SOCIETIES
FP7-ICT
Contract no.: 257493

# SOCIETIES

## Deliverable D1.3

## Technical Coordination Report

| | |
|---|---|
| Editor: | Alec Leckey (INTEL) |
| Deliverable nature: | Report |
| Dissemination level: (Confidentiality) | PU |
| Contractual delivery date: | 28th February 2014 |
| Actual delivery date: | TBC |
| Suggested readers: | Project Collaborators |
| Version: | 1.0 |
| Total number of pages: | 22 |
| Keywords: | SOCIETIES, Self Orchestrating CommunIty ambiEnT IntelligEnce Spaces, Communications, devices, services, dependability, social networks |


***Abstract***

The intention of this deliverable is to report on the technical coordination activities undertaken during the execution of this task, including overall technical direction, strategy and key achievements for SOCIETIES. This report summarises the overall technology and development strategy adopted and implemented for the SOCIETIES project. It then discusses the key achievements and challenges that occurred during the project and where the role of Technical Coordinator impacted the results.

**Impressum**

[Full project title] Self Orchestrating Community Ambient Intelligence Spaces

[Short project title] SOCIETIES

[Number and title of work-package] WP1 Project Management

[Document title] D1.3 Technical Coordination Report

[Editor: Name, company] Alec Leckey, INTEL

[Work-package leader: Name, company] Kevin Doolin, TSSG

[Estimation of PM spent on the Deliverable] 1 PMs

**Copyright notice**

# Executive summary

The vision of the SOCIETIES project is to research and develop a concept of Ambient Intelligent (AmI) Communities (also known as Pervasive Communities) that extend ambient intelligent or pervasive systems beyond the individual to dynamic communities of users. Driven by context awareness, preference learning and privacy protection, AmI Communities will intelligently connect people & things to communicate, share, consume and organise communities. SOCIETIES will embrace online community services, such as Social Networking, thus offering new and powerful ways of working, communicating and socialising. The project intends developing and trialling use cases for disaster management, university living and conference support.

The intention of this deliverable is to report on the technical coordination activities undertaken during the execution of this task, such as overall technical direction, strategy and key achievements for SOCIETIES. This report summarises the overall technology and development strategy adopted and implemented for the SOCIETIES project. It then discusses the key achievements and challenges that occurred during the project and where the role of Technical Coordinator impacted the results.

The project's objectives were challenging and it required strong technical management to keep the project on track to achieving its results. The design and development phases of the project were very challenging but also successful, given the project structure and natural complexities of a large collaboration project with partner organisations of different approaches. The role of Technical Coordinator was to manage the technical activities of the project and ensure the project objectives were met by the work packages as much as possible. This was achieved through the adoption of a unified development methodology, good communication, hands-on management of various activities and a strong team effort from many project members.

# List of authors


| Company | Author |
|---|---|
| INTEL | David McKitterick, Alec Leckey |

# Table of Contents

# 1 Introduction

An Ambient Intelligent (AmI) Community (also known as a Pervasive Community) is a group of, two or more, individuals who have agreed to share some, but not necessarily all, of their pervasive resources – personal information, context data, services, devices – with other members of that community. AmI Communities have the potential to completely transform traditional online social networks, freeing them from web-applications and letting them loose in the real physical world. SOCIETIES supports the creation of AmI Communities by discovering, connecting and organising relevant people and things from both physical and digital environments. SOCIETIES will use pervasive technologies to form adaptive communities, while leveraging social networks and crowd computing techniques.

The enabling features of the SOCIETIES platform for discovering, connecting and organising communities is driven by context awareness, preference and intent learning and privacy protection. SOCIETIES helps users discover people and communities relevant to them and their context so they can communicate, share and collaborate. In our university living scenarios, relevant students discover each other to discuss topics, share study notes and meet up when they are automatically discovered to be nearby. SOCIETIES enables users to connect to other people and things relevant to them in order to discover information and consume services. In our conference support scenarios, attendees access personalised agendas and real-time session streams, while utilising indoor navigation and professional networking services. SOCIETIES empowers AmI Communities to be self-organising by utilising the community context and preferences to direct actions, taken on behalf of the community. In our disaster management scenarios, disaster victims are safely evacuated away from disaster areas by providing individual evacuation instructions, based on the collective community context.

This project builds on previous research from the PERSIST project (1) which defined a context-aware and self-improving personal smart space (PSS) for users to interact dynamically with smart environments and other users' devices. Building on the PSS concept, SOCIETIES has introduced a set of more community-centric concepts. A Co-operating Smart Space (CSS) represents a single participant (user or organisation), and includes their information, and services within a distributed system of CSS Nodes (user devices/cloud instances). A Community Interaction Space (CIS) represents and provides the interaction mechanisms for an AmI community to exist. Members of an AmI community interact with a CIS via their own personal CSS. The individual members of an AmI community do not need to be human beings but can also be organisations, smart space infrastructures or autonomous agents. AmI Communities can be dynamic in nature, with CISs being formed in an ad-hoc fashion. CISs can also spawn sub-communities or merge with other communities, and will be self-orchestrating, making decisions for its members based on the community content and preferences.

The task of Technical Coordination has one main objective, which is to execute the role of Technical Coordinator and achieve the duties outlined within this role. This role includes:

- Responsibility for overall technical coordination and management for the project
- Defining the technology and development strategy
- Coordinating the technical activities between all work packages
- Working with the project coordinator, research coordinator and work package leaders to plan, monitor and direct all general technical aspects
- Ensuring that the technical strategy and development roadmap meet the needs to the project's objectives and innovation targets
- Ensuring that all technical milestones are meet and delivered to a high standard
- Promoting the use of open standards and technical collaboration with other projects and bodies
- Moderating on technical decisions and managing any situations of conflicting choices

The intention of this deliverable is to report on the technical coordination activities undertaking during the execution of this task, such as overall technical direction, strategy and key achievements for SOCIETIES.

