D.4.1 – Application scenarii and Design of infrastructure

Yann Busnel

D.4.1 – Application scenarii and Design of infrastructure

SOCIOPLUG members

Email contact: Yann.Busnel@ensai.fr

ANR-13-INFR-0003

socioplug.univ-nantes.fr

SocioPlug is a research project, funded by French National Agency for Research, which aims to investigate on Social Cloud over Plug Networks, Enabling Symmetric Access to Data and Preserving Privacy.

In this project, we will perform both theoretical and practical evaluation of the solutions proposed in the all work packages. Task 4 (Infrastructure and Experimentation) will be structured around use-cases where partners developed previous expertise such as distributed collaborative systems, social web and Smart Building. These well known distributed systems will be revisited to fit federation of plug constraints using results of other tasks.

In this deliverable, we present some details of the infrastructure of the federation of plug computers, that will be developed in this project. We plan to provide a demonstrator and deploy on it some application according to the use-cases presented in this deliverable.

Keywords: Infrastructure, Use-cases, Complementarity of tasks

1 Introduction

SOCIOPLUG aims to leverage interconnected plug computers to offer an alternative approach for the implementation of personal cloud services. The resulting "social cloud" will address privacy issues by storing data on personal plugs under the control of end-users, while enabling symmetric valorization activities over users' data. In particular, SOCIOPLUG will seek to develop models for securing, programming and querying such a plug-based social cloud, by exploring the three complementary axes of data production, data organization, and data protection.

In this context, we selected complementary use-cases highlighting different issues of SOCIOPLUG and on which each partner has a strong expertise. In Section 2, we address an alternative to "Google Now" service. This use-case demonstrates how personalized suggestions can be obtained from a federation of personal information systems. In Section 3, we focus on a "Google Doc" alternative. This use-case demonstrates how we can achieve massive collaborative editing while maintaining privacy of groups. In Section 4, we demonstrate how plugs infrastructure can be turned on collaborative sensor network for smart cities without sacrificing users privacy. Finally, in Section 5, we focus on an alternative to "Google Drive" services. This use-case illustrates how it is possible to share contents with no central provider while preserving privacy. All these use-cases address fundamental scientific challenges SOCIOPLUG: consistency and programming federation of plugs, social query evaluation and valorization, data privacy and safety.

For evaluation purpose, we installed in Nantes a farm of currently 50 plugs that will grow up to 200 plugs in June 2015. This farm can be easily accessed by any partner of SOCIOPLUG and configured to fit requirement of different use-cases. The farm is already operational, has been tested for preliminary experiments for scientific publication, teaching and also by GDD and IVC research teams for distributed file system experimentations.

2 Scenario 1: PlugTrack & Google Now

Preamble Personalized and recommendation services have become a fundamental component of today's world-wide web. Made famous by companies like Amazon or Netflix in the context of e-commerce services, they have found applications in social networks, news websites [Das07], and even discussion forums [ABH08].

Current personalization systems operate following a data-concentration model. They centralize data about users and items in order to mine correlations and provide users with pertinent suggestions. While effective and simple from a data-mining perspective, concentrating user information either limits the scope of personalization and recommendation to non-sensitive data or it introduces important privacy concerns.

Consider the example of Google Now [Goo14c], Google's personalized information service that provides users with customized suggestions and information. The recommendation cards displayed by Google Now rely on information about the browsing history of users, their email activity, their calendars, their commutes, their travel information, their phone calls, and whatever other personal information Google has been able to harvest about the user.

While clearly offering useful services to users, applications like Google Now therefore constitute a significant threat to the privacy of users. While users may always opt out of such personalized services, with this scenario, we show that Google-Now-like applications may operate without disclosing information to a centralized player who may act as potential Big Brother.

Objectives and challenges This scenario will demonstrate the effectiveness of the SOCIOPLUG architecture in supporting personalized applications. Let us consider John, a prototypical user, who lives near Liffré, a small town north-east of Rennes. John works as a system administrator in the center of Rennes and drives to work every day for two reasons. First, his schedule does not correspond to that of the few bus lines that connect the towns of Liffré and Rennes. Second, his house is a 5 minute drive from the bus stop in Liffré, while the highway that connects the two cities is just one minute away. The city of Liffré offers a web service that allows people to advertise their routes and find car-sharing partners. John was initially enthusiastic about it, but he quickly got discouraged. His schedule varies too much not only from one day to the next but also from week to week. As a result, John never managed to car share and always drives his car to work.

