44\] (FLOSS). Since then Eloi has successfully live-streamed and recorded many of the Guifi.net events such as SAXs\[45\], WBMv4\[46\], IS4CWN2012\[47\], etc. using Flash Media Server\[48\] (proprietary software).

Thanks to Eloi, video streaming has become a very popular service in Guifi.net. All the videos recorded (109 as of April 2014) are available at Guifimedia\[49\]. COMMUNITY must definitely benefit from his experience and expertise. COMMUNITY can make a great contribution by bringing a FLOSS alternative to the proprietary solution currently used.

### 3.5 Non-technical aspects

CNs are characterised by attracting very active and committed members, frequently with broad experience in other collaborative projects such as FLOSS, cooperatives of all kinds, etc. Thus, social activities such as face to face meetings, workshops, debates on no strictly networking topics, etc. are part of the common CNs activities. Guifi.net not only is no exception to this fact but it has become a reference for the rest of CNs in many of this social and organisational aspects.
3.5.1 User engagement and participation

CNs are participative and inclusive by nature because their main goal is to build and maintain assets by the people and frequently members participation extends far beyond what is absolutely necessary to achieve this goal.

CNs use any kind of Information technology (IT) available, such as mailing lists, forums, etc. to share knowledge and for action coordination. Just as an example, in Guifi.net there are more than 50 active mailing lists with a total of hundreds of emails per day.

Moreover, face to face self-organised events are rather common. Almost each CN has at least an annual gathering event. For instance Guifi.net organises the SAX, FreiFunk the Wireless Community Weekend, Ninux the Ninux day, etc.. In addition, inter-CN events are becoming increasingly popular, being the Wireless Battle Mesh and the International Summit for Community Wireless Networks the most well-known.

CLOMMUNITY must benefit from the already existing social networks of CNs to disseminate its results and by doing so it will also contribute to strengthen these networks.

3.5.2 Privacy awareness

As part of security related topics, privacy is a very sensitive topic in CNs circles and it is a recurrent topic at the discussion forums. Unfortunately the related actions are to often restricted to these debates meaning that, in practice, the people do not take any additional measure. In Guifi.net we certainly know that credentials are shared by plain e-mail and sensible data such as bank accounts is stored in unencrypted files, for instance.

We are convinced that by providing some read-to-use services which implicit guarantee privacy CLOMMUNITY can effectively contribute to put all these concerns in practice and to instill the habit of using cryptography tools.

3.6 General considerations

As already said, CNs social implication and general awareness is clearly above average. Nonetheless, some contradictions between what is preached and the daily activity can still be found. As in the case of privacy and security, already discussed in subsection 3.5.2, CLOMMUNITY must contribute to ameliorate this contradictions by practical examples

3.6.1 Free, Libre, Open Source Software (FLOSS)

FLOSS is the perfect example of a case where, despite of being outside of the scope of CNs domain, one might expect full commitment to it. That is to say, from its principles to to the last practical case, from, at least, their core members. Nonetheless, not even in the theoretical plane all of these members are, to date, yet convinced of the necessity of FLOSS to be coherent with CN principles. Moreover, the more closer we get to the practical field the more the dominance of proprietary software increases. Even amongst the most convinced members is possible to find “exceptional” cases where the usage of proprietary software is justified, either because “unfortunately, there isn’t any real FLOSS alternative”

---

50 All CN are intrinsically self-organised.
or because “bah, just for this case”. In Guifi.net case on the one hand all developed software (thou-
sands of lines of code) is FLOSS, but on the other hand most of the nodes run proprietary firmwares. Fortunately, all the servers run GNU/Linux.

We are convinced that openness evangelism, FLOSS included, is a matter of insistence and persis-
tence. Thus CLOMMUNITY must only integrate FLOSS solutions and all its developed code must be made public available under a FLOSS license.

3.6.2 Distribution and decentralisation

Distribution and decentralisation as a general way of doing things is yet another case where there is a contradiction between what is expected and wished and how eventually things are really implemented. Indeed, distribution and decentralisation are concepts closely related to CNs philosophy, nonetheless, since centralised solutions are generally much easier to develop and implemented, most of the cases end up being implemented in according to the classical client-server approach.

