SEVENTH FRAMEWORK PROGRAMME

THEME ICT-2009.1.2

“Internet of Services, Software and Virtualization”

D4.1
Integration Plan

Project acronym: SocIoS

Project full title: *Exploiting Social Networks for Building the Future Internet of Services*

Contract no.: 257774

Workpackage:	WP4	Integration and Testing
Editor:	P. Brigden	ATC
Author(s):	P. Brigden	ATC
	I. Spais	ATC
	S. Porat	IBM
Authorized by	D. Varvarigou	ICCS/NTUA
Doc Ref:		
Reviewer	F. Aisopos	ICCS/NTUA
Reviewer	A. Polonsky	Cognium
Dissemination Level	PU	

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 257774

SocIoS Consortium

Beneficiary Number	Beneficiary name	Beneficiary short name	Country	Date enter project	Date exit project
1(coordinator)	Institute of Communication and Computer Systems/National Technical University of Athens	ICCS/NTUA	Greece	Month 1	Month 30
2	IBM Haifa Research Lab	IBM	Istrael	Month 1	Month 30
3	Athens Technology Center	ATC	Greece	Month 1	Month 30
4	Google Ireland Limited	Google	Ireland	Month 1	Month 30
5	Cognium Systems	Cognium	France	Month 1	Month 30
6	Center for the Study of the Information Society, University of Haifa	HU	Israel	Month 1	Month 30
7	Deutsche Welle	DW	Germany	Month 1	Month 30
8	Stefi Productions S.A.	Stefi	Greece	Month 1	Month 30
9	Katholieke Universiteit Leuven (K.U.Leuven) – Interdisciplinary Centre for Law and ICT	KULeuven	Belgium	Month 1	Month 30

Document History

Version	Date	Changes	Author/Affiliation
v.0.1	30-06-2011	Add content	ATC
v.0.2	15-07-2011	Add content	ATC
v.0.3	14-08-2011	Internal ATC review	ATC
v.0.4	05-09-2011	Prefinal version distributed for internal review	ATC
v.0.5	06-09-2011	NTUA review	ICCS/NTUA
v.0.6	06-09-2011	Cognium review	Cognium
v.0.7	06-09-2011	Incorporation of comments/suggestions	ATC
v.1.0	06-09-2011	Final version distributed	ATC

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 257774

Executive Summary

The SocIoS system, far from being a simple container for the individual modules, is a coherent platform, where several different components reside and collaborate in harmony. The prototype version of the system is a proof of concept to the target users and stakeholders. It encapsulates all underlying technologies and gives a clear and easy to use graphical interface, exposing all the main features.

The benefit of the current architecture is that any additional functionality can be wrapped into a separate component and be added to the system, provided that it abides by the basic communication standards exposed by the SocIoS architecture. The scope of this practice is to keep the system evolving and enable future extensions to the supported functionalities, which may maximise the potentials for further exploitation of the system beyond the project end.

This document is the integration plan of SocIoS project. It describes the integration of the SocIoS middleware services that will be implemented by WP2 with the toolset services delivered by WP3.

The integration plan defines internally all steps of the technical coordination and verifies the time-line of the activities, the responsibilities and the final decisions on technology selection issues. Furthermore, the non-functional requirements of the SocIoS platform (such as security, availability, scalability, etc) were defined, according to the overall design and the service developers' needs and preferences. The main technical work, which is the main action conducted in this report, incorporates a mixture of customization and adaptation of existing services and toolsets and integration of all modules in two (intermediate and final) platform instances.

After summarizing the SocIoS context, the report presents the SocIoS architecture focusing on the physical deployment of the component and on the communication interfaces that each one has defined. Based on the latter, a logical diagram has been created and a number of workflow diagrams presenting SocIoS functionalities and capabilities graphically.

The SocIoS hardware facilities and some important installation guidelines are also mentioned. Finally, the deployment and configuration of the SVN environment as a central versioning repository as well as the usage of a bug-tracking system are also described in details.

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 257774

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 257774

List of Figures

List of Tables

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 257774

Terms and Abbreviations

SN	Social Network
IoS	Internet of Services
SOA	Service Oriented Architecture
SNs	Social Network Sites

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2011] under grant agreement no. 257774

1 Introduction

This deliverable reports on the activities and effort placed in the integration of the various technologies and tools into a functional SocIoS system. It compiles a detailed plan on how the individual services and components will be placed together to constitute the 1st (intermediate) integrated SocIoS prototype, addressing the requirements defined in [1].

The SocIoS project aims to deliver a system for aggregating content posted in the most well-known Social Networks, namely Facebook, Twitter, Youtube, etc., and providing the end users with capabilities of building applications to exploit them. In this way, SocIoS will pave the way for building qualitative, functional and usable business applications and provide an added value towards the realization of the Future Internet of Services.

In concrete terms, the project produces a system that integrates the following technological advances:

- A SOA infrastructure that supports the deployment of a collection of services that constitute SocIoS capabilities and functionalities and exposes to the end-users a workflow enactment mechanism to handle them (use them as they are, combine them into new workflows, compose new services, etc.).
- The SocIoS API, which aggregates the methods of the underlying SNs APIs and provides the developers a single access point to the data (or actions) that are hosted in the SNs.
- The SocIoS middleware, which supports the SocIoS platform in executing fundamental SocIoS functionalities.
- A SocIoS toolset for supporting the business extensions, which allows the incorporation of quality notions in the SocIoS services while investigating all the implications that will occur from that concept.
- The user interface component that exposes SocIoS platform capabilities and functionalities to the end-users of the project.

All the above are implemented in WP2 and WP3 according to the Technical Annex of SocIoS. Considering their specifications and focusing on achieving an efficient integration that will exploit all the technological advances offered, this report compiles a detailed integration plan, which defines internally all steps of the technical coordination and verifies the time line of the activities, the responsibilities and the final decisions on technology selection issues.

Furthermore, the non-functional requirements of the SocIoS platform (such as security, availability, scalability, etc) are defined, according to the overall design and the service developers' needs and preferences.

With respect to project's pilots execution and to the technical developments needed, SocIoS consortium decided to incorporate a mixture of customization and adaptation of existing services and toolsets and integration of all modules in two (intermediate and final) platform instances. The integration prototypes will be tested and validated before distributed to the partners that are responsible for executing the pilot trials. The evaluation results of each iteration will be examined in details from the technical partners in order to implement any refinements needed.

Note that this document is aiming to provide the architectural description of the system and the guidelines for its implementation in terms of integrating the various components. The actual design of the implementation is subject to change during the various stages of integration, being adapted to the needs or the limitations of the system. Enhancements or other modifications are also possible to be applied to the system while the implementation takes place.

1.1 *Architectural principles*

Based on the needs of the SocIoS system, the architectural design is focusing on the following principles:

- **Simplicity**: The system has to be simple to understand and integrate. The collaboration of several partners for the same project imposes a restriction for high level of complexity. Moreover, the needs for long term maintenance make it necessary to follow a simple approach, not only for the system architecture, but for the internal implementation of each component as well.

- **Scalability**: The architectural approach for SocIoS project does not intend to impose limitations on the possibility to enhance and enrich the project with further functionality. Therefore during development we should bear in mind that the project is not finalized, on the contrary it may be subject to multiple changes for either optimization or compatibility purposes.

- **Decoupling of functionality and technology**: Since multiple partners are not using necessarily the same technology and platforms, SocIoS system needs to decouple logic from implementation. Contemporary architectural approaches are in favor of loose coupling between the functional components of large systems. Even if the solution described states that the independent components reside on the same machines, this should not be a prerequisite. The structural components of the system shall be totally independent and shall be able to act in a standalone and distributed way. Their communication will be based on the common file sharing approach while their synchronization will be controlled by a central component, no matter where the distributed components are hosted.

- **Flexibility**: Technological limitations for each partner have to be taken into account. For this reason, the system should have the ability to be flexible and easily overcome any limitation regarding bandwidth, processing capacity, security constraints and lack of resources. Possible failing scenarios shall be investigated and for every case, the system shall come up with a solution that overcomes the difficulties to provide the desired functionality. This means that modifications shall be easily applicable to the project without seriously affecting the initial design, and with reasonable effort and time.

1.2 Outline

This document is structured as follows. Section 2 summarizes SocIoS context by presenting briefly SocIoS objectives, the user groups and their roles, SocIoS use cases and functionalities and the non-functional requirements. Then, Section 3 draws logical and physical diagrams of the system's architecture and states the architectural principles for the creation of SocIoS system. It also mentions the communication framework that the project will adopt and the respective protocol. Next, Section 4 discusses the components of the SocIoS platform and based on their interfaces draws the SocIoS workflow diagrams. Furthermore, the hardware facilities are described and some important installation issues. The section ends by presenting the SVN environment that is used as a central versioning repository and a supplementary bug-tracking component. The test cases of the intermediate integrated prototype are provided in Section 5. Finally, Section 6 overviews the time-line integration activities and Section 7 concludes the document.