Chapter 2 to summarise the overall technology and development strategy adopted and implemented for the SOCIETIES project. The general project methodology and planning is discussed, followed by the key aspects on the project's technology and development strategy.

Chapter 3 aims to discuss the key achievements and challenges that occurred during the SOCIETIES project and where the role of Technical Coordinator impacted the results.

A glossary of terms used in this document can be found on the SOCIETIES project website (2).

# 2 Technical and Development Strategy

This chapter aims to summarise the overall technology and development strategy adopted and implemented for the SOCIETIES project. The general project methodology and planning is discussed, followed by the key aspects on the project's technology and development strategy.

 

## 2.1 Project Methodology

This section describes the project wide methodology that was adopted by the project in order to achieve its objectives. Each technical WP is involved in a phase, or number of phases, of the project lifecycle. These phases include:

- Requirements: specified the requirements for the entire project, including scenarios, use cases and business cases
- Research: specified research activities needed to realise the many innovations proposed by the project
- Implementation: developed prototype software and services in preparation for user trials
- Integration: provided a completely integrated platform for the SOCIETIES innovations and features, integrated with a set of services specified for user trials
- Trials: rolled out the SOCIETIES system in real user environments
- Evaluation: evaluated the results from a number of perspectives (sociological, business and technical).

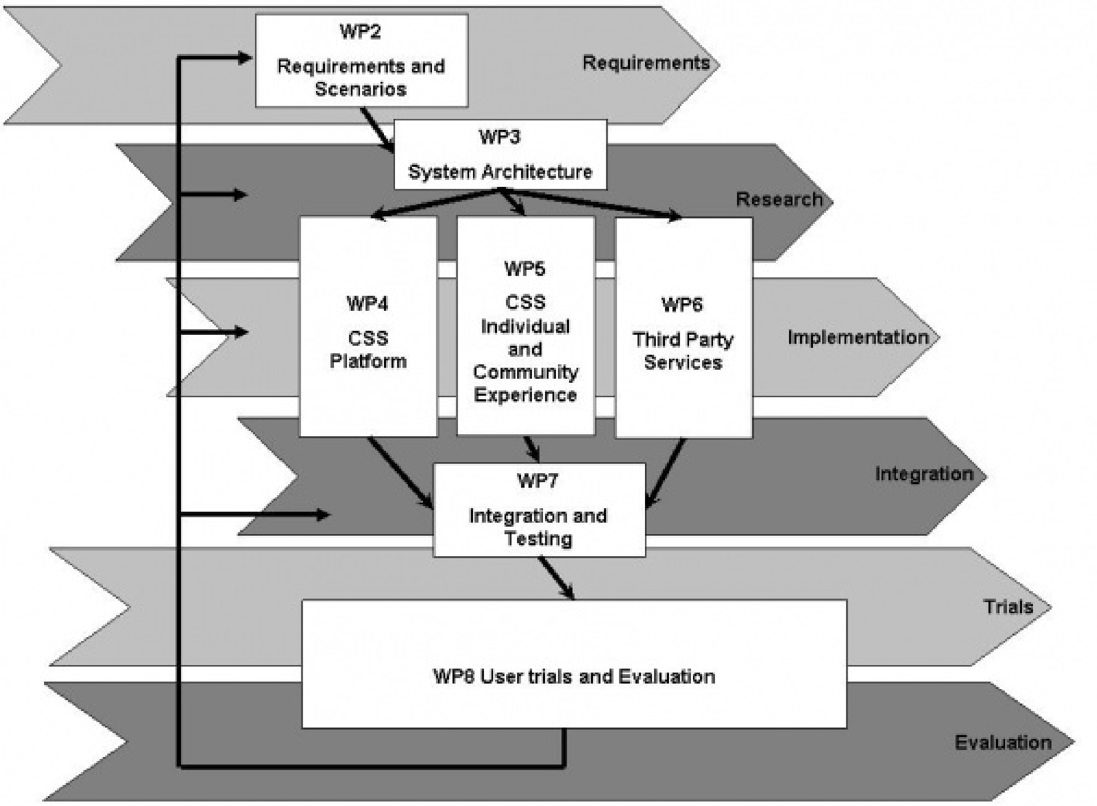


**Figure 1: High Level Project Methodology**

Work package 2 is responsible for establishing the complete set of requirements for SOCIETIES - these requirements come from a number of different perspectives:

- User perspective: documented the requirements specified by the three user groups identified for the SOCIETIES trials and evaluation.
- Technical perspective: technical requirements were derived through WP2's scenario and use case research.
- Business perspective: documented the requirements of the business community of the concepts of the SOCIETIES project.

The scenarios, requirements and use cases produced by the tasks of WP2 were used as input to nearly all other work packages within SOCIETIES. Work package 3 was responsible for specification of the SOCIETIES system architecture. WP3 took guidance from the requirements specified in WP2, and ultimately produced the complete integrated system architecture. With the architecture specified, the project moved in to a phase of deep research and implementation encapsulated by work packages 4, 5 and 6.

Work package 4 had the goal of researching, designing and prototyping the functionality required to realise the SOCIETIES platform. It took its direction from the architecture produced within WP3 and proceeded to produce an overarching design for the SOCIETIES core system, which in turn was used to inform tasks T4.2-T4.6 each of which concentrated its research and implement efforts on its own unique but integrated area. The output deliverables from WP4 constituted the integrated SOCIETIES platform which was the base upon which the CSS individual and community experiences (WP5) and support for third party services (WP6) could be provided. Work package 5 took the requirements and scenarios from WP2, with the architecture from WP3 and produced the functionality which supported the provision of the enhanced user and community experience aspects of SOCIETIES. Work package 6 built on the design and integrated prototypes from WP4 and WP5, and along the system architecture and requirements, it designed and prototyped a service creation environment and the required 3rd party services necessary to execute the user group software trials.