Here again, CLOMMUNITY is a great opportunity to help reversing the situation. Thus, special efforts must be made to go beyond the “easy” client-server solutions.
4 Performance objectives

When comparing the current situation with what we envisioned the project’s outcome to be, a set of objectives arise to further develop and improve the services. Some of these objectives fall into the development of the tools and services, adding features and improving existing ones, while other objectives would fall into the UX and engagement field.

In this chapter the performance objectives are identified for each case identified in Chapter 2 following its sections structure together with an additional section, Section 4.6, that gathers those objectives that apply to most, if not all, cases and areas. For instance, a high level of automation and extensive unit tests throughout the software would be desired in most of the cases presented below.

4.1 Infrastructural services for community clouds

<table>
<thead>
<tr>
<th>Case</th>
<th>Performance Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>• Significant number of hosts</td>
</tr>
<tr>
<td></td>
<td>• Automatic testbed configuration (slice -in Plantelab terminology)</td>
</tr>
<tr>
<td></td>
<td>• Foster social interaction among VMs providers and users</td>
</tr>
<tr>
<td>Distro</td>
<td>• Become (again) the reference</td>
</tr>
<tr>
<td></td>
<td>• Good UX (WEB base GUI)</td>
</tr>
</tbody>
</table>

Table 4.1: Performance objectives of ISCC cases.

4.2 Networking services for community clouds

<table>
<thead>
<tr>
<th>Case</th>
<th>Performance Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2-L3</td>
<td>• Self-discovery</td>
</tr>
<tr>
<td></td>
<td>• Restricted to management traffic</td>
</tr>
<tr>
<td></td>
<td>• Low traffic</td>
</tr>
</tbody>
</table>

Table 4.2: Performance objectives of NSCC cases.

4.3 Platform services for community clouds
4. Performance objectives

4.4 Software services for community clouds

<table>
<thead>
<tr>
<th>Case</th>
<th>Performance Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Monitoring</td>
<td>• ISCC monitoring (nodes and links)</td>
</tr>
<tr>
<td></td>
<td>• Monitoring framework for PSCC and SSCC (modular design, service tailored)</td>
</tr>
<tr>
<td>User Authentication</td>
<td>• Standalone service</td>
</tr>
<tr>
<td></td>
<td>• Offered by a third-party with a good reputation</td>
</tr>
<tr>
<td></td>
<td>• Technically secure</td>
</tr>
<tr>
<td></td>
<td>• Basic credentials</td>
</tr>
<tr>
<td></td>
<td>• Free-Schema</td>
</tr>
<tr>
<td></td>
<td>• High Availability (HA)</td>
</tr>
<tr>
<td>Service Announcement</td>
<td>• Develop a friendly model and promote it</td>
</tr>
<tr>
<td></td>
<td>• Low transit</td>
</tr>
<tr>
<td>Service Discovery</td>
<td>• Recognise announced services</td>
</tr>
<tr>
<td></td>
<td>• Publication of discovered services</td>
</tr>
<tr>
<td>DDBB</td>
<td>• Service available</td>
</tr>
</tbody>
</table>

Table 4.3: Performance objectives of PSCC cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Performance Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data storage</td>
<td>• Privacy (as an option)</td>
</tr>
<tr>
<td></td>
<td>• Redundancy</td>
</tr>
<tr>
<td>Media streaming</td>
<td>• Video on demand</td>
</tr>
<tr>
<td></td>
<td>• Video streaming</td>
</tr>
</tbody>
</table>

Table 4.4: Performance objectives of SSCC cases.

4.5 Non-technical aspects

<table>
<thead>
<tr>
<th>Case</th>
<th>Performance Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Engagement</td>
<td>• Dissemination in CNs forums</td>
</tr>
<tr>
<td></td>
<td>• Dissemination in academic and industrial forums</td>
</tr>
<tr>
<td></td>
<td>• Adoption by third parties</td>
</tr>
<tr>
<td>Awareness</td>
<td>• Dissemination in CNs forums</td>
</tr>
<tr>
<td></td>
<td>• Promotion through practical examples</td>
</tr>
</tbody>
</table>

Table 4.5: Performance objectives of Non-technical cases.