2 Summary of SocIoS context

Starting from the core SocIoS objectives, this section summarizes the main characteristics of the SocIoS architecture, addressing the use cases, user needs, system requirements the functional specifications and the non-functional requirements. A detailed description can be found in [1].

2.1 Overview of SocIoS objectives

The main goal of the SocIoS project is to Exploit Social Networks for building the future Internet of Services. SocIoS is a framework for allowing application developers of variable level of expertise to combine content and services from a wide range of SN sites into complex workflows. The approach for services development and deployment is "write once, run everywhere". The resulting services can be deployed on the developer's own server or inside the SocIoS platform. There they can be discovered and used by other users of the platform. The terms of service usage, including billing information, are defined by Service Level Agreements (SLA service). The SocIoS framework provides the functionality of developing applications that adheres to all the related regulation, be it national, or enforced by the individual platforms. Overall SocIoS aims to support:

- the full exploitation of the main SN assets namely, user-created content (UCC) and social graphs,
- the SocIoS user -be it individual or organization- to easily develop, deploy, share, exploit and configure applications and services,
- the interoperability, the sustainability and the extendibility of the platform,
- the integration of the developed applications to a service provisioning market.
- the viability of the SocIoS platform through the adherence to the respective regulations and privacy protection.

2.2 User groups and roles

SocIoS has four main goals and target groups. It helps the consumers of the SocIoS application to efficiently exploit the assets that live in SNs for their everyday work. It exposes to application developers (organizations or individuals) with various knowledge of software development for the SocIoS services, in order to manipulate them and create new or updated SocIoS applications. It provides service developers with a good understanding of how the SocIoS API works, to create new SocIoS Core Services. It enables the SNs users to actively (e.g. upload a photo for a SocIoS Service to get it) or passively (i.e. allow access to their existing details to SocIoS platform) providing data.

Based on these considerations and with respect to their roles and responsibilities, the potential users of the SocIoS system are divided into the following categories:

- Administrators
- End Users

The administrator role has full access rights in the system and is able to manage users and assign access rights via SocIoS AAA module, while configuring the main setup parameters defined by each of SocIoS individual components and maintaining the platform as a whole integrated operational prototype.

End users include all interested organizations or individuals that may access the SocIoS system to assist them in their everyday activities for monitoring the news and associating them with their plan of action. Potential end users lay on the following categories (detailed description in [1]):

Table 1. Users of SocIoS platform

User group	Description	Example
SocIoS End Users	The consumers of the SocIoS Applications. Their requirements may vary because their purpose of using the SocIoS applications may also vary. For example, a domain expert using a SocIoS application can be operating under different framework because she is a freelancer than another domain expert who is working for an organization.	DW journalist or Stefi producer
SocIos Application Developers	Organizations or individuals with variable knowledge on software development that will use the SocIoS Services to create new SocIoS Applications.	DW or Stefi's New Media/IT department
SocIoS Service Developers	Developers with a good understanding of how the SocIoS API and middleware work, that use the SocIoS API so as to create new SocIoS core services.	Developer community
SNS Users	Anyone who allows the SocIoS Services to extract data from their accounts in the underlying SNSs. SNS users make their profile details or their content available to SocIoS Services. There might be SNS users that actively provide data (e.g. upload a photo for a SocIoS Service to get it) and those who passively provide data, i.e. they simply allow access to their existing details to SocIoS services.	John Doe

2.3 *SocIoS use cases and functionalities*

In the context of SocIoS, we performed a study in order to capture the user needs of all the aforementioned user groups and record practices and problems of every group. The following section summarizes the results of the user study – use cases and supported functionalities – and a detailed description can be found in [1].

2.3.1 Use cases – supported functionalities

This section recalls the scenario use cases, first introduced in [1] and summarised on Table 2, and refers to the functionalities that can be realised from an end user perspective. These functionalities wrap up the end user requirements and the use cases and are partially or fully supported in the intermediate and final versions of the integrated prototype [2], reflecting the necessary components needed in order to realise the SocIoS objectives and drive the technical developments.

Table 2. SocIoS envisioned use cases and functionalities

Use case	Functional requirements	Priority	User Group	Supported in the intermediate prototype	Supported in the final prototype
Register	Sign-up	High	SocIoS End User	☑	☑
	Authentication	High		☑	☑
Login	Login	High	SocIoS End User	☑	☑
Search SNS users	Search SNS Users	High	SocIoS End User	☑	☑
	Location-based search	High		☑	☑
	Filter list	Medium		☑	☑
View details of SNS users	SNS user profile info	Medium	SocIoS End User	☒	☑
	SNS user reputation	High		☑	☑
Manage groups of SNS users	Monitor SNS Users	High	SocIoS End User	☑	☑
	Group SNS Users	High		☑	☑
Contact SNS users	Contact SNS users	High	SocIoS End User	☑	☑
Search Media Items and activities	Search MediaItems (and Activities)	High	SocIoS End User	☑	☑
	Location-based search	High		☑	☑
	Identify originality of media items	Low		☒	☒
View details of Media Items and activities	View MediaItem	High	SocIoS End User	☑	☑
	See relevant metadata	High		☑	☑
	Translate	High		☑	☑
Request relevant SNS users	Request SNS users relevant to a topic	High	SocIoS End User	☑	☑
	Request SNS users relevant to other SNS Users	High		☑	☑
Monitor Events	Monitor event	Medium	SocIoS End User	☑	☑
	View notification	High		☑	☑
Purchase Media Item	Clear IPR	Medium	SocIoS End User	☒	☑
	Purchase Media Item	High		☑	☑
Identify eye-witness accounts and request input	Set-up crowdsourcing game	Medium	SocIoS End User	☒	☑
Identify Media items of similar concept	Set-up crowdsourcing game	Medium	SocIoS End User	☒	☑
Register new service	Service registration and Subscription	High	Service Developer	☒	☑

Negotiate service price	Negotiate service price	Medium	SocIoS End User	☒	☑
Create workflow (applications)	Create an application workflow	Medium	Application Developer	☒	☑
Manage SNS user data	Allow SocIoS to access data	Medium	SNS user	☒	☑
	Sign consent form	High		☒	☑
	Access, rectification and removal - blocking of personal data	High		☒	☑
Use an application	Invoke an application	Medium	SocIoS End User	☒	☑
Set up Auction	Purchase Media Item	High	SocIoS End User SNS user	☒	☑
	Contact SNS users	High		☒	☑

2.4 Non-functional requirements

The non-functional requirements that should be considered during the development of the SocIoS platform prototype are presented in the table below.

Table 3. SocIoS non-functional requirements

Non-functional requirement	Description
Capacity	How many concurrent users need to be supported by the system (network capacity). Storage capacity, how much content will be stored and processed. The database(s) containing the material to be analyzed must be very well equipped as it will store a great amount of data
Back-up	The system offers back-up options for the content
Security	Single authentication process is required for accessing the functionality of the system. The system should be secured against sabotages arising from all types of hacking attacks. It should ensure secure interactions between the end users of all the categories
Scalability/Expandability	The system must be able to run on any of the major modern hardware platforms and operating systems. Specifically, it must run on Windows, Linux or Solaris operating systems and any hardware architecture that is supported by these operating systems. Moreover, it should be able to retrieve items from the most common Database Management Systems (e.g. SQL Server, Oracle, mySQL, etc). Furthermore, the system must be horizontally and vertically scalable. After the architecture has been finalized, all components should be upgradeable by adding more hardware resources.
Availability	The system should ensure that users have always access to data and associated assets 24/7 with 99.9% reliability. This requirement entails stability in the presence of localized failure
Usability	Easy to use. User documentation should not be necessary for ordinary tasks
User Interface	Should give access to all system functionalities providing easy navigation through all features
Deployment	Distributed components communicate based on a service-oriented architecture
Fault tolerance	The system must handle failure of individual components or infrastructure effectively and in case of failures that cannot be recovered, user friendly messages will be provided to the user. Transactional integrity poses an important challenge and should be taken into account when designing the integrated system and the individual components. Specifically the system should be able to recover failures of any of the individual components without suffering loss of data
Interoperability	Individual components should be able to exchange information and use the information exchanged

Performance	Performance should be defined as the response time of the user interface to the users under a certain load, i.e. affecting the QoS characteristics of the system. Performance should be acceptable, provided that the system and network load is normal. The service oriented architecture will pose an extra performance cost, depending on the amount of data exchanged between the systems, since the individual functional components will not be bundled in a standalone application, but will communicate over web services. What should be considered is keeping large volume data exchange at a minimal level
Stability	Under heavy load the system implementation should ensure robustness, availability and error-handling

3 Architecture Overview

This section describes the way that the various components of the system will communicate with each other in order to retrieve and store data produced during the operation of the whole system. It also defines the way that the work performed in each WP is synchronized with the other components in order to accomplish the functionalities foreseen for the intermediate and final integrated prototype.