WP7 tied the implementation work from WP4, WP5 and WP6 into a complete integration and pre-validation test-bed. The goal of WP7 was to integrate and to pre-validate the integrated SOCIETIES prototype on a smaller scale than in the trials. This test-bed acted as a staging post prior to trial deployment in WP8. Therefore, it tightly cooperated with work packages 3, 4, 5 and 6. A test-suite and an integration plan was used to enforce an efficient and successful integration of the core platform modules provided by the other work packages. As is the case with many projects of this scale and nature, the integration task required strong management and planning, and strict enforcement of project milestones, hence there were a number of integration/test plans, prototypes and reports. Finally, work package 8 was responsible for carrying out the user trials for SOCIETIES. In a similar case to WP7, strict adherence to milestones and deliverable requirements was essential in order to correctly manage the user trials carried out for the SOCIETIES project.

### 2.1.1 Project Critical Paths and Iteration Cycles

Following the project launch, Milestone 2 (MS2) ensured that all user requirements were captured. Requirements were fed into the initial paper trial, following which the requirement set were revised based on initial user feedback. In addition, the initial project architecture work took input directly from the finalised set of requirements. Feedback from the paper trial was given to WP3,4,5,6 as input to their respective design work.

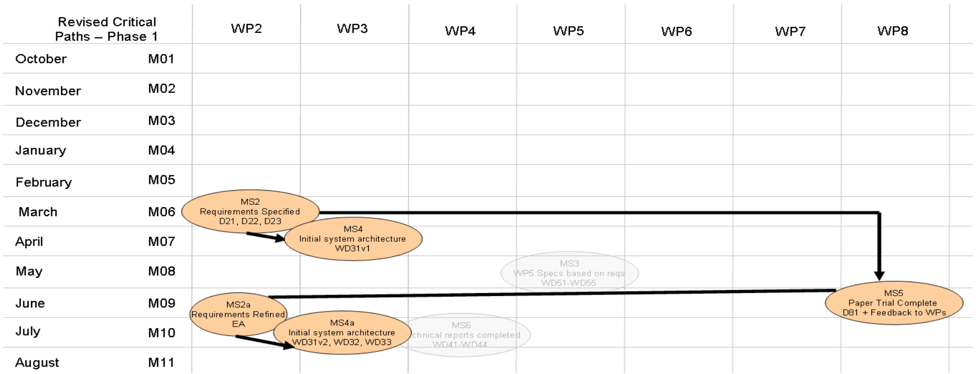


**Figure 2: First Critical Path**

Following the development of the SOCIETIES initial architecture, WP4 and WP5 each started a specification and design phase (verified at MS8/9), followed by development and internal integration activities (MS12/13). In parallel, WP6 embarked on the initial design and development of the SOCIETIES service creation environment and first set of services (MS14) for the first user trial phase. At MS15, WP7 produced the first SOCIETIES integrated prototype, which was deployed to the trial sites for onsite testing and pilot trials before delivery to the first project trials (MS19). Outputs from the trial were given as feedback to WP3 (for architecture) and WP4,5,6 (for design and development).

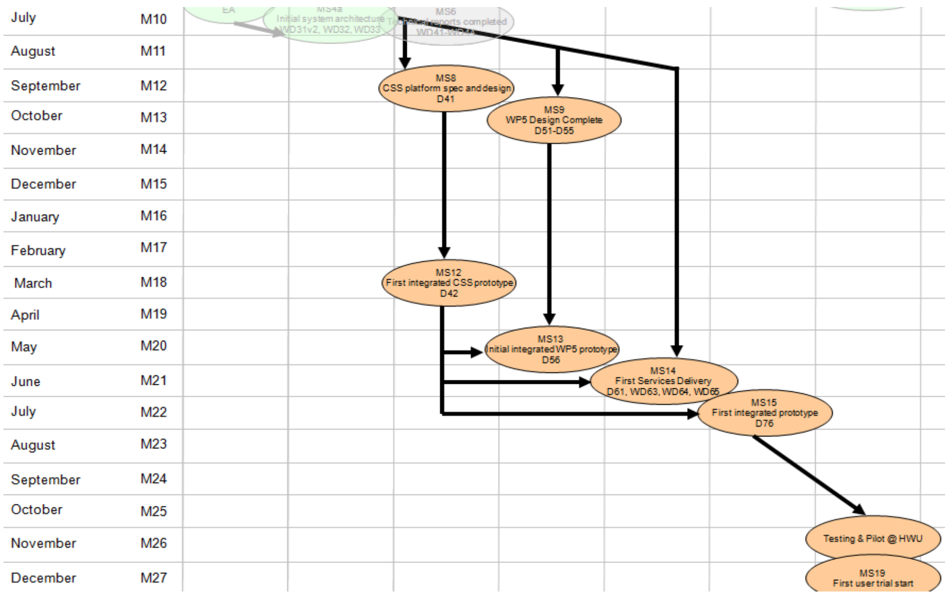


**Figure 3: Second Critical Path**

Following a revision of the SOCIETIES architecture, taking trial feedback into account, WP4, WP5 and WP6 produced their second integrated prototypes which were combined within WP7 (MS27) as the integrated prototype used for the second SOCIETIES trial period (MS30/31). This prototype passed conformance testing against the SOCIETIES requirements (MS28) before trials commenced.

