SmartSociety

Hybrid and Diversity-Aware Collective Adaptive Systems
When People Meet Machines to Build a Smarter Society

Grant Agreement No. 600584

Deliverable 8.2 Working Package 8

Platform Prototype: Early Results and System Design

Dissemination Level (Confidentiality):[1]	*CO*
Delivery Date in Annex I:	*31/12/2014*
Actual Delivery Date	*December 15, 2014*
Status[2]	*D*
Total Number of pages:	??
Keywords:	*Platform, Prototype, Integration*

[1] PU: Public; RE: Restricted to Group; PP: Restricted to Programme; CO: Consortium Confidential as specified in the Grant Agreeement

[2] F: Final; D: Draft; RD: Revised Draft

Disclaimer

This document contains material, which is the copyright of *SmartSociety* Consortium parties, and no copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights. The commercial use of any information contained in this document may require a license from the proprietor of that information. Neither the *SmartSociety* Consortium as a whole, nor a certain party of the *SmartSociety*s Consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information. This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.

Full project title:	SmartSociety: Hybrid and Diversity-Aware Collective Adaptive Systems: When People Meet Machines to Build a Smarter Society
Project Acronym:	SmartSociety
Grant Agreement Number:	600854
Number and title of work-package:	8 Architecture and Integration
Document title:	Platform Prototype: Early Results and System Design
Work-package leader:	Iacopo Carreras, UH
Deliverable owner:	Iacopo Carreras, Daniele Miorandi
Quality Assessor:	Luc Moreau, SOTON

List of Contributors

Partner Acronym	**Contributor**
UH	Daniele Miorandi, Iacopo Carreras, Tommaso Schiavinotto

Executive Summary

This report briefly describes the context and components of the first version of the SmartSociety platform.

The SmartSociety platform integrates the instantiation of the components designed and developed by the Consortium members within the framework of the technical workpackages WP2-WP7.

The report at hand includes a set of revised architectural diagrams, developed by the Consortium building upon the results in Deliverable D8.1. The revised architecture is now fully aligned with the specification of components as developed within WP2-WP7, and has been developed according to the high-level requirements identified in WP1 and further on analysed in D8.1.

The report then describes the interfaces of the platform components integrated up to M24 (the peer manager, the orchestration and negotiation manager and the communication middleware). The software produced is then presented.

The report concludes with a roadmap detailing the integration milestones for the third year of project activities.

Table of Contents

List of Acronyms

Acronym	Full Name	Description
EC	Evaluator Component	System component in charge of evaluating the outcomes of each computation task (Sec. ??).
IM	Incentives Manager	System component in charge of managing the implementation of incentive schemes (Sec. ??).
KB	Knowledge Base	System component in charge of storing and managing the knowledge in the platform (Sec. ??).
OC	Orchestration and Coordination	System component in charge of coordinating the orchestrating the execution of SmartSociety programs (Sec. ??).
PDE	Peer Discovery Engine	System component in handling peers querying and discovery (Sec. ??).
PF	Programming Framework	System component in charge of the interface between the SmartSociety platform and developers (Sec. ??).
PM	Peer Manager	System component in charge of managing peers (Sec. ??).
TCO	Task Coordination and Orchestration	System sub-component in charge of coordinating the execution of a single task (Sec. ??).
TEM	Task Execution Manager	System component in charge of managing the execution of a single task by external peers (Sec. ??).
WE	Workflow Engine	System component in charge of managing the execution of a program workflow (Sec. ??).

1 Introduction

This report is a short accompanying document whose aim is to describe the platform components developed and integrated within a coherent platform by the Consortium partners during the second reporting period. The actual deliverable is a prototype of the SmartSociety platform, which can be found at: . . .

Starting from the analysis of requirements and initial platform architecture in D8.1, WP8 undertook an intense dialogue with technical workpackages (WP2-WP7) in order to ensure full alignment of the top-level platform architecture and of the scientific and technical outcomes of the single WPs. The results of this first phase was a revised architecture for the SmartSociety platform, which is presented in Sec. 2. According with the progress of the technical WPs, interfaces specifications were defined, which are summarised in Sec. 3. During the second half of the year the actual integration of components started, resulted in a 'minified' version of the platform which is described in Sec. 4. The missing components will be integrated during the third year according to the roadmap outlined in Sec. **??**.