Based on [3], each partner provided all the technical details of the components they had developed, including information like the name, description, the technology used, installation instructions, description of the interfaces for communicating with each other etc. This information was finalised for the intermediate prototype that will be the basis of the first pilot execution and validation. The following sections briefly describe the core architectural concepts (utilization of SOA architecture and SOAP communication protocol in the context of the project) and diagrams, namely conceptual architecture, physical deployment and logical diagram, resulted from the technical partner's developments during the 1st year of the project.

3.1 Architectural concepts – communication framework

3.1.1 Service-Oriented Architecture (SOA)

Service-Oriented Architecture is widely used as an architectural pattern for contemporary systems development and integration. Its main principle is the separation of functionality into different units which interoperate with each other over a network. These units are usually developed as web services and are reused by the system in order to achieve the desired functionality. A 'Web service' is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network". The web services can interoperate by exchanging data with each other directly, or by being coordinated by a central mechanism of the system. In general, SOA focuses on loose coupling of services in terms of used technology and programming language. By using common standards, web services have little or no dependency on each other, a fact that offers flexibility and scalability to the system. SOA is considered to be the evolution of distributed systems architectural approach.

SOA is typically characterized by the following characteristics:

- **Logical view**. The service is an abstracted, logical view of actual programs, databases, business processes, etc., defined in terms of what it does, typically carrying out a business-level operation.
- **Message orientation**: The service is formally defined in terms of the messages exchanged between provider agents and requester agents, and not the properties of the agents themselves.

- **Description orientation**: A service is described by machine-processable meta-data. The description supports the public nature of the SOA: only those details that are exposed to the public and important for the use of the service should be included in the description. The semantics of a service should be documented, either directly or indirectly, by its description.
- **Granularity**: Services tend to use a small number of operations with relatively large and complex messages.
- **Network orientation**: Services tend to be oriented toward use over a network, though this is not an absolute requirement.
- **Platform neutral**: Messages are sent in a platform-neutral, standardized format delivered through the interfaces. XML is the most obvious format that meets this constraint.

The concept of SOA and Web Services is typically associated with the Simple Object Access Protocol (SOAP) due to wide adoption and industrial support for SOAP-based web services.

3.1.2 Simple Object Access Protocol (SOAP)

In the context of SocIoS integration, the SOAP protocol will be used.
SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of web services in computer networks[1]. SOAP is a messaging protocol between requestor and provider objects, allowing the requesting objects to perform remote method invocation on the providing objects in an object oriented programming fashion.
SOAP relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Remote Procedure Call (RPC) and Hypertext Transfer Protocol (HTTP), for message negotiation and transmission. It can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. This XML based protocol consists of three parts: an envelope, which defines what is in the message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing procedure calls and responses.

[1] http://en.wikipedia.org/wiki/SOAP

3.2 Architectural diagrams

3.2.1 SocIoS conceptual architecture

Figure 1 presents an overview of SocIoS functionality, as it was presented in [3]. On the lower level are the Social Networks and on top of them the SocIoS core services, providing a level of abstraction on top of the heterogeneous Social Network APIs. The SocIoS API has its own generic object model encapsulating the entities and relationships residing in the underlying SNS. For each supported SN API, an adaptor has been implemented transforming SN data to the SocIoS object model, allowing the API consumers to access data from each underlying SN in a uniform way. In the context of the project, the SNs supported by the API are YouTube, Twitter, Flickr, Facebook and the SNS acting as OpenSocial containers such as hi5, LinkedIn and MySpace.

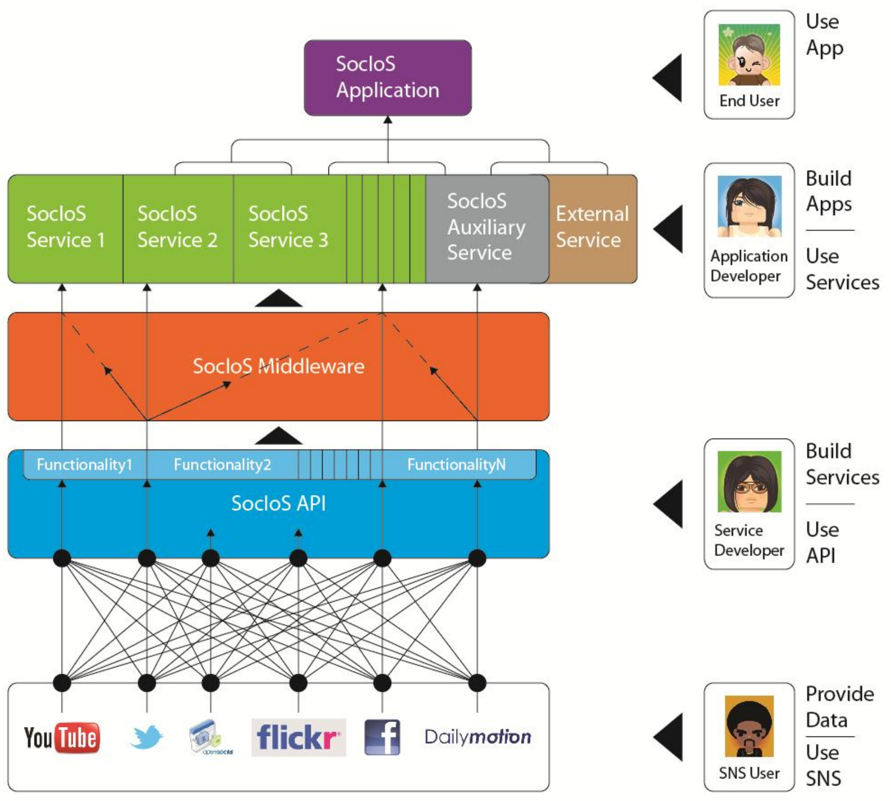


Figure 1. SocIos conceptual architecture

3.2.2 Physical deployment

With respect to partner's specifications for each component that will be integrated in the SocIoS platform, the physical diagram demonstrates the finalized system, as it will be set up in practice. As presented in the following figure (Figure 2), four different servers will be utilized for the whole system hosted by ATC and IBM. It should be mentioned that this physical diagram does not impose any restriction to the system. On the contrary, the system could be distributed among more servers communicating over the world wide network. However, we choose to divide the system into as few machines as possible for maintenance, stability and easier integration purposes. It should be noted that the interfaces provided remain the same, no matter if the components are located into the same server or not. With this approach we minimize the risk of communication problems due to network difficulties.