John does not know it, but he's not the only one in this situation. Although he does not have any colleagues that live in his area, the center of Rennes hosts a large number of working people that live near Liffré and that have irregular work schedules. Like John, a lot of this people work either in the IT industry or have a technical job. As a result, they have been among the first to install a plug computer with the SOCIOPLUG software in their homes as well as the new mobile PlugTrack application.

PlugTrack collects location information from a mobile device (phone or table) in pretty much the same way as built-in Google or Apple applications. However, instead of sending this information to Google or other third parties, they upload the information to the user's own plug computer.

The plug computer uses location information in several ways. Similar to services like Google Now [Goo14c], or FourSquare [Fou14], it computes personalized recommendations for users based on their current location. In the case of John, recommendations will comprise bars and restaurants where to eat while at work, or activities to do in the evening near his home. These recommendations rely not only on the data collected by John's own plug computer, but they combine this information with that of other SOCIOPLUG users. This has two advantages. First, it makes it possible to obtain better recommendations than by only exploiting one user's information. Second, it enables recommendation that would be impossible without accessing data about other users.

In particular, it may enable John to leverage the presence of other users that share part of his home-work route and that he does not even know about. Shortly after installing the PlugTrack mobile application, John receives notifications about the presence of other users who travel along the same route at very similar times. Each user, including John, records his/her commuting plans on a weekly or even daily basis, and SOCIOPLUG matches users with each other in order to minimize the number of employed cars.

This kind of recommendation also proves useful to Philippe. Unlike John, he has been driving to work because he's convinced that public transports would make him waste time. Indeed this was the case a long

time ago, but the transport service has since been renewed. Fortunately, SOCIOPLUG can detect that some users travel to destinations close to where Philippe works, at compatible times, and with little or no delay with respect to a car. The recommendation application therefore suggests Philippe to explore these people's mode of transportation.

Task interaction In this scenario, we showcase an approach to personalized services relying on SOCIOPLUG's decentralized architecture. Instead of collecting all information about users in a single server (or set of servers), SOCIOPLUG leaves user information on the plug computers made available by users and exploits decentralized epidemic protocols to identify meaningful correlations between users and items.

Clearly, decentralization removes the threat of a big brother that can access information about all users, but it also introduces new potential threats coming from users that may want to spy on other users. To address this challenge, this scenario will complement the contributions of Task 1 with those of Task 3.

3 Scenario 2: Collaborative editing & Google Doc

Preamble Collaborative editors allow people to work distributed in space, time and across organizations. GoogleDoc, EtherPad, ShareLatex are now commonly used by millions of people. However, most of these editors requires a centralized component. this raises issues concerning fault tolerance with a single point of failure, privacy, economic intelligence and some limitations in term of service. For the economic intelligence issue, it is easy for a dominant collaboration provider to know who work with who, when and on which topic. For limitation of service, Google allows only 50 users to write a document together at a same time, others participant can only see the document without changing it.

To demonstrate SOCIOPLUG benefits, we propose to deploy alternative version of collaborative editors on a federated infrastructure of plugs in order to tackle previous issues.

Objectives and challenges This use-case demonstrates massive collaborative editing on SOCIOPLUG infrastructure. Compared to state of art, it requires no central component, have no intrinsic limitations in number of users, and all communication remains private among participants. We can imagine a massive online lecture where thousands of attendees can edit lectures notes in real-time.

A user can start a collaborative editing session on his browser as he can do with google doc. As side effect, it will creates a "collaborative editing session identifier" on his personal plug and share this with invited participants. Reading this invitation allows remote collaborators to access necessary data to establish browser-to-browser connections. Once connections established, real-time editing can run with as in traditional editors. Plugs are included as participants for ensuring reliability of real-time session and ensure persistency and indexing of documents

This use-case is challenging on many aspects:

- for consistency management. It requires to write an efficient CRDT for managing sequence that ensure at least eventual consistency. Such CRDT should offer bounded identifiers and resist to causality errors.
- for communication layer. It requires to have a reliable broadcast able to tolerate small groups and big groups.
- for federated queries, it should be possible to query efficiently the whole federation about all documents concerning a particular topic. The results set should include personal document, document shared by other but visible by me and public documents.
- for usage control. Documents should be shared with others through contracts fixing usages authorized on the particular document. Usage rules can include no-indexing, no distribution or deletion rules. The system should be in change to control if usage control is ensured by each member of federation.
- Finally, for monitoring protocol, system is in charge to detect any attack on real-time session.