2 The SmartSociety Platform Revised Architecture

During the second year of project activities integration activities started. In the integration process, the architecture initially presented in Deliverable D8.1 was deeply revised. This process was required in order to align the activities carried out within the project's technical work packages (WP2-WP7) and to ensure interoperability among the components developed by the various partners. In this section we briefly present the architecture as it currently stands (at M24). No major changes are currently foreseen, even if —given the research-oriented nature of the SmartSociety project— this cannot be guaranteed. In this sense, the architectural specifications of the SmartSociety platform have to be seen as a live document, which reflects the actual progress of the research activities carried out by Consortium partners.

2.1 Logical View

A perspective view of the SmartSociety platform is reported in Fig: 1.

Two types of applications interact with the platform: User Applications, which provide a service to (human) end users, and Peer Applications, which are used to connect to peers (human or machine, individuals or collectives) as a resource.

User applications comprise both a client and server-side component. The server-side component is considered part of the platform. Applications need to be registered with the platform to access its functionality.

Peers (be them human or machines) need to be registered on the platform to be able to provide resources, competences and services.

External sources and external services may interact via the SmartSociety platform via appropriate connectors.

A functional diagram of the SmartSociety platform is reported in Fig: 2. The Consortium has identified 9 key components which jointly provide the required functionality:

- Knowledge base:it contains an agreed ontology about peers, task, workflows, incentives, etc. and it is used for ensuring semantic interoperability among the other platform components.
- Provenance store: it logs actions performed by platform components and peers according to the W3C PROV recommendation. It further supports an auditing service which is able to reconstruct and visualize provenance trails.

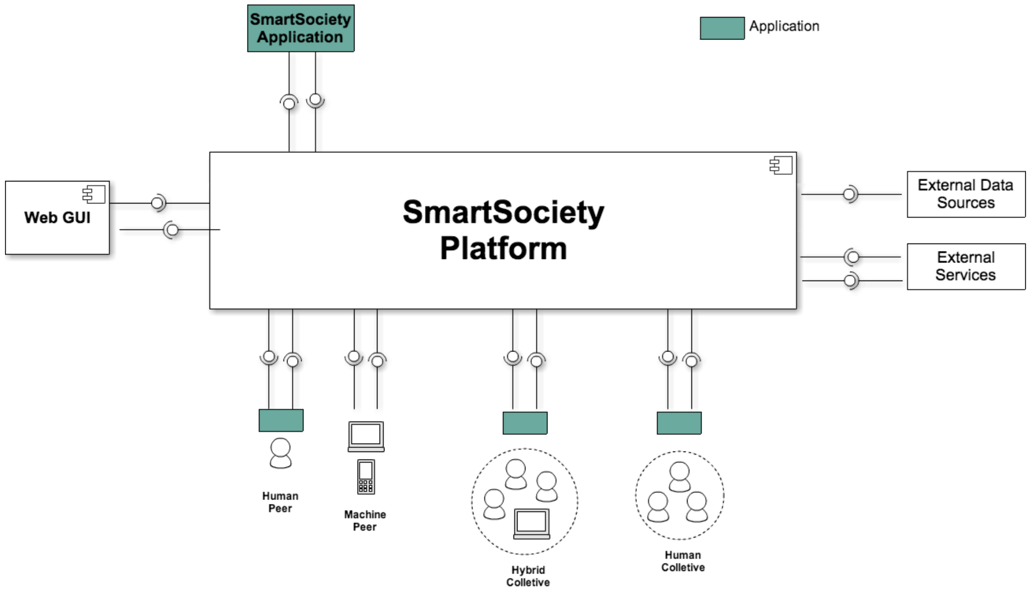

Figure 1: Perspective view of the SmartSociety platform.

- Peer manager:it is in charge of managing the peers that are part of the SmartSociety system. It maintains a profile of each peer, which represents a model thereof in terms of knowledge, resources and capacity. It provides a peer search functionality that allows other components to find for the most appropriate peers for a given tasks.

- Context manager: it dynamically monitors the context the human peers are in (in terms of, e.g., location, activity etc.). This information is fed to the peer manager and represents the dynamic part of the peer profile.

- Incentives manager: it provides insight on incentives and interventions that can be used to achieve higher quality results.

- Orchestration and coordination manager: it provides two key functionalities: composition and negotiaton. Composition takes as inputs tasks and interacts with the peer manager to find suitable peers for completing the task. The negotiation manager is in charge of handling the negotiation process with peers in order to ensure that the necessary services and resources required to carry out the task can be guaranteed.

- Reputation manager: it computes the reputation of a given peer based on feedback from users. It uses data from the provenance store in order to carry out the computation.