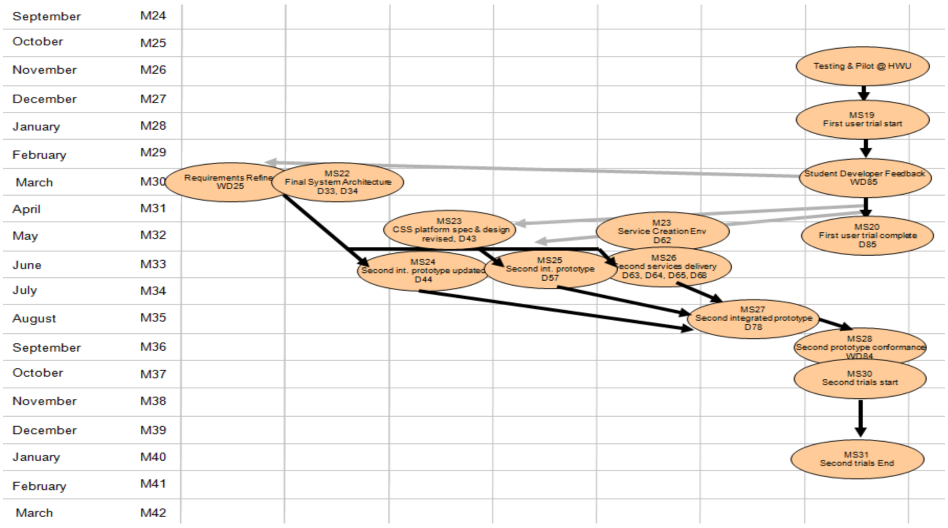


**Figure 4: Third Critical Path**

As the project progressed, there were various updates to the project plan to enable the project to adapt to changes in research objectives, delays with development and other challenges that occurred. The critical paths as discussed previously in this section are the revised plans as of Month 35.

### 2.1.2 Scenario Driven Requirements and Use Cases

During the first six months of the project, WP2 managed the requirements analysis effort focusing on initial user requirements from our specific user groups, technical requirements and use cases from the project team, and also business requirements and modelling to ensure that the business interests are captured. The process used in WP2 was based around a scenario driven methodology, where various revisions of conceived scenarios were used to drive the extraction of user, technical and business requirements.

To extract the user requirements, a systematic investigation regarding people's current experiences using systems similar to a SOCIETIES system was conducted, as well as regarding user's view on the applications and services offered by the envisioned SOCIETIES environment. To this end, participatory methodologies were used, as the necessary data was collected by communicating directly with potential end-users of the SOCIETIES system. Based on the extracted user requirements and guided by the SOCIETIES' vision, a collection of scenarios were designed. Thus, both generic scenarios and use cases were produced that were application independent, as well as scenarios focusing on the three specific application domains, i.e. Enterprise, Student and Disaster scenarios. Based on a thorough analysis of these scenarios, the functional and non-functional technical system requirements were specified. Finally, in order to ensure that the SOCIETIES system was of considerable business value introducing new sources of revenue and new business opportunities for the stakeholders involved, business requirements was also extracted. Thus, based on a market analysis of the related domains, commercial scenarios and business use cases suitable were designed and thus used to extract the business requirements and models, as well as the respective business opportunities, guidelines and methodologies.

The resultant output of the scenario and requirements analysis was entered into the project's UML modelling repository, via the Enterprise Architect tool, and these were used in a consistent structure as inputs to the system architecture and implementation work package design. The inputted models included detailed scenarios, user requirements, technical requirements and use cases, business requirements and stakeholder analysis.

### 2.1.3 System Architecture Viewpoints

The methodology for designing the system architecture was to use a four viewpoint specification to capture all perspectives based on the system requirements analysis. These viewpoints were:

- Application Architecture: Non-functional principles and definitions enabling to define a platform independent distributed system
- Technical Architecture: Technical services, facilities and patterns required to develop, deploy and run the application using specific Platform Technologies
- Functional Architecture: Functional viewpoint of the system reflected the problems in implementing formal decomposition of requirements. The SoaML process was used for defining the functional viewpoint, through using the WP2 use cases for the SoaML capability analysis and applying the WP2 requirements to extracted services.
- Information Architecture: Often the Information Architecture is defined within Functional Architecture. A glossary was the first draft outcome that was then further refined by creating a well defined class models with attributes and relationships using UML class diagrams and notation. It described the relation across the domain class entities.
- Management Architecture: Governance principles enabled a large team to cooperate in building a large system efficiently

#### 2.1.4 Iterative Design & Development Cycles with User Evaluation

The section describes the inter WP and internal WP processes needed to achieve the major iterative cycles as discussed above in the critical paths. Each cycle included a design, development and user evaluation phase which corresponded to planned work performed by work packages 3, 4, 5, 6, 7 and 8.

#### 2.1.4.1 Iterative Design Phase

This phase followed a mostly iterative approach and started when the requirements (user, technical and business) and use cases were revised and updated. The UML repository of requirements artifacts were updated by the requirements owner WP and these revisions pushed a further revision to the WP3 system architecture design and eventually a revision of the respective WP4-6 specifications. During each major cycle, there was a one design phase planned with output defined as UML models which was produced as part of working documents to deliverables.

#### 2.1.4.2 Agile Development Phase

The prototype development phase followed an agile approach where features of the defined specifications were developed from rapid prototyping and continuous integration. The WP4-6 specifications were modular in design thus allowing for de-coupled component implementations with API dependencies. A detailed release plan with regular interim releases of individual components allowed for continuous integration within the integrated WP prototype and continuous feedback between respective WP developers. Each agile development phase ended according to the delivery milestones in the overall workplan. Continuous integration included continuous testing and WP7 coordinated a common testing framework to ensure consistent in test coverage.