Task interaction Addressing consistency issue challenge is a clear objective of task 1 and require tight collaboration with communication layer. How documents can be indexed and retrieved from queries requires collaboration with task 2. how documents are used after retrieval is clearly a challenge for task 3.

4 Scenario 3: Smart city & collaborative sensor networs

Preamble In this scenario, we consider some prototypical individual named Alice with different concerns about her life in a big city. For example, she does not like providing her energy supplier with the details of her home energy consumption without possibility of any control of the supplied data. Alice's kids are sensitive to birch and ambrosia pollen, especially when the temperature exceeds 22 °C and the rate of nitrogen dioxide is greater than 50 $\mu g/m^3$. The more she would able to approximate and anticipate the current quality of the air in her family's neighborhood, the better she could provide them with the most appropriate treatment. She would like a finer-grained information than air quality of the whole agglomeration such as provided by current observatories. As for her terrace plants, the sooner she is alerted of stressing factors (specific weather and pollution conditions, presence of insect pests) the better she can take care of them. Finally, Alice has got a car and a bike and she also uses public transportation. Everyday, she decides which one to use according to the traffic and weather conditions and to her needs for flexibility. For example, she favors public transportation when the roads are frozen or about to be so and when there is an air pollution alert.

In other words, to adapt and improve her urban life, Alice should be able to (continuously) query data, such as sensor-issued measures or people' alerts.

On the other hand, recent technological advances make it possible to publish sensor data and other data streams in a rather simple way. One could then envision a near future in which, in addition to institutions and corporations, Alice and her fellow citizens could also publish some data on their own. The problem then is to provide a solution that would enable Alice to take a full advantage of the diversity and richness of the existing streams.

Centralized solutions such as Google Alert [Goo14a], based on the monopoly of a third party, suffer from several drawbacks. First, the user's choice is limited to the streams chosen by the third party. Second, the users' queries become the third party's property, who is the only one able to exploit them (either to optimize query evaluation or to mine them). Third, data and query privacy is not always ensured.

In the next section, we describe how a decentralized plug-based solution such as SOCIOPLUG could lead to a more open, participatory solution where everyone may contribute while keeping control over one's personal data.

Objectives and challenges The aim of SOCIOPLUG is to provide a whole computer-plugs-based infrastructure that would enable people, both corporations and individuals, to query, combine and publish data streams in order to acquire more fined-grained and more dynamic information to enhance people's everyday life, while preserving one's control over her private data. SOCIOPLUG seeks a collaborative distributed query evaluation that enables to compute (continuous) queries in particular those that require a greater capacity than what a single plug may offer. Considering prototypical participant Alice, the general objective is threefold.

First, we want to provide an easy to use infrastructure that enables simple insertion of new plugs and connection of additionnal sensors to plugs, and provides an interface for human-plug interaction, for example on a smartphone.

Second, SOCIOPLUG should enable Alice to query the system using already existing streams and to combine them for personal use or dissemination. Once her plug (eg. raspberry pi) is connected to the network, she only has to ask for the information she needs. For instance, Alice could ask for the temperature, rate of nitrogen dioxide, and levels for ambrosia and birch pollen every hour in Lyon 3rd district (query Q_1) without knowing a priori which streams should be used. Notice that the result could use both streams provided by corporations such as "Observatoire Air Rhone-Alpes" and streams provided by Alice's own neighbors as their sensors very precisely provide the temperature in her district. In order to compute the answer, the system performs a discovery of relevant available data streams. When query Q_1 is already being

computed in the system, Alice's plug is just invited to contribute to its collective computation ; otherwise, the result starts being computed using Alice's plug(s) and plugs that compute related queries. In the same way, in order to ease car-pooling of scenario 1, SOCIOPLUG should enable Alice to know the current traffic conditions together with the temperature and humidity every day, and every minute between seven and nine in the morning, on the road from Corbas to Villeurbanne (query Q_2). The objective of SOCIOPLUG is to consider queries that not only involve different types of sensors (temperature and humidity sensors, magnetic loops to get the numbers of vehicles in real time...) but also manual alerts provided by individuals (drivers who mention a car crash, people at home who find their ground is frozen...). SOCIOPLUG also aims at answering "historical" queries, such as "daily presence of aphids and scale insects in my district, starting from three months ago". Here, even she didn't care for aphids and scale insects before, Alice is not only informed of possible current problems but also of past ones. Here the challenge is to provide SOCIOPLUG with a recording service that enables to store some results and to share them.