4.6 General considerations

Deliverable D4.1
### 4.6. General considerations

#### 4. Performance objectives

<table>
<thead>
<tr>
<th>Case</th>
<th>Performance Objectives</th>
</tr>
</thead>
</table>
| All  | • Suitable for CNs (physical traits, social traits, etc.)  
|      | • Tested  
|      | • Decentralisation  
|      | • High automation  
|      | • High flexibility  
|      | • Scalable  
|      | • Low overhead  
|      | • Improvement of existing solutions  
|      | • Members as consumers and providers  
|      | • "Open" (FLOSS, available to all members, etc.) |

**Table 4.6:** Common performance objectives of all cases.
5 Evaluation criteria and metrics

In this chapter we elaborate the metrics and the evaluation criteria to achieve the performance objectives set in Chapter 4 keeping its sections and subsections structure. Metrics refer to total amounts, unless otherwise stated.

5.1 Infrastructural services for community clouds

<table>
<thead>
<tr>
<th>Case</th>
<th>Metric</th>
<th>Evaluation criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>No. of CLOMUNITY-project contributed hosts</td>
<td>5/10/100</td>
</tr>
<tr>
<td>VM</td>
<td>No. of third-party contributed hosts</td>
<td>5/10/100</td>
</tr>
<tr>
<td>VM</td>
<td>No. of third-parties participating</td>
<td>3/5/10</td>
</tr>
<tr>
<td>VM</td>
<td>No. of experiments</td>
<td>5/10/20</td>
</tr>
<tr>
<td>VM</td>
<td>No. of VMs per experiment</td>
<td>3/5/10</td>
</tr>
<tr>
<td>Distro</td>
<td>No. of installations</td>
<td>10/25/80</td>
</tr>
<tr>
<td>Distro</td>
<td>Installation and management Graphical User Interface (GUI)</td>
<td>Yes</td>
</tr>
<tr>
<td>Distro</td>
<td>UX (Satisfaction survey)</td>
<td>Positive</td>
</tr>
<tr>
<td>Distro</td>
<td>No. of microclouds</td>
<td>2/4/10</td>
</tr>
<tr>
<td>Distro</td>
<td>Average no. of nodes per microcloud</td>
<td>3/5/50</td>
</tr>
</tbody>
</table>

Table 5.1: Metrics and evaluation of ISCC cases.

5.2 Networking services for community clouds

<table>
<thead>
<tr>
<th>Case</th>
<th>Metric</th>
<th>Evaluation criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2-L3</td>
<td>Self-discovery</td>
<td>Yes</td>
</tr>
<tr>
<td>L2-L3</td>
<td>Restricted to management traffic</td>
<td>Yes</td>
</tr>
<tr>
<td>L2-L3</td>
<td>Average Mbs of traffic of middle-size cloud server</td>
<td>0.5/1</td>
</tr>
</tbody>
</table>

Table 5.2: Metrics and evaluation of NSCC cases.

5.3 Platform services for community clouds
### 5.4 Software services for community clouds

#### Table 5.3: Metrics and evaluation of PSCC cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Metric</th>
<th>Evaluation criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Monitoring</td>
<td>Interfaces Traffic</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Nodes RTT</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Amount of supported servers</td>
<td>3/5/10</td>
</tr>
<tr>
<td></td>
<td>Modular design (to allow future additions)</td>
<td>Yes</td>
</tr>
<tr>
<td>User Authentication</td>
<td>Autonomous service</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>No. of services using it</td>
<td>3/5/10</td>
</tr>
<tr>
<td></td>
<td>Average queries (daily)</td>
<td>100/1000/10000</td>
</tr>
<tr>
<td></td>
<td>Basic credentials</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Free-Schema</td>
<td>Desirable</td>
</tr>
<tr>
<td></td>
<td>HA</td>
<td>1/1/2</td>
</tr>
<tr>
<td>Service Announcement</td>
<td>Service implemented</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Modular design</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>No. of services using it</td>
<td>3/5/10</td>
</tr>
<tr>
<td>Service Discovery</td>
<td>Recognition of announced services</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Publication of discovered services</td>
<td>Yes</td>
</tr>
<tr>
<td>DDBB</td>
<td>No. of services using PSCC DDBB</td>
<td>1/2/5</td>
</tr>
</tbody>
</table>