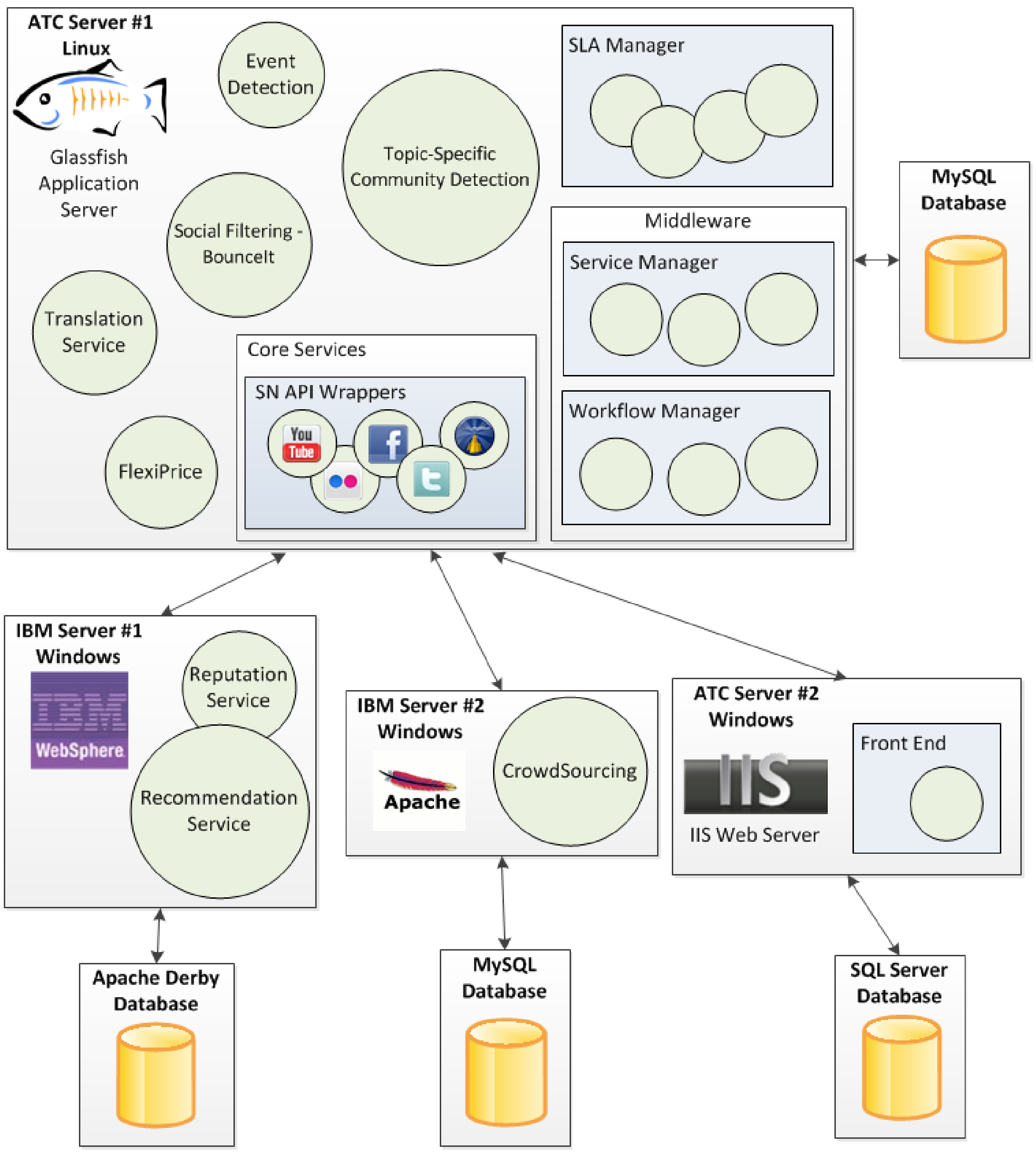


Figure 2. SocIoS physical deployment

3.2.3 *Logical diagram*

The abstract logical diagram of Figure 3 is representing SocIoS system from the scope of the actors. The interoperability between the components is clearly defined with the arrows which state the kind of communication that is taking place. This is a simple approach of the system, which makes it easier to understand the functionality and architecture of SocIoS. For a more detailed description of the operations and the flow of data between the components a number of data flow diagrams are provided in Section 5.

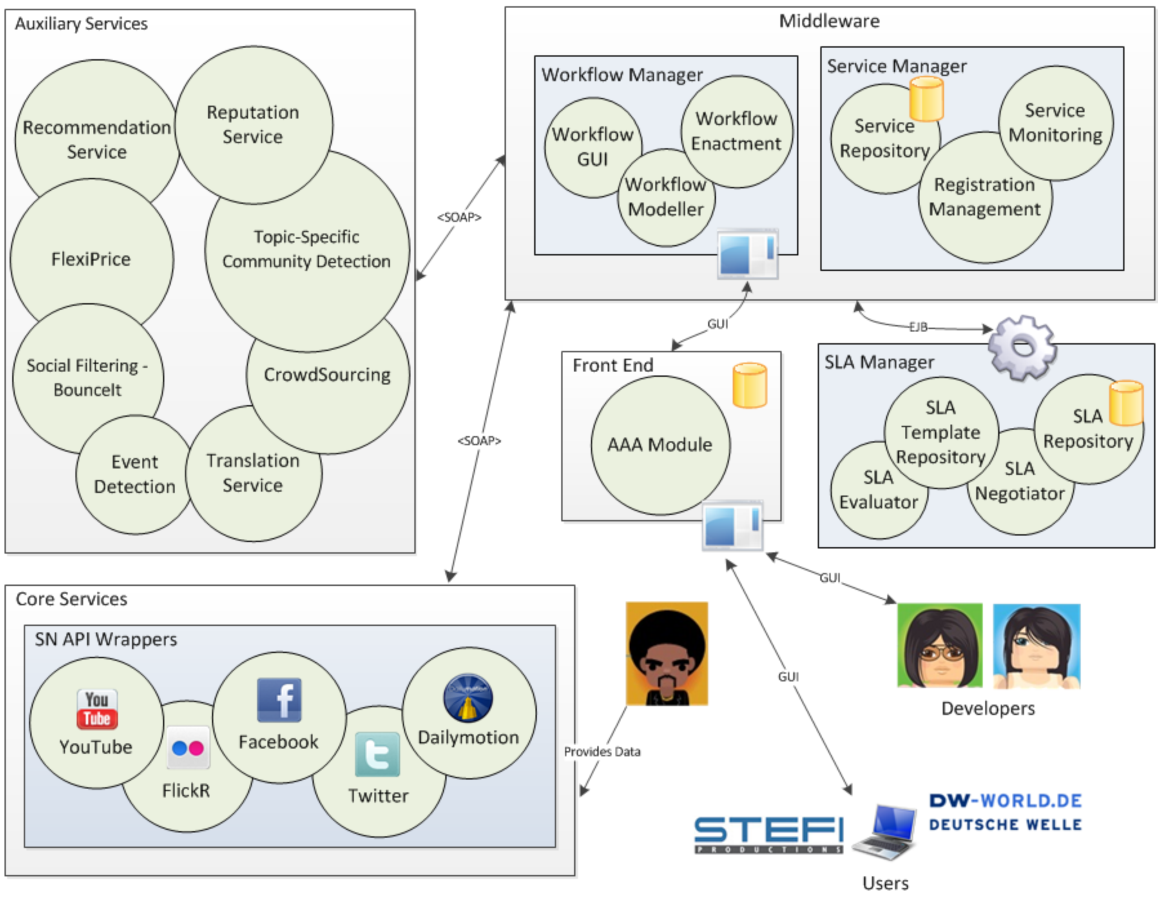


Figure 3. SocIoS Logical diagram

4 Description of SocIoS Integrated Prototype

4.1 Components and Services of SocIoS platform

This section briefly summarizes the main entities involved in SocIoS platform. A detailed description can be found in [3].

- **The SocIoS API**: An aggregation of methods provided by underlying SNS APIs. SocIoS API maps a standard interface to collections of methods and objects of the SNS APIs. This allows the developer to initiate multiple instances of the same functionality to various SNSs by a single method call and retrieve different objects mapped into a single reference data model. Moreover, SocIoS API is extended with the addition of new concepts and methods that are provided by the SocIoS Auxiliary Services.

- **The SocIoS core services**: The SocIoS core Services are a set of adaptors that implement the SocIoS API methods as well as the data transformation between the supported SNSs and the service layers (SocIoS Auxiliary Services). Those services provide a unique point of interaction with SNS exposing operations that encapsulate the functionality of the underlying SN APIs. All the operations of the core services are part of the SocIoS API.

- **The SocIoS auxiliary services**: third party services made available through SocIoS. There are two types of those services.

  - **Integrated Auxiliary services**: Those are services that extend the functionality of the SocIoS core services, use the same data models as the core services and, once registered with SocIoS, their operations are exposed as part of the SocIoS API. All Auxiliary services apart from the three listed below fall under this category. They cover a wide range of functionalities, from strictly SN related (e.g. reputation, event detection) to other functionality that can be useful in the context of SocIoS (translation). The integrated Auxiliary services are:
    - i. Recommendation service
    - ii. Reputation service
    - iii. Topic-Specific Community Detection
    - iv. Event Detection
    - v. Social Filtering
    - vi. Google Translate

  - **Non-integrated Auxiliary services**: Those are services that perform some independent functionality that is not part of the SocIos API but could be of great value to SocIoS end-users and is therefore accessible through the SocIoS Front-End. The non-integrated Auxiliary services are:
    - i. The FlexiPrice service

ii. The Crowdsourcing
iii. The SLA Management service

- **The SocIoS middleware**: SOA infrastructure consisting of loosely coupled services. The objective of this infrastructure is to provide end users a means of discovering auxiliary services and creating new applications by combining those services in workflows.

- **The SocIoS Front-End**: Front End to the SocIoS platform providing a user interface for interacting with the components as well as an authentication mechanism.