#### 2.1.4.3 Iterative User Trial Evaluation Phase & Continuous User Feedback

The user paper and software trials were planned be executed following the respective design and development phases. Each user evaluation phase included conformance to requirements testing (except for the paper trial), a user trial period and an analysis of user feedback in terms of an evaluation report. Continuous user feedback was recorded as feature and bug requests from 3rd party developers (internal and external) and end users during the trials.

## 2.2 Technology Strategy

The technology strategy of this project needed to meet many requirements from the initially defined scenarios coming from the 3 user group domains in addition to the original project vision which specified the need for open standards, interoperability and portability. The system architecture (as summarised Figure 5), which was based on the derived technical requirements, also provided extensive requirements to the technology strategy for a complex distributed system which was platform independent, could interface with existing systems such as social network platforms, could be deployed to multiple device types and pervasive environments, and could support a variety of 3rd party services which were targeted to built on top of the SOCIETIES platform.

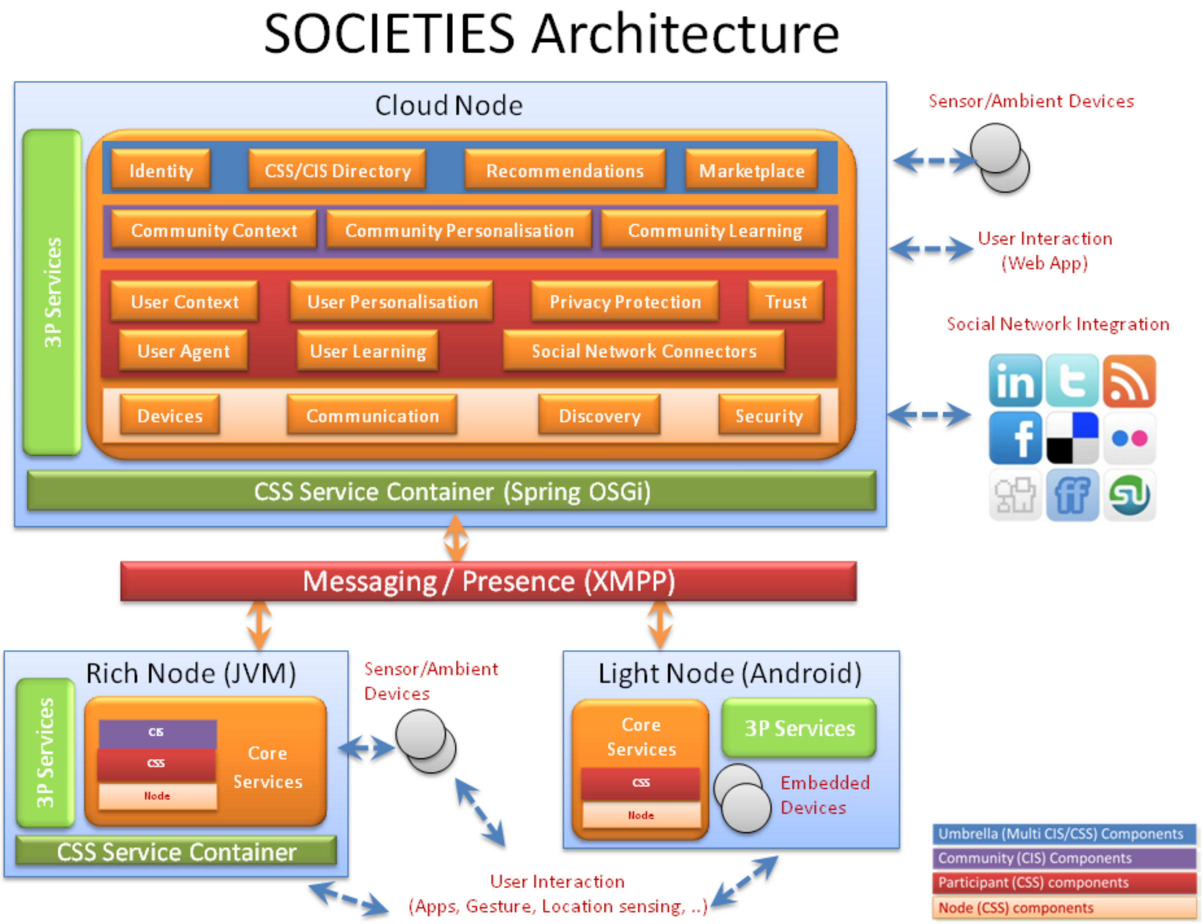


**Figure 5: SOCIETIES System Architecture**

Before any final technology options were decided, within WP4, extensive technical research that was conducted based on research objectives and perceived technical problems that the SOCIETIES platform would be required to address. Each area was called a Collaborative Research Area (CRA) as multiple partners and tasks contributed to this work. The CRAs investigated existing standards, protocols and leading edge technology platforms covering areas on communication and device abstraction, service infrastructure, dependability and security, CSS and CIS management, and social network integration. The design deliverable, D4.1 'SOCIETIES Platform Specification and Design' (3), reported on background technical research that was conducted. D4.1 also described some design patterns for failure, discovery and interoperability which are used to guide the design and development of the integrated prototype. The updated design deliverable, D4.3 'SOCIETIES Platform Specification and Design (4), Revised', provided a revision to the design of the SOCIETIES platform based on technical requirements, system architecture design and technology considerations arising from the deployment of a prototype implementation.

 

### 2.2.1 Target Platforms

Based on the system architecture, as defined in the WP3 deliverable D3.3 'System Architecture' (5), technology platforms were required for mobile devices and a cloud infrastructure. The user trial requirements also required a solution which could be portable to multiple mobile handsets given the variety in use at the time of the initial user analysis. The smartphone mobile platforms that were investigated included: Android, MeeGo, Windows Mobile and iOS. The laptop/netBook/tablet platforms included Chrome, MeeGo, iOS, Linux and Windows. In order to support multiple mobile platforms, it was decided to prototype a hybrid native web application which included a HTML5 and Javascript based presentation layer, native components for better device integration with Phonegap (6) extensions to enable interfacing with the presentation layer. Figure 6 shows the components of the hybrid web application using Phonegap as an interoperability layer. The target for the prototype implementation was to be developed based on the Android OS initially and then aim for porting the application to other OSs such as iOS.