Finally, SOCIOPLUG aims at having people contribute and publish streams of there own. To do so, Alice who stores personal data on her plug should specify the privacy policies provided by SOCIOPLUG, describing how her data may be used and how/when it should be forgotten. Alice then can publish several own streams such as those coming from the sensors she has installed in her flat. She can also decide to query existing streams, to combine them with her own and disseminate the resulting new stream, for example to provide the average temperature at every floor of her building, every hour. Or she could aggregate all the temperature streams of the region. Using SOCIOPLUG Alice should also be able to provide a stream of manual alerts, such as the presence of aphids on her terrace plants every day. If several people do the same, combining their streams could lead to an early alert of the first insects detection. She could also inform of her window temperature being below 2 °C which means that the streets are probably frozen. Finally, as presented in scenario 2, with other people of a district she may want to collaboratively edit a document about the presence of aphids and scale insects in the district, with description of the insects found on the plants. Then a stream is published which notifies of new versions.

Reaching the goals presented in this scenario raises several challenges. First, SOCIOPLUG should provide an efficient and easy to use infrastructure, with effective and fast integration of computer plugs into the whole network. It should also propose a way to connect sensors and a man-machine interface to one's computer plug.

A very challenging problem consists in collaborative distributed query evaluation and optimization of continuous queries and the storage of results. This requires defining algorithms and protocols to maintain overlays of collaborating plugs and distributed query evaluation algorithms within a community, and inter-communities. A particular attention should be paid to the plugs capacity as a goal is to have people get the results they are interested in even if their plug, taken alone, has not enough capacity to compute them. Finally, distributed storage of results within a community is challenging due to scalability and dynamicity issues, raised by the size of the data and their frequency, the potential numbers of streams and queries, and even the possible disconnections of plugs.

Task interaction This scenario is challenging for Task 2. Solutions must be provided for distributed query evaluation and optimization of continuous queries. The possibly very high number of contributing plugs and users' queries makes it essential to design scalable solutions, through query-driven organization of collaborating plugs, while considering the plugs capacity. This scenario is challenging for Task 3. Data streams should be shared with usage control policies (i.e., contracts) establishing authorized usages. Usage rules can include no distribution, no combination with other streams, time to deletion, time to obfuscation (e.g., after a while, only aggregated data can be shared), usage purposes (e.g., no commercial use), etc. As each stream has a particular usage control policy, when combining streams, policies should be combined too. The whole federation can be organized in (overlapping) trusted communities and each (sub) federation should be able to guarantee that usage control policies are preserved.

5 Scenario 4: Collaborative content sharing & Google Drive

Preamble Personal cloud application have become a common and convenient way to share data among people, and between devices belonging to the same person. Services such as Dropbox [Dro14] and Google

Drive [Goo14b] for instance allow their users to store files on one device and access that file on another, while a simple application takes care of its synchronization in the background. While convenient, this mode of operation has two important drawbacks. First, companies offering this services often retain the right to access user data for their own purposes. This discourages the use of such tools for sensitive and/or confidential information. Second, in order to transfer data between two devices that may be in the same building or even in the same room, personal cloud services may transfer the data back and forth across a continent. This can be particularly inefficient in terms of both network usage and latency, in particular if one considers the increasing volumes of data users generate (photos, movies). This can also be problematic in terms of legal protection, and customer rights, as sensitive content and data might end up stored in data centers that don't fall under the jurisdiction of the users' home country, substantially weakening the threat of potential trials for infringing players.

The objective of this scenario is to explore how a decentralized plug-based social network such as the one envisaged in the SOCIOPLUG project can provide an alternative to content-sharing services provided by a centralized player, with a richer set of functionality, and mechanisms optimized to the specifics of a decentralized plug-based network.