4.2 SocIoS workflow diagram(s)

With respect to SocIoS objectives and to end users requirements, SocIoS consortium defined four scenarios that correspond to the SocIoS applications for the end users in the project [2]. These scenarios will help the technical partners to align their work accordingly by customizing their components in a way that will expose SocIoS capabilities and functionalities.
SocIoS integrated prototype will be validated based on the selected scenarios. To this end, SocIoS integration team analysed the scenarios from a technical perspective and considering the implementation details of each component (interfaces, communication protocols) created the following workflow diagrams.
The SocIoS workflow diagrams (Figures 4-7) show how the data flows through the SocIoS system using a graphical representation and denoting the main input and output data of the components. It does not represent either hardware components or the location of the components in the system (e.g. servers/processes/memory). However, the interface characteristics, the way SocIoS components are linked to each other and their associations are realized.
It must be noted that until the compilation of this report, only the scenario of the 1st pilot trial has been defined in details from SocIoS technical team. Following the piloting plan, the 1st integration prototype (intermediate) will focus on the technical details and implementations of the 1st pilot execution. The workflow diagrams of the 2nd pilot trial (final SocIoS integrated prototype) will present whatever is defined so far and the label "ACK[2]" is used for the on-going implementations and used in order to have an overall view of the SocIoS platform.
Furthermore, it must be mentioned that all SocIoS components are being updated continuously according to the requirements/difficulties that are occurring. Thus, these presentations of SocIoS workflows may not be accurate in total (minor differences are expected). The first development phase will be continuously until M14.

[2] ACK = Acknowledgment

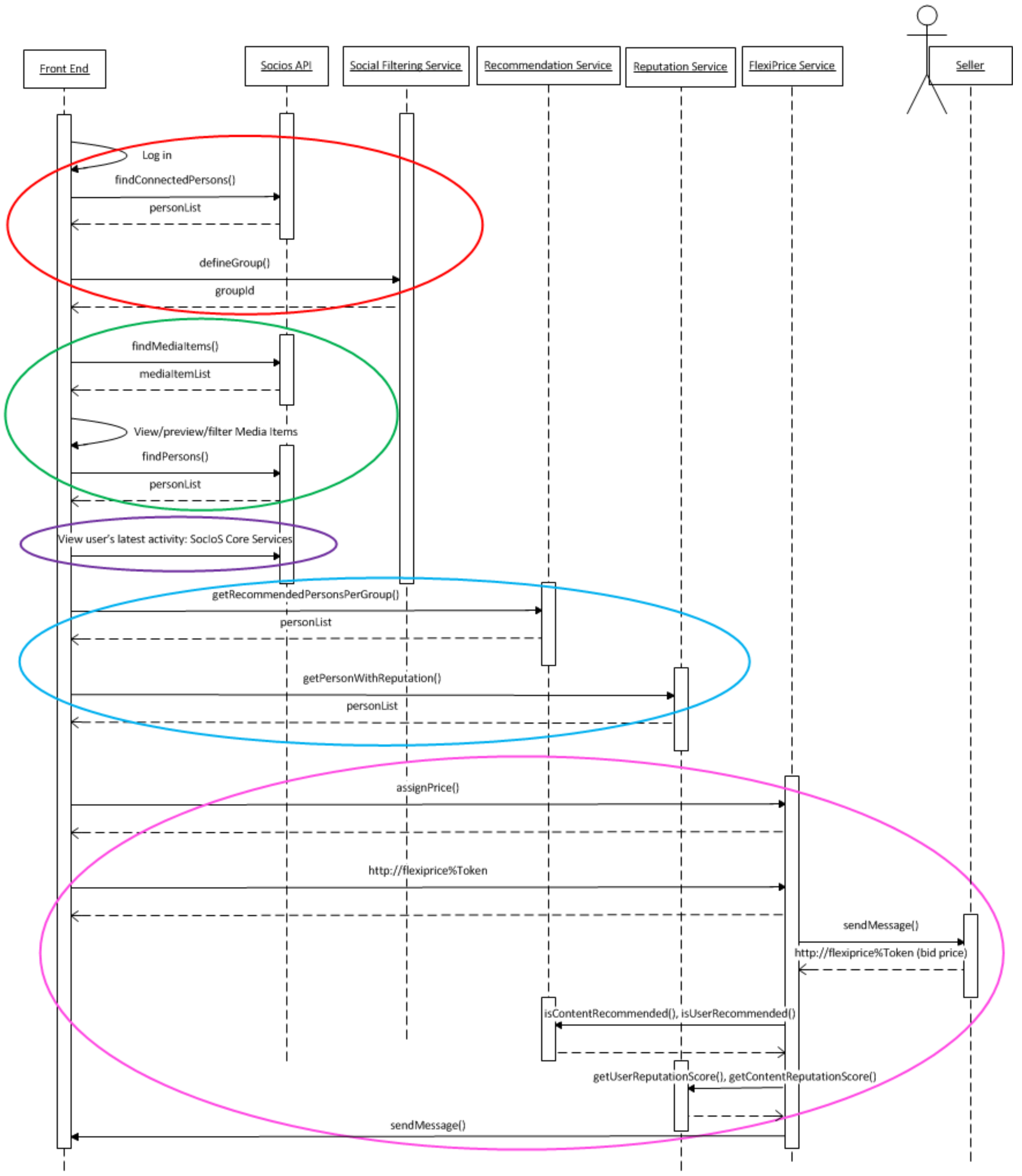
The following summarizations of the scenarios are extracted from [2].

Scenario A - Journalism

A journalist from DW plans to create a multimedia dossier article for online distribution that provides in-depth background information on a given topic, i.e. the current situation of conflict in Syria or any other topic that is relevant for the journalist. For the process of research she will consult traditional sources (e.g. experts, news agencies or online articles) as well as the new source, SocIoS platform. In order to obtain relevant content from social networks she needs to monitor the content activity of a group of SNS users who are related to "Syria" in terms of topic and location across the major social networks (at least Twitter, Facebook and YouTube). It is her aim to gather a large social media content pool from which she can then select media items, text information or "witnesses" to consider for her final dossier article. She is also interested to monitor the content-related activities of a group of people and then use the system to effectively sort these activities on the basis of their relationship to an important real-life event.

The scenario involves the following sequence of activities:

- Create a topical user group
- Add further SNS users and topical content (by topic and/or user location)
- Obtaining recommendations and adding users on the basis of their reputation
- Monitoring content activities and identifying real-life events
- Purchase selected content items.



- 🔴 Create a topical user group
- 🟢 Add further SNS users and topical content (by topic and/or user location)
- 🔵 Obtaining recommendations and adding users on the basis of their reputation
- 🟣 Monitoring content activities and identifying real-life events
- ● Purchase selected content items

Figure 4. Journalism scenario workflow diagram

Scenario B – Journalism App creation

A member of staff from DW's new media department has been asked to build a SocIoS application that will allows DW journalists to automatically get notifications for events related to the "Syria Conflict". He logs into SocIoS in the role of "Application Developer" and sets up a workflow using SocIoS Core and Auxiliary Services to achieve his goal. He expects that if something important happens in Syria, his SocIoS application will capture it, immediately allowing the DW journalists to be among the first to know. This is possible by leveraging the viral dissemination effect of social networks but also the ability to dynamically track the sources of interest in the social graph. Once this important event is manifested as a content item (a photo, tweet, etc.) the journalists can consider it via the application as part of their research and newsgathering process. DW's SocIoS digital purse account would be billed with the expenses from using the intermediate services of the workflow.

The scenario involves the following sequence of activities:

- Service access via authorisation and authentication
- Reviewing available services
- Designing and deploying workflows
- Purchasing services and executing workflows