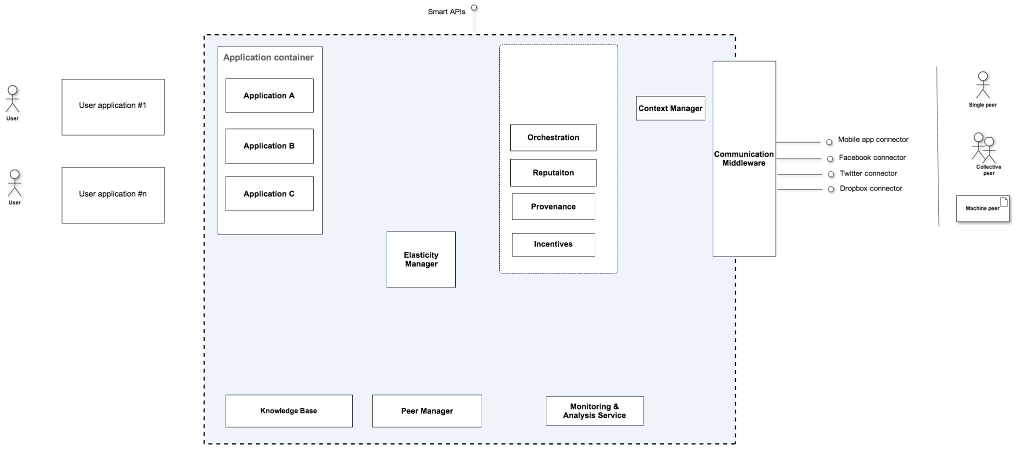

Figure 2: Functional diagram of the SmartSociety platform architecture.

- Monitoring and analytics service: this service logs and monitors the platform jobs and can be used by platform administrators to perform root-cause analysis and to extract analytics on the performance of the system.
- Elasticity manager:
- Communication middleware:

2.2 Deployment View and Network Diagrams

The SmartSociety platform has been designed around a REST architecture with the aim of supporting flexibility in the deployment model. This means that the platform shall seamlessly support both single-tenant as well as multi-tenant deployment models. Also, the platform component can be centralised on a single infrastructure or can be distributed across different servers. The choice of the specific deployment model to be used depends on technical as well as business considerations. In the remainder of this section we present, as a use case, the current deployment utilised for integration, testing, validation and experimentation purposes. This is by no means to be considered the only model supported, but it provides an actual example of the supported configuration.

2.3 Dynamic View

The SmartSociety platform is meant to support a rather wide range of social computation patterns (or templates). In order to provide insight into the flexibility of the platform

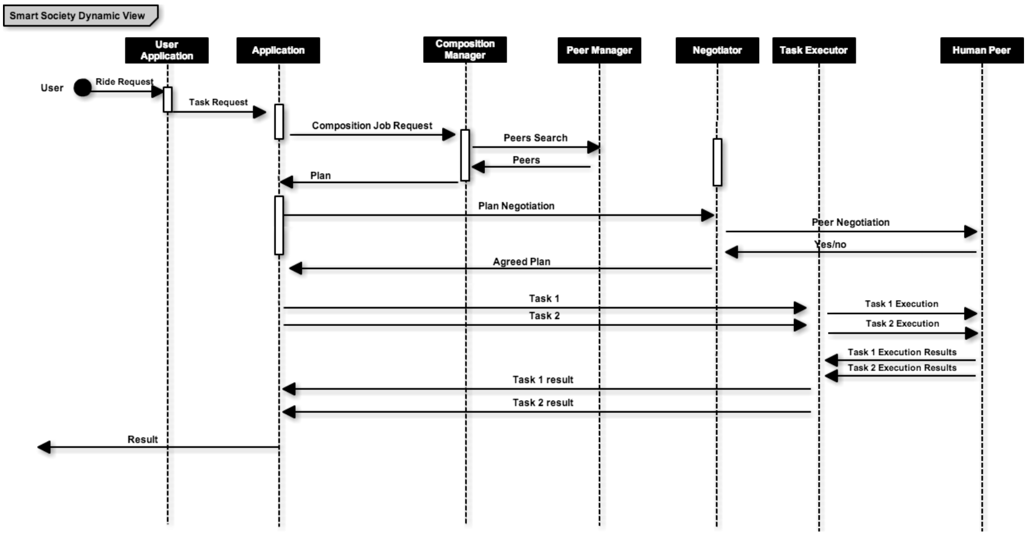

Figure 3: Sequence diagram of the SmartShare application.

and the actual interworking of components, we have developed sequence diagrams for two 'extreme' applications:

- SmartShare is a ridesharing system able to account for user's preferences and to compute recommendations based on the feedback provided by other service users. It is what we call a 'full negotiation' scenario, in which the computational task of finding an agreement on the rides is left to individuals and collectives. The platform in this case is used to carry out administration tasks, in particular keeping track of the rides and ride requests, their status and to maintain reputation of drivers and passengers.