Objectives and challenges We consider again a prototypical user, Mary, who wishes to share files with other users as well as among the various computing devices she owns. Mary has a busy life: she works from home as an interior designer, and cares, with her husband John, for their three children. As a home worker in the creative industry, she must often share content and documents with customers, agencies, agents, etc. As a mum of three, she also often has pictures and movies of everyday life that she and her husband want to share with their family and close friends. John is also a seasoned blogger about maritime life in North-West Brittany, and wants to share content about this topic as broadly as possible on SOCIOPLUG's social network.

At a very basic level, SOCIOPLUG may operate as a standard file server replicating the features of services like Dropbox or Google Drive, offering the ability to distribute and replicate content between devices, and share files with specific users. But SOCIOPLUG also offers the ability to exploit knowledge gathered about users to enrich and improve their experience, without incurring the risks of one or several players acting as omniscient Big Brothers.

Both Mary and John might share certain types of pictures with different subgroups of users, depending on the pictures' topics and content. Mary might predominantly share pictures of home life and holidays with her close family and friends. She will share pictures related to her work as an interior designer with prospective customers, providers, architects, and magazine editors. John will mostly share pictures about maritime life with his blog followers. For both of them, SOCIOPLUG can learn to detect which picture corresponds to which audience (e.g. based on tags, image analysis, or context), and it can proactively send copies of the media to the targeted audience in advance (while respecting Mary and John's privacy setting). Moreover, the plug-based architecture can achieve this while exploiting idle cycles and the nightly availability of the network.

Generalized to unknown users, the same mechanism can uncover people with interests that are related to those of Mary and John, and thus recommend the content they have marked as public. For example, SOCIOPLUG might create dynamic communities consisting of users who have shown interest in maritime life, and start proactively disseminating some of John's recent work. This kind of biased dissemination is important (but challenging to achieve) because it avoids two extreme behaviors. On the one hand, it avoids replicating all publicly published content to all users' devices (a fully impractical solution with a large user base). On the other, it goes beyond replicating content on demand (i.e. sending John's pictures to Mark only when Mark requests them), a particularly slow solution that is hard to scale with home-based devices.

Finally, current content-sharing solutions do not typically handle conflicts well: conflicting copies are simply marked as such and stored away. SOCIOPLUG can improve on this model, particularly for meta-data (tags, people on a photo, or topics), using some of the CRDTs (Consistent Replicated Data Types) developed in the project.

Task interaction This scenario is mainly linked to Task 1 (Consistency and Programming), and to some extent to Task 2 (Social Query Evaluation and Valorization) and Task 3 (Data Privacy and Safety). Task 1 will provide the mechanism to adapt the underlying SOCIOPLUG overlay to provide an efficient dissemina-

tion of content taking into account a user's typical diffusion patterns, and enable the proactive replication of content based on earlier distribution and access patterns. Task 1 will also contribute the CRDT structure to enable concurrent meta-data processing of content (e.g. to allow the concurrent identification of people and topics, the addition of tags, or the creation of a web of related documents). All these different abilities must be properly protected and framed to ensure the privacy of users, which is where the mechanisms developed in Task 3 will be applied. We will also explore how the abilities of Task 2 can be used to identify users with shared interests, and propose content proactively while respecting the architectural constraints of the plug network.

6 Architecture and Summary

6.1 Architecture

Infrastructure Computing plugs such as the Raspberry Pi are in the direct continuation of works on micro-clouds on edge-computing infrastructure that seek to leverage the storage and computing power installed in customer homes to lower latencies and costs, while alleviating reliance on third party cloud services. Plugs are particularly interesting because they open up the possibility of decentralized services on low-cost, low-churn infrastructures.

Plug computers allow any of us to maintain cheap nano-clusters of Raspberry Pis at home, and host data and services. These plug-based systems offer a middle-ground between traditional data-centers and P2P networks. In contract to data-centers, plug-based systems rely on autonomous participants. Differently from P2P network, plug-based systems provide a stable infrastructure that exhibits little churn, and opens the possibility of novel and optimized mechanisms.