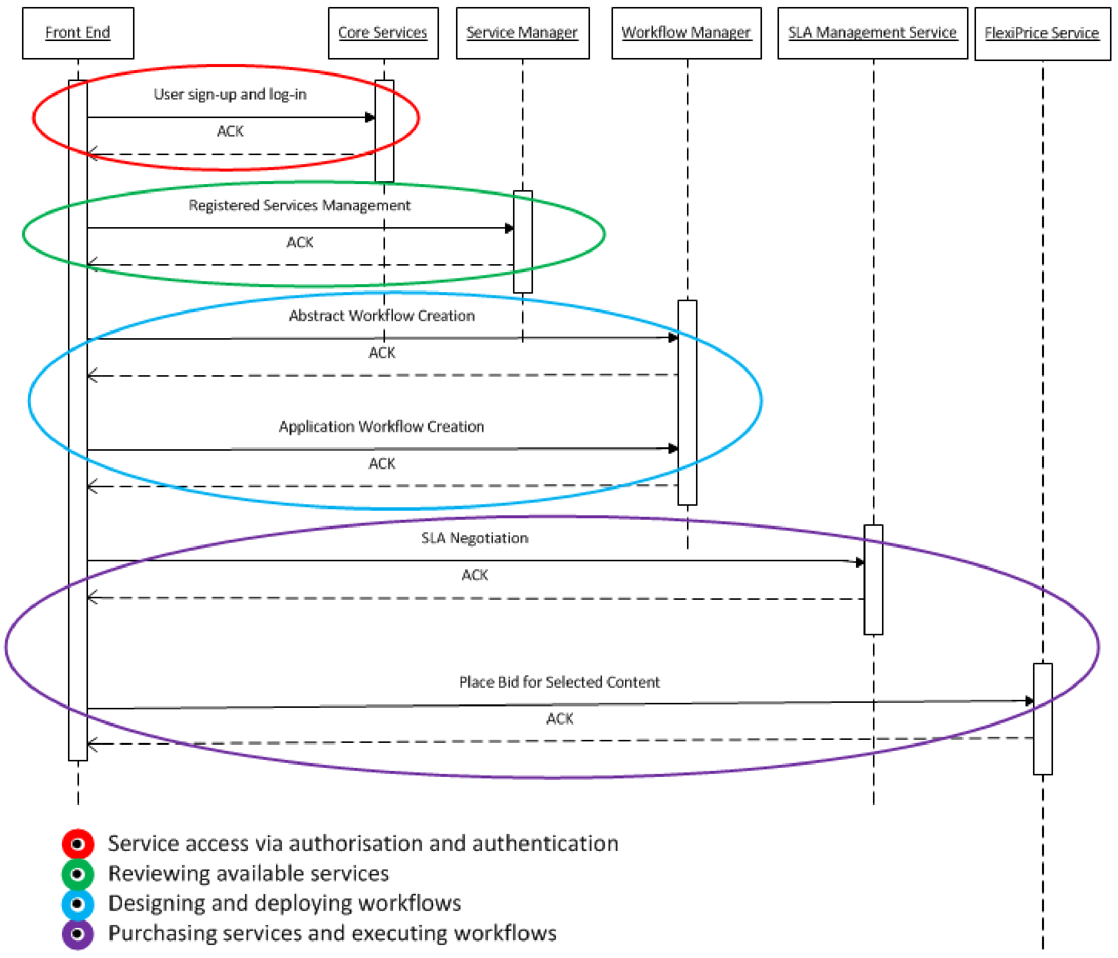

Front End
Core Services
Service Manager
Workflow Manager
SLA Management Service
FlexiPrice Service
User sign-up and log-in
ACK
Registered Services Management
ACK
Abstract Workflow Creation
ACK
Application Workflow Creation
ACK
SLA Negotiation
ACK
Place Bid for Selected Content
ACK
Service access via authorisation and authentication
Reviewing available services
Designing and deploying workflows
Purchasing services and executing workflows

Figure 5. Journalism app creation workflow diagram

Scenario C – Casting Extras

A TV commercial producer from Stefi, wants to cast a number of extras for a new commercial. The requirements for the extras appearance are specific: male, over 1.80 m height, less than 90kg weight and blonde hair. He wants to send out the description and get back proposals. He wants to take advantage of the fact that in SNSs, users are already providing information and photographs about themselves. Since the producer himself cannot search all the SNS users for that purpose, he leverages the viral dissemination effect (the fact that users are connected) of SNS so as to get hints about his request even from other users and not necessarily directly to the ones that match the description. He believes that a simple request will quickly generate thousands of results that can be easily filtered, but he is concerned about the reliability of the source of the information. The scenario involves the following sequence of activities:

- Creation of crowdsourcing games with rewards model
- Receiving input from SNS users
- Analysing/monitoring results and creating a user group
- Locating Extras Using Social Graph Auctions
- Selecting users for casting purposes.

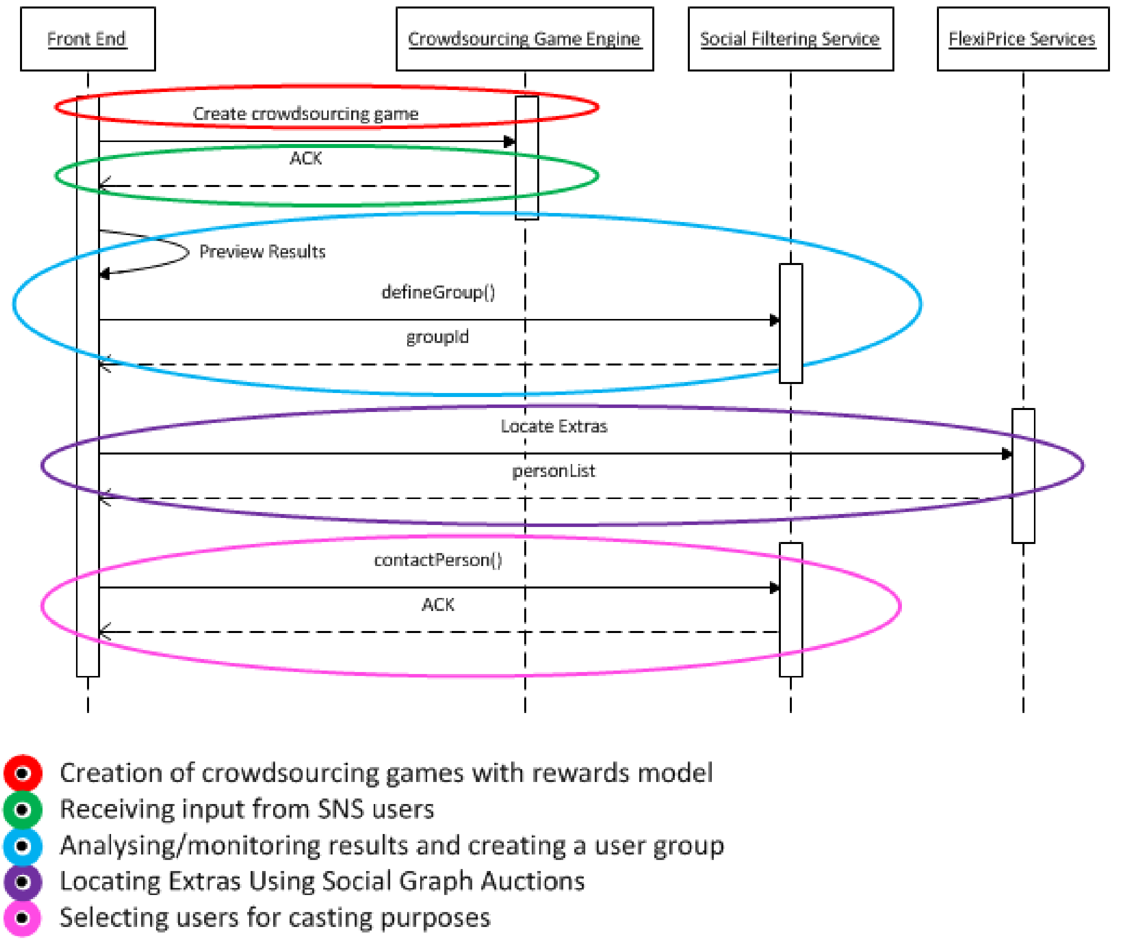

Front End
Crowdsourcing Game Engine
Social Filtering Service
FlexiPrice Services
Create crowdsourcing game
ACK
Preview Results
defineGroup()
groupId
Locate Extras
personList
contactPerson()
ACK
Creation of crowdsourcing games with rewards model
Receiving input from SNS users
Analysing/monitoring results and creating a user group
Locating Extras Using Social Graph Auctions
Selecting users for casting purposes

Figure 6. Casting extras workflow diagram

Scenario D – Location Scouting

This time the producer from Stefi needs to locate a place where they can conduct the shooting. The place description is very specific, with sketches identifying the front view of an 18th century classic-architecture building, with a wooden door and 2 windows. The producer wants to send out the description and get back photographs that depict a place that looks as close as possible to the target location. He wants to take advantage of the "wisdom of the crowd" that will help him answer his question, quickly and at a low cost.