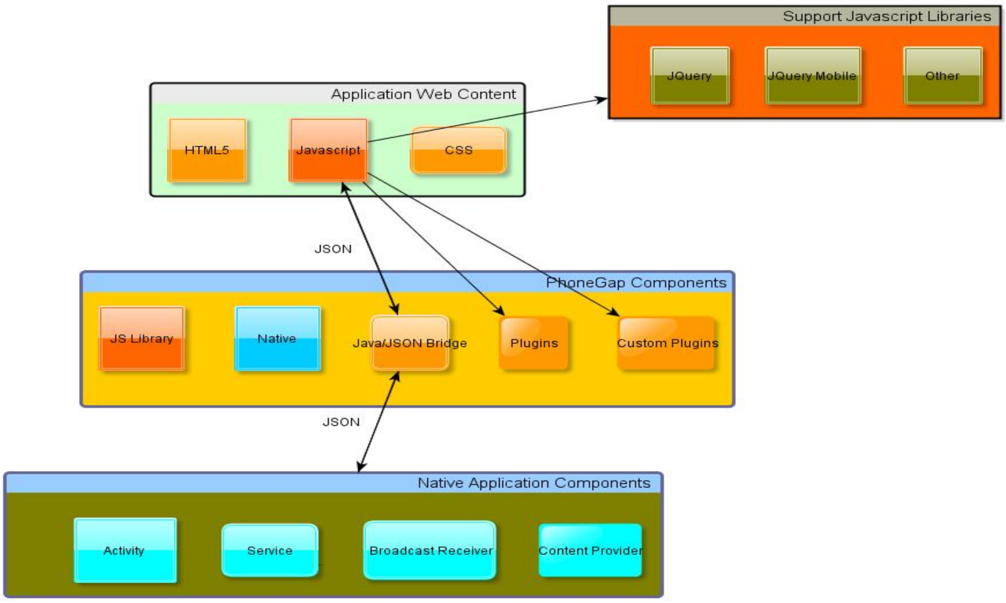


**Figure 6: Components of the Hybrid Web Application using Phonegap**

For other client devices, the user interface would be pure web based, a web application for standard browsers, with the web application either hosted from the cloud servers or on local devices, and thus supporting the rich client node type requirement from the disaster management scenarios.

The server side or cloud infrastructure was specified as a federated system distributed over various nodes. The target platform was decided to be Java based given the large availability of open source third party components and reusable software modules from project partners. The requirement of a supporting the dynamic deployment of 3rd party services was provided by the adoption of the OSGi compliant Spring Framework (7) enterprise application technology and the Eclipse Virgo implementation (8). These technologies provided an enterprise grade application server with the flexibility of creating light weight dynamic deployments. Being Java based the prototype solution was OS and cloud hosting independent, although cloud app solutions such as Google App Engine and Amazon WS were investigated.

### 2.2.2 Standards, Protocols & Interoperability

During the collaborative research tasks many standards and protocols were assessed and compared. The full details of the research and comparisons are documented in D4.1 'SOCIETIES Platform Specification and Design' (3). A comparison for suitable communication protocols was conducted between XMPP (9), AMQP (10), REST (11) and UPnP, but the XMPP technology stack was selected based on its maturity, wide deployment with existing systems and the additional features for identity authentication and extensibility. There was a requirement of the SOCIETIES platform to integrate with existing social network, so extensive research was conducted into open standards such as:

- OneSocialWeb because of its efficient scalability and federation based on the existing XMPP infrastructure
- BuddyCloud for its very careful dealing with groups of users and their associated permissions
- OpenSocial (12) as an application container that connects to a potentially federated social network backend

Other standards in use with the platform included OSGi for service definition and dynamic deployment, and HTML5 for web based interfaces.

## 2.3 Development Strategy

The development strategy for this large collaborative open source project, with ambitious objectives of bringing prototype software to end users, required the adoption of standard techniques and approaches best suited for the development of early prototype technologies. Although the wider project plan following a waterfall methodology, as discussed in section 2.1, the development plan adopted a more agile software development methodology. Key components of this approach included:

- Continuous integration and testing: From the start of the development phase, a central source repository, hosted on github.com, and central build/integration/testing server infrastructure was established. This allowed for nightly builds running the latest code merges, executing a series of unit, component, integration and performance tests, and a monitoring and notification facility to keep all development teams up to date on status, failures and issues. The technologies used included Maven as the build system, Jenkins as the build server, Junit and Cobertura for testing.
- Re-use of open source software: The re-use and adoption of existing open source software was attempted as much as possible, making sure that time was not unnecessarily spent re-implementing components that are already available and so that the development activities could be more focused on the implementation of the innovative features of the SOCIETIES platform.
- Release early and often: The SOCIETIES open source project provided source code and binary distribution releases for project developers, particularly the 3rd party service developers within the project, the user trial deployment teams, and external end users and 3rd party developers.
- Continuous User Feedback: User feedback was gathered at various stages in the project, during initial paper prototype trials and the end-user and developer software trials. The feedback was evaluated and fed back into the development teams in order for them to adapt and respond to the users' experiences.