Moreover, federation of plugs also breaks with other federations by the number of participants they encompass. For example, Linked data federation is estimated to 300 to 700 participants, Diaspora pods or Mediagoblin servers are estimated to contain less than a few hundreds of servers. It is clear that a federation of plug computers should be estimated in millions of plugs. A federation of plugs should also consider limitation of plug computers i.e. low memory (<1Go), low reliability of plugs, CPU limitations. At last, plug computers are installed in users' homes by people with no skill in system administration. Finally, the topology of the federation will be mostly based on social links.

In our infrastructure, the core problems rely on *(i)* memory constrains, *(ii)* latency concerns (shared memory and message passing share the same bandwidth). By the way, our infrastructure is design similarly to a classical cluster for grid computing. Such, developing and deploying services on our platform is more or less the same than on Grid5000. Some simple application has been developed recently as proof of concept on a 20-plug cluster. We plan to extend some functionalities such as fine-grain configuration (setting the latency of each plug separately for instance).

The next step is to increase this infrastructure up to 200 plugs located in Nantes, and a dozen of plugs spread among participants (at IRISA and LIRIS laboratory and at members homes) in order to test social-based and large-range applications in real environment. The main objective to this platform is to provide an emulator of real environment, instead of cluster-based computing, which is not the target infrastructure of our services.

Architecture Based on limited capacities of plugs and the objectives of the different scenarii presented above, we present some characteristics that the architecture of the platform has to reach.

The local plug, installed at home for instance, is the only point of view of a user that use the federation of plugs. This plug acts for such user as the entry point of the network, by direct connection or via different connected devices (computer, tablet, smartphone, *etc.*), but also receive several information, such as data stream produced by connected IoT as meteorological sensors for instance.

Thus, we identify some mandatory modules that should be present on each participating plug, in order to provide distributed services on the infrastructure. The following list introduce some topic-based modules, but is not exhaustive. It will be adapt during the cycle life of the project:

Stream processing module Treatment of data flow from sensors, actuators, messages, *etc.* using SoCQ for instance;

	Scenario1: Plug Track & "Google Now"	Scenario2: Collaborative editing & "Google Doc"	Scenario3: Smart cities & collaborative sensor networks	Scenario4: Content Sharing & "google drive"
Collaborative Edition		++++		
Social Overlay	++++	++	++++	+++
Models & Learning	++		+	
Privacy & Usage control	++++		+	+
Continuous requests	++	+	++++	
Recommendation	+++		+	+
Data consistency		+++++		+++
Security & Safety	+++	++	+	+++
Data mining	++++		++	

Table 1: Use-cases aims wrt project objectives

Query processing module Management of collaborative queries, based on QTor approach for instance;

Fellowship module Handling of gossip-based mechanisms, aggregate queries, learning, recomendation system, *etc.*;

Collaborative editing module ;

Privacy module Administration of confidentiality, privacy and usage control;

Safety module Fault tolerance, network monitoring via stream analysis, performance evaluation, *etc.*

6.2 Summary

In this report, we have presented four use-cases we aim to evaluate in SOCIOPLUG and progress on evaluation infrastructure. Table 1 summarizes these use-cases and focuses of each scenario. These scenarii emphasize different aspects of SOCIOPLUG contributions and require collaboration between at least two partners. On each use-case, there is at least one partner with strong expertise. Moreover, these scenarii has been chosen according to their complementarity and specificities. We argue that each use-case represents a pertinent socio-economic interest while their goals cover the whole objectives of the SOCIOPLUG project, as illustrated by Table 1.

References

[ABH08] Fabian Abel, II Bittencourt, and Nicola Henze. A rule-based recommender system for online discussion forums. ... *Web-Based Systems*, pages 12–21, 2008.

[Das07] Abhinandan Das. Google News Personalization : Scalable Online. pages 271–280, 2007.

[Dro14] Dropbox Inc. Dropbox. `https://www.dropbox.com/`, 2014. accessed 30 August 2014.

[Fou14] Foursquare Inc. Introducing the all-new foursquare, which learns what you like and leads you to places you'll love. `https://foursquare.com/download`, 2014. accessed 29 August 2014.

[Goo14a] Google Inc. Google alert. `https://www.google.com/alerts`, 2014. accessed 30 August 2014.

[Goo14b] Google Inc. Google drive. `https://drive.google.com`, 2014. accessed 30 August 2014.

[Goo14c] Google Inc. Google now. `http://www.google.com/landing/now/`, 2014. accessed 29 August 2014.