The scenario involves the following sequence of activities:

- Creation of crowdsourcing game with reward model
- Receiving results from SNS users
- Monitoring and filtering results
- Scouting for locations Using Social Graph Auctions
- Contacting users with information about suitable locations

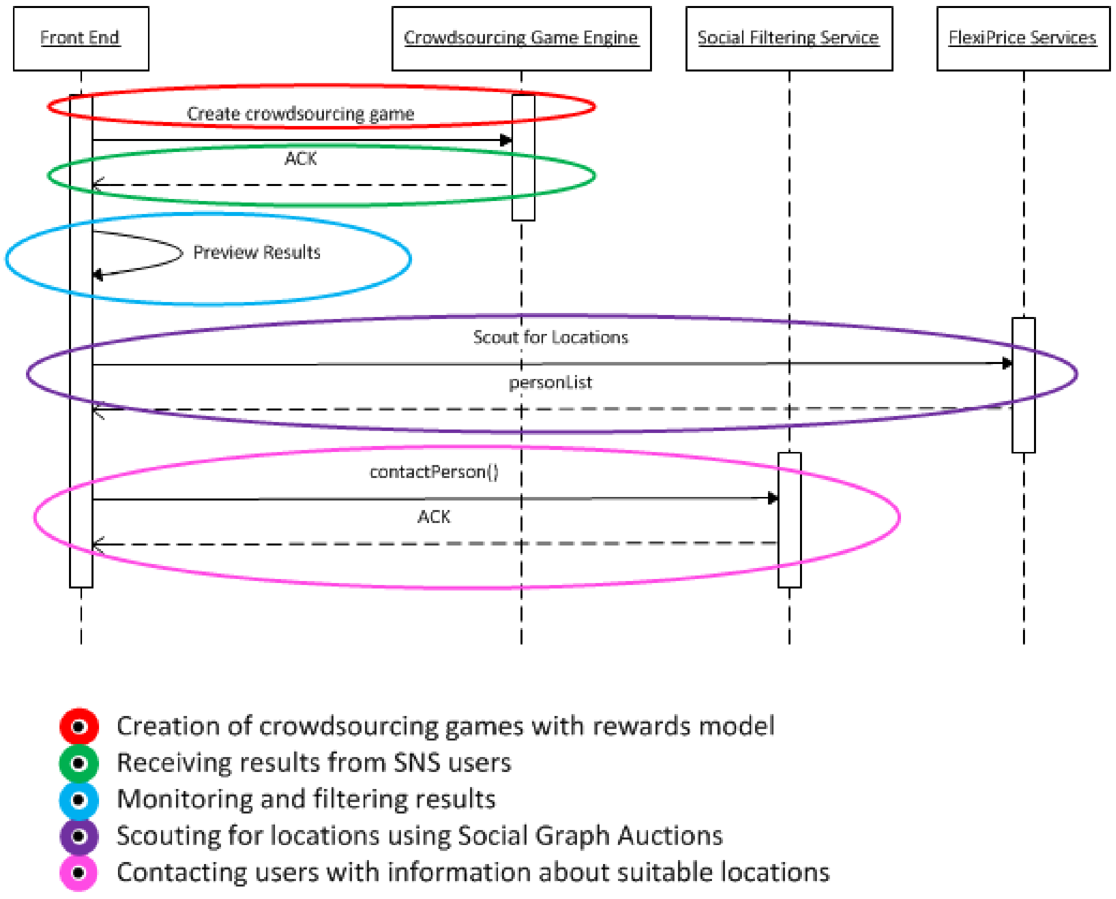

Front End
Crowdsourcing Game Engine
Social Filtering Service
FlexiPrice Services
Create crowdsourcing game
ACK
Preview Results
Scout for Locations
personList
contactPerson()
ACK
Creation of crowdsourcing games with rewards model
Receiving results from SNS users
Monitoring and filtering results
Scouting for locations using Social Graph Auctions
Contacting users with information about suitable locations

Figure 7. Location scouting workflow diagram

4.3 *Hardware facilities and installation issues*

Based on the physical deployment depicted in Figure 2, the current configuration of the SocIoS platform involves the setup of the following physical servers that will host SocIoS components:

PC hosting Core (Auxiliary) Services, SLA Manager and Middleware (Glassfish Application Server[3])

- CPU:
    - Intel(R) Xeon(R) CPU E5520 @ 2.27GHz,
    - CPU speed: 2.27 GHz
- Memory:
    - 2 GB RAM
- Operating system: Ubuntu 10.04 LTS
- Software installed: GlassFish Server Open Source Edition 3.1 (build 43)

PC hosting MySQL database[4] (supporting Glassfish Application Server)

- CPU:
    - Intel(R) Xeon(R) CPU E5520 @ 2.27GHz,
    - CPU speed: 2.27 GHz
- Memory:
    - 2 GB RAM
- Operating system: Ubuntu 10.04 LTS
- Software installed: MySQL ver. 14.14 distrib. 5.1.41

[3] Please refer to http://glassfish.java.net/ for more detailed information

[4] Please refer to http://www.mysql.com/ for more detailed information

PC hosting Front End (IIS web server[5])

- CPU:
  - Intel Xeon CPU E5520 @ 2.27 GHz (2 Processors)
  - CPU speed: 2.27 GHz
- Memory:
  - 2 GB RAM
- Operating system: Windows Server 2008 R2 Standard
- Software installed: IIS ver. 7.5.7600.16385

PC hosting Microsoft SQL Server database[6] (supporting IIS Web Server)

- CPU:
  - Intel(R) Xeon(R) CPU x2 E5520 @ 2.27GHz, 2267 Mhz, 1 Core(s), 1 Logical Processor(s)
  - CPU speed: 2.27 GHz
- Memory:
  - 2 GB RAM
- Operating system: Microsoft Windows Server 2008 R2 Standard 64bit
- Software installed: Microsoft SQL Server 2008 R2

PC hosting IBM Recommendation and Reputation Services

- CPU:
  - Intel Core™ 2 Duo E6550 2.33GHz
  - CPU speed: Minimum 2 GHz
- Memory:
  - Minimum 2 GB RAM

[5] [5] Please refer to http://www.iis.net/ for more detailed information

[6] Please refer to http://www.microsoft.com/sqlserver/en/us/default.aspx for more detailed information

- Operating system: Microsoft Windows XP sp3
- Software installed:
    - IBM WebSphere Application server v7.0
    - Apache Derby 10.4.2.0

PC hosting IBM CrowdSourcing Service

- CPU:
    - Intel Core™ 2 Duo E6550 2.33GHz
    - CPU speed: Minimum 2 GHz
- Memory:
    - Minimum 2 GB RAM
- Operating system: Microsoft Windows XP sp3
- Software installed: WAMP Server 2.1

For installation guidelines and technical details for each one of the SocIoS components separately refer to the relevant deliverables, as follows:

- For the SocIoS API, refer to [3] [4].
- For the SocIoS core and Auxilliary Services, refer to [3] [6] [8] [9].
- For the FlexiPrice, Crowdsourcing and SLA Manager Services, refer to [3] [7] [9].
- For the SocIoS Middleware, refer to [3] [5] [6].
- For the Front-End, refer to [3].

4.4 Deploy and configure SVN environment as a central versioning repository

For enabling the code sharing between the technical partners of SocIoS, as well as supporting the collaboration for code delivery to the Project Officer and reviewers, an efficient code repository is needed. The most known repository software for developers is Apache Subversion. Subversion repository (SVN) is a software versioning and a revision control system founded and sponsored in 2000 by CollabNet Inc.

However, apart from code sharing, other functionalities maybe also useful, such as documentation creation and developer forums. Thus, the SocIoS integration team decided to utilize a GForge server[7] instance. GForge is an open source software providing a modern and extensible platform with an intuitive interface that ties together a huge toolset, from Source Code Management (SCM) to extremely customizable Trackers, Task Managers, Document Managers, Forums, Mailing Lists. All of these are controlled by a centralized permission system and maintained automatically by the system. For supporting the SocIoS project an already existing GForge community edition server in NTUA was decided to be used, hosted in an Ubuntu server with the following specifications:

PC hosting GForge Community Edition:

- CPU:
    - Intel(R) Pentium(R) 4 CPU
    - CPU speed: 3.20GHz
- Memory:
    - Minimum 2 GB RAM
- Operating system: Ubuntu 10.04
- Software installed: GForge Community Edition v5.7

4.5 A bug-tracking component (BugZilla)

Considering that SocIoS developments will be executed in two phases that each one will develop an integrated prototype (intermediate and final) that will be utilized in the pilot trials, it is essential to deploy a bug-tracking component in order to help each iteration. In this way the transition from the 1st development to the 2nd will be smoothly as each partner will be able to submit any bug noticed for any of the SocIoS component. Furthermore, it is expected that during the integration test of the intermediate prototype and its evaluation through the 1st pilot trial, refinements maybe needed. The bug-tracking component will be used offline to store any observations of that kind without interrupting the integration and evaluation phase. The 2nd development iteration will be initiated via the bugs reported in this component.

[7] http://gforge.org/gf/

The integration team decided to utilize the BugZilla[8] software due to its following characteristics:

1. Open source
2. Defect tracking feature embedded (allow individual or groups of developers to keep track of outstanding bugs in their product effectively)
3. Bug-Tracking feature embedded
4. Several other useful features like advanced search, email notifications, multiple formatted bug lists, reports and charts, time tracking etc.
5. Favorite of thousands organizations across the globe

Finally, it must be mentioned that the integration leader has been used before the software in several mobile and other applications that was engaged.