#### Table 5.4: Metrics and evaluation of SSCC cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Metric</th>
<th>Evaluation criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data storage</td>
<td>Available storage space [GB]</td>
<td>100/500/10000</td>
</tr>
<tr>
<td></td>
<td>Used storage space [% of available]</td>
<td>5/10/50</td>
</tr>
<tr>
<td></td>
<td>Average traffic [GB/month]</td>
<td>10/50/1000</td>
</tr>
<tr>
<td></td>
<td>Average no. of active users (monthly)</td>
<td>50/100/1000</td>
</tr>
<tr>
<td>Video on demand</td>
<td>No. of files available</td>
<td>50/100/1000</td>
</tr>
<tr>
<td></td>
<td>Average no. of requests per available file (monthly)</td>
<td>2/10/50</td>
</tr>
<tr>
<td></td>
<td>Average no. of consumers (monthly)</td>
<td>10/100/5000</td>
</tr>
<tr>
<td></td>
<td>Average no. of new contents providers (monthly)</td>
<td>2/10/50</td>
</tr>
<tr>
<td>Video streaming</td>
<td>No. of channels available</td>
<td>5/10/100</td>
</tr>
<tr>
<td></td>
<td>No. of requests per channels available (monthly)</td>
<td>5/10/25</td>
</tr>
<tr>
<td></td>
<td>Average no. of consumers (monthly)</td>
<td>2/10/250</td>
</tr>
<tr>
<td></td>
<td>Average no. of new contents providers (monthly)</td>
<td>5/10/25</td>
</tr>
</tbody>
</table>

#### 5.5 Non-technical aspects
Table 5.5: Metrics and evaluation of non-technical cases.

### 5.6 General considerations

The performance objectives identified in Section 4.6 apply to all cases.
6 Conclusions

T4.1 defined the cases of the pilot studies and their performance objectives, evaluation criteria and metrics, bringing inputs to the rest of WP4 tasks as well as to the rest of WPs.

This document gives the outcomes of T4.1 devoting a chapter to the case definition, another to report on the existing experiences, a third one to identify the performance objectives and a last one to define the metrics and to set the evaluation criteria. The cases were identified and developed in the context of the following cloud for CNs domains: IaaS, NaaS, PaaS and SaaS, devoting a section for each of them in each chapter. In order to better cope with the binding nature of these communities a section to attend the non-technical considerations was added to each chapter together with another developing the common aspects.

Additionally, as part of the efforts to grasp the essence of CNs, the domains were renamed, IaaS to Infrastructure Services for Community Clouds (ISCC), NaaS to Networking Services for Community Clouds (NSCC), PaaS to Platform Services for Community Clouds (PSCC) and PaaS to Platform Services for Community Clouds (PSCC).
Acronyms

ASP  Application Service Provider
BMX6  BatMan-eXperimental version 6
CN  Community Network
CNML  Community Network Mark Up Language
DDBB  Database
DNS  Domain Name System
EoIP  Ethernet over Internet Protocol
FLOSS  Free, Libre, Open Source Software
GUI  Graphical User Interface
HA  High Availability
IaaS  Infrastructure as a Service
ISCC  Infrastructure Services for Community Clouds (Community specific)
ISP  Internet Service Provider
IT  Information technology
LDAP  Lightweight Directory Access Protocol
NSCC  Networking Services for Community Clouds (Community specific)
NaaS  Network as a Service
NOC  Network Operations Center
PaaS  Platform as a Service
PSCC  Platform Services for Community Clouds (Community specific)
qMp  Quick Mesh Project
RTT  Round Trip Time
SaaS  Software as a Service
SME  small and medium enterprise
SSCC  Software Services for Community Clouds (Community specific)
T  Task (EC projects specific -followed by task ID)
UX  User eXperience
VM  Virtual Machine
VPN  Virtual Private Network
WP  Work Package (EC projects specific)
VoIP  Voice over IP
Bibliography

Licence

The CLOMMUNITY project, April 2014, CLOMMUNITY-201404-D4.1:

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 Unported License.