- AskSmartSociety! is a Q&A service supporting hybridity. The computational pattern here is that typical of micro-tasking applications (á la Mechanical Turk, roughly speaking), where the task in this corresponds to a question to be answered. The service supports hybrid computation in that questions can be transparently provided by machine peers or human peers. Quality criteria can be specified in order to define when a chosen answer has to be presented to the user.

In the following we present details about the two aforementioned applications.

2.3.1 Example: SmartShare

2.3.2 Example: Ask SmartSociety!

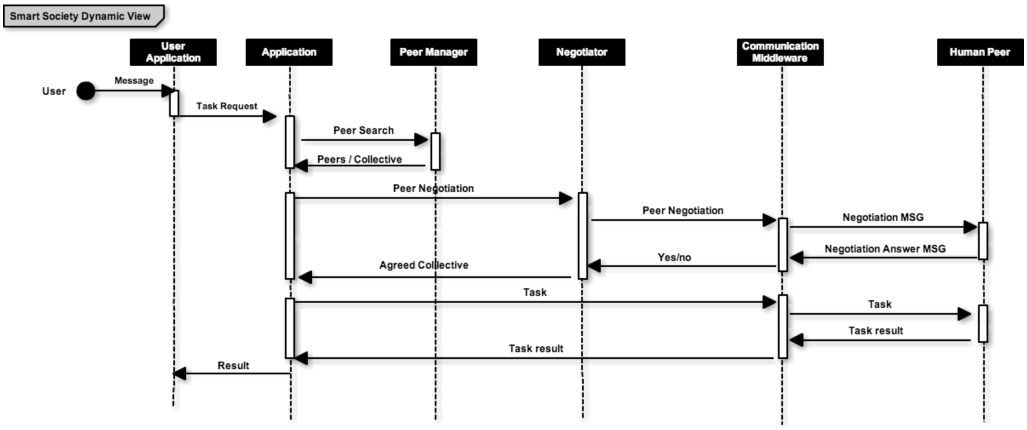

Figure 4: Sequence diagram for AskSmartSociety! applications.

3 Interfaces Specifications

In this section we will list the components integrated at the moment and the API subset used.

3.1 Peer Manager

The Peer Manager (PM) API specifications can be found at `http://demos.disi.unitn.it:8080/smartsociety/`. The PM is used for:

- **Retrieving peers satisfying some requirements**: This request is made by the Orchestration Manager. Endpoint used:
  `???`
- **Creating collective**: This request is made by the Application/Application Runtime at the end of the negotiation. Endpoint used:
  `/peers/collectivePeer` (POST)

3.2 Communication Midlleware

The SmartCom Communication Middleware APIs specifications can be found at url-https://github.com/tuwiendsg/SmartCom and are further described in Deliverable D7.1. The Communication Middleware is mainly used for:

- **Passing messages intended for collectives through a variety of communication channels**: This request is made by the application *TBC*. Endpoint used: at.ac.tuwien.dsg.smartcom.Communication

- Do we need the other functionality? I guess not...

3.3 Orchestration Manager

The Orchestration and Negotiation Manager APIs specification can be found at https://bitbucket.org/rovatsos internal/wiki/PeerManager/APIBGU and are further described in Deliverable D6.2. The OCM is mainly used for:

- **Posting a new task request**: In order to post a task request the following endpoint is used:
  `/applications/:app/taskRequests` (POST)
- **Compose collectives adequate for fulfilling a given task:** this is carried out by the composition manager (see D6.2) and is executed every time a task request is posted in the system. Endpoint used: `/applications/:app/compositions` (POST)
- **Negotiate with peers to create an agreed plan:** it negotiates with peers in order to have their explicit agreement for carrying out a given task. It returns a plan. Endpoint used: `/applications/:app/negotiations` (POST)

4 Description of Artifacts

here goes description of single components (functionality and technologies used for building it), where it can be found, how it can be installed/called etc.

5 Discussion and Furhter Integration Steps

Summarizes the content of the deliverable and describes the integration steps that will be carried out in Y-3 (integration roadmap).

References