### 2.3.1 Collaborative Development

Collaborative development in a large project with 16 organisations based in remote locations is a challenging task and it requires effective approaches and mechanisms to achieve the project's goals. In SOCIETIES, we adopted many approaches for communication on all matters, such as instant messaging, email, conference calls, and collaboration tools. In addition to these general communication tools, the development teams used other tools for sharing technical details and having discussions, such as a common repository for UML designs with *Enterprise Architect*, a common source code repository with *Github*, wiki and collaborative writing tools, and issue tracking and bug reporting features with *Redmine*. For group discussions and tech-transfer of ideas, we used a mixture of weekly conference calls with screen sharing, bi-monthly face-to-face developer and code camp meetings, and adhoc virtual meetings for internal tutorial sessions and issue solving.

For the duration of the project, 3 code repositories were utilized:

- **SOCIETIES-Platform:** included the code for core modules and modules related to individual and community experience
- **SOCIETIES-SCE-Services:** Hosts the service creation environment and all the 3rd party services that showcased the SOCIETIES Platform
- **SOCIETIES-Seed:** Used at the start of the project only for advanced prototyping of key concepts. These were used to kickstart the development cycle.

The development methodology followed 2 week sprint cycles of which the feature list was managed by Redmine. The following graphs show the level of contributions to each repo over the lifetime of the project:

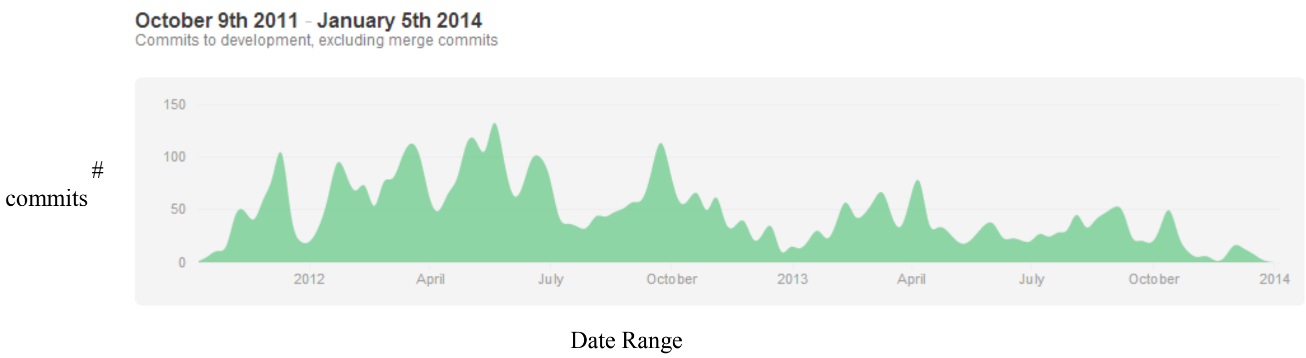


**Figure 7:** Code contributions to the SOCIETIES Platform Repo by date

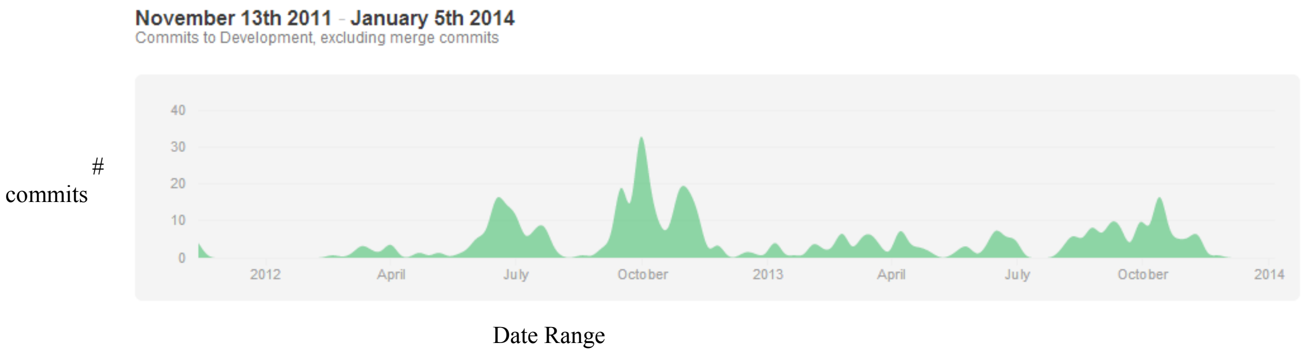


**Figure 8:** Code contributions to the 3rd Party Services Repo by date

 

# 3 Achievements & Challenges

This chapter aims to discuss the key achievements and challenges that occurred during the SOCIETIES project and where the role of Technical Coordinator provided input to the results.

## 3.1 Implementing SOCIETIES as an Integrated Software Platform

The objective of the SOCIETIES project was to prototype an integrated software platform, based on the project concepts and to satisfy the requirements from 3 distinct user groups, and then bring the prototype software to these user groups for evaluation. From the outset this was seen as a challenging objective because of two key aspects, the broad technical requirements of such a platform and that it would have to be achieved through the close collaboration of 16 partner organisations spread across Europe.

The project plan envisaged an open source integrated software platform that was to be extensible and adaptable to the requirements of various user scenarios and this was achieved with the SOCIETIES platform. The benefit of having the large extensible platform was that many of the pervasive and social features of the platform would not have to be duplicated in separate applications and various deployment scenarios could be more easily achieved by customising the platform with its extensible technologies. The project plan also envisaged bringing the prototype platform to different user groups at various times during the project to evaluate the concepts and software, and therefore provide feedback to the development teams. This was achieved but not to the extent as originally envisaged. The challenge of developing a large platform at the same as continuously providing end users with robust software for evaluation purposes was evident during the execution of the project and this impacted the ability to achieve the original plans for user trial evaluations. A different approach for a more rapid development of distinct software features as standalone components would have been better for achieving the original user trial objectives, although this approach was adopted for the development of the 3rd party applications for the user trials.

The collaborative design and development of the SOCIETIES platform was a challenging task but by using good development practices and tools, it was possible to overcome this challenge and achieve good results. Agreeing on technologies in collaborative projects is always a difficulty, but the approach taken was to adopt the best open source technology solutions that would meet the technical requirements, enable re-use of existing software components and provide interoperability with other platforms and applications. Varying skill levels and development techniques across the project partners also impacted the collaboration effort.

 

## 3.2 Open Source Releases

The SOCIETIES open source project was created in the early part of year 2 with the common source code repository hosted on github.com (13). SOCIETIES have published 12 open source releases which included the full source code branch of the platform, distributions for the Virgo and Android builds, and documentation for API usage, installation and platform integration. The collaborative process of each release required various testing stages from nightly builds to dedicated testing source branches. Deployment to the trial sites was also part of the release testing process and allowing the development teams to discover issues related to these varying deployments. Given the large size of the code base and the numerous remote development teams, it was difficult to produce clear and consistent documentation targeted for the end users and 3rd developers, and this was provided as feedback during the 1st student developer trial.

## 3.3 1st User Evaluation

During the 1st round of user evaluations, which occurred from January to April 2013, the project achieved 2 end user evaluations with the Enterprise and Disaster Management user groups, and 1 developer evaluations with a group of student developers at HWU in Edinburgh. As already discussed, the challenge of supporting these user evaluations with stable software while proceeding with the ongoing development of the SOCIETIES platform was difficult and this was reflected in the received user feedback. Although the technical issues did not fully restrict the evaluation process and, with focused technical and trial management, much valuable feedback was gathered using various evaluation techniques depending on the specific trial. The results, achievements and challenges are documented in detail in the deliverable D8.5 ‘First Prototype Evaluation Report’ (14).

## 3.4 2nd User Evaluation

The 2nd round of user evaluations occurred in the second half of 2013. HWU hosted a 6 week developer evaluation with the goal of assessing the SOCIETIES platform and service innovations from the point of view of a student user community consisting of early adopters of new technology. ICT2013 provided the venue for the Enterprise trial which was ran over 3 days with 50+ users signing up. The conference attendees installed a companion android application that allowed them to discover relevant events and exhibits, connect with communities of interest and organise ad-hoc meetings. The main difference this time was an "in the wild" aspect to the evaluations where users could assess the applications following a non-controlled flow. Any questions either in person or via electronic media were logged to form a comprehensive view of what the users experienced during the trial. Feedback from both trials was captured

No major technical difficulties were encountered, with minor patches being deployed on the fly as issues occurred. The feedback of these trials including the results, accomplishments and the challenges encountered are documented in detail in the deliverable D8.9 'Final Evaluation Report' (15)

# 4 Conclusion

The objective of the SOCIETIES project was to prototype an integrated software platform, based on the project concepts and to satisfy the requirements from 3 distinct user groups, and then bring the prototype software to these user groups for evaluation. This was a challenging objective and it required strong technical management to keep the project on track to achieving its results. There were many revisions to the project plan, which can be expected for a 42 month long project, and parts of the project plan were not achieved as originally envisaged. In particular, the level of user engagement and opportunities for evaluation was impacted by other project delays, although there were many good results achieved here due to a lot of effort committed by various project members. The design and development phases of the project were very challenging but also successful, given the project structure and natural complexities of a large collaboration project with partner organisations of different approaches. The role of Technical Coordinator was to manage the technical activities of the project and ensure the project objectives were met by the work packages as much as possible. This was achieved through the adoption of a unified development methodology, good communication, hands-on management of various activities and a strong team effort from many project members.

 

# 5 List of Abbreviations

| | |
|---|---|
| AmI | Ambient Intelligence |
| PSS | Personal Smart Space |
| CSS | Co-operating Smart Space |
| CIS | Community Interaction Space |
| SNS | Social Network |
| CRA | Collaborative Research Area |
| XMPP | Extensible Messaging and Presence Protocol |
| AMQP | Advanced Message Queuing Protocol |
| REST | Representational state transfer |
| UPnP | Universal Plug and Play |
| XML | Extensible Mark-up Language |
| API | application programming interface |
| UI | User Interface |
| OSGI | Open Services Gateway Initiative |
| CSS | Cascading Style Sheets |
| 3P | Third Party |
| GUI | Graphical User Interface |

# 6 References

1. **PERSIST.** PERSIST Project Website. [Online] http://ict-persist.eu/.
2. **SOCIETIES.** Project Glossary. [Online] http://www.ict-societies.eu/glossary/.
3. —. *D4.1 SOCIETIES Platform Specification and Design.* 2011.
4. —. *D4.3 SOCIETIES Platform Specification and Design, Revised.* 2013.
5. —. *D3.3 System Architecture.* 2013.
6. *Phonegap.* [Online] http://www.phonegap.com/.
7. **Spring.** Spring Source DM. [Online] http://www.springsource.org/osgi.
8. **Eclipse.** VIRGO. [Online] http://www.eclipse.org/virgo/.
9. **xmpp.org.** XMPP Website. [Online] http://xmpp.org/about-xmpp/.
10. AMQP Website. [Online] http://www.amqp.org/.
11. REST definition. [Online] http://en.wikipedia.org/wiki/REST.
12. **OpenSocial.** OpenSocial Social Data Specification 2.5.1 DRAFT. *http://opensocial-resources.googlecode.com/svn/spec/trunk/Social-Data.xml.* [Online] 2013.
13. **SOCIETIES.** SOCIETIES Platform Open Source Repository. *github.com.* [Online] https://github.com/societies/SOCIETIES-Platform.
14. —. *D8.5 First Prototype Evaluation Report.* 2013.
15. —. *D8.9 "Final Evaluation Report".* 2014.