[8] http://www.bugzilla.org/about/

5 Testing cases for the Intermediate Integrated Prototype

As already mentioned in the previous sections, the 1st SocIoS Integration phase will deploy the intermediate SocIoS prototype that will be utilized in the 1st Pilot execution. In order to assure that this prototype is a bug-free stable platform that will comply with the end-user requirements and project's objectives, SocIoS consortium decided to perform some technical testing activities foreseen within the integration work package (WP4). These activities are necessary to assess the performance and the level of fulfilment of the required functionality of the SocIoS system, from the operational point of view. The platform must be checked on usability of the resulting system in terms of ease of use, capacity of the subsystems and overall accessibility of the technical features and services. These tests will take place both at the level of individual components and at the overall system performance level, including activities related to software testing in order to verify the SocIoS environment suitability and stability.
The test results, together with the user evaluation outcomes (WP5) and reporting of possible problems by project partners or other developers/users of the platform, will provide feedback to the enhancement of the functionality of the SocIoS system, occurring from the first version towards the final one.

Deliverable D4.3.1. "Verification and Validation Report" (M15) will describe in details the main findings from the V&V process during integration as well as the procedure followed and suggestions for further improvement of the software.

With respect to the 1st pilot execution, SocIoS consortium analysed the pilot scenario and divided it to seven sub-workflows in an attempt to capture all the methods and functionalities. These workflows will be the actual test cases that the integration team will focus on validating and are very useful as i) consistency will be checked and all the services are going to provide the methods needed for the scenario, ii) they comprise a first integration plan and iii) the SocIoS API will be updated accordingly in case something is not included.

NOTE: Preview, Translation and any suchlike functionalities are taking place at the level of GUI, therefore are not presented here.

The following figures present the respective sub-workflows (test cases) of the 1st SocIoS Pilot scenario.

Sub-Workflow 1: Journalist gets her SNS friends and then searches in their content to find something related to “Syria clashes”. Then she gets the filtered list of SNS users and tags them with a Group name

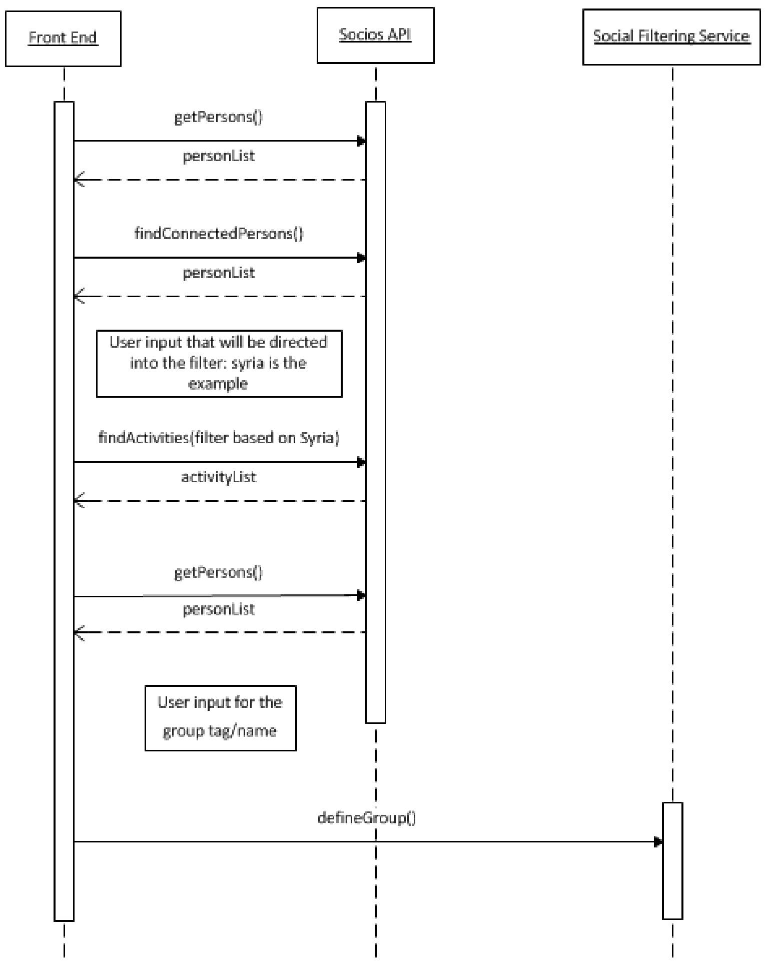


Figure 8. Sub-workflow 1, “Journalist searches in the content of her SNS friends”

Sub-Workflow 2: The journalist searched in the whole SNS user base that SocIoS has access, for users with content that relates to "Syria clashes" and when she finds them she extends the previously created user Group.

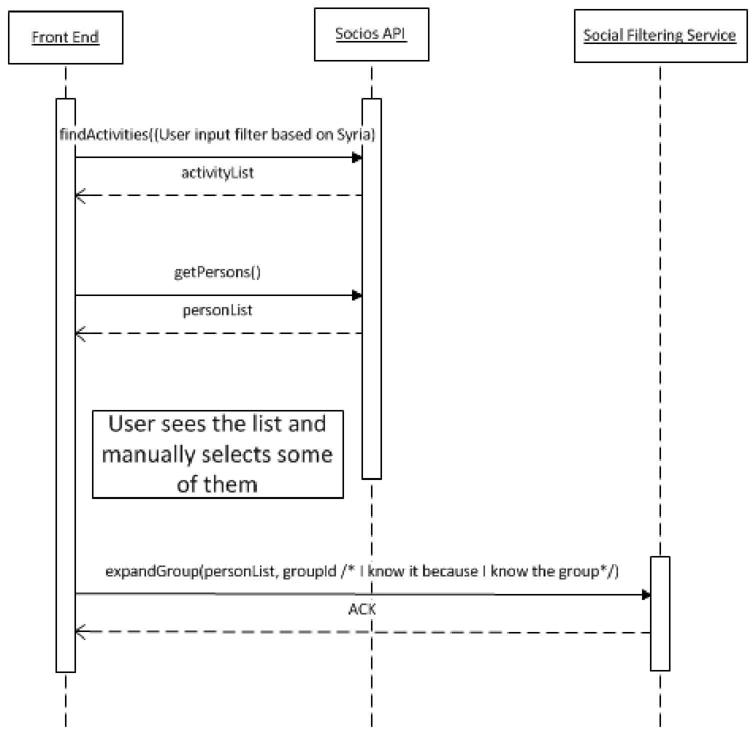

Front End
Socios API
Social Filtering Service
findActivities((User input filter based on Syria)
activityList
getPersons()
personList
User sees the list and manually selects some of them
expandGroup(personList, groupId /* I know it because I know the group*/)
ACK

Figure 9. Sub-workflow 2, "Journalist searches in the whole SNS user base of SocIoS"

Sub-Workflow 3: The journalists initiates a search for SNS users that SocIoS can access their data, who are located or related in any way with "Syria". She gets the returned users and extends the user Group by adding them too.

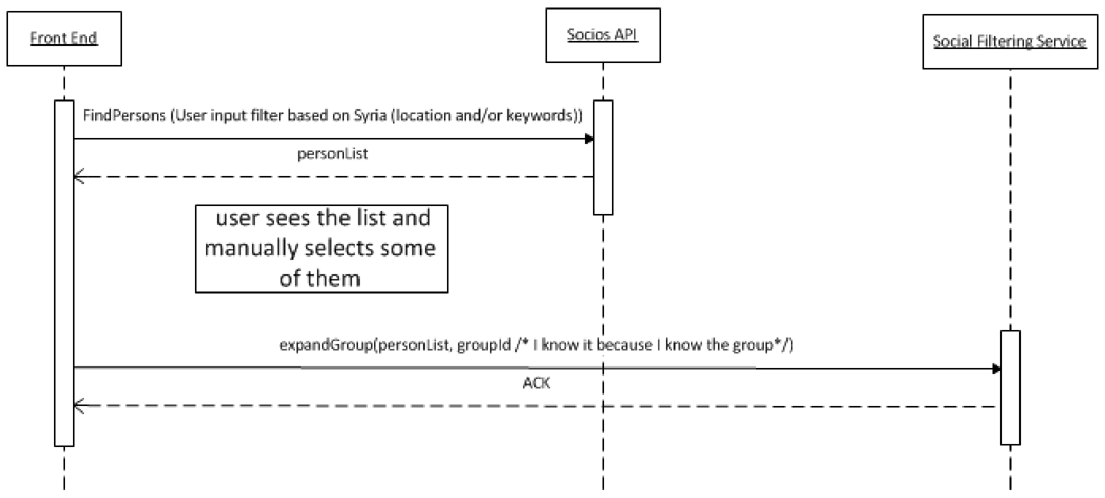

Front End
Socios API
Social Filtering Service
FindPersons (User input filter based on Syria (location and/or keywords))
personList
user sees the list and manually selects some of them
expandGroup(personList, groupId /* I know it because I know the group*/)
ACK

Figure 10. Sub-workflow 3, "Journalist searches for SNS users based on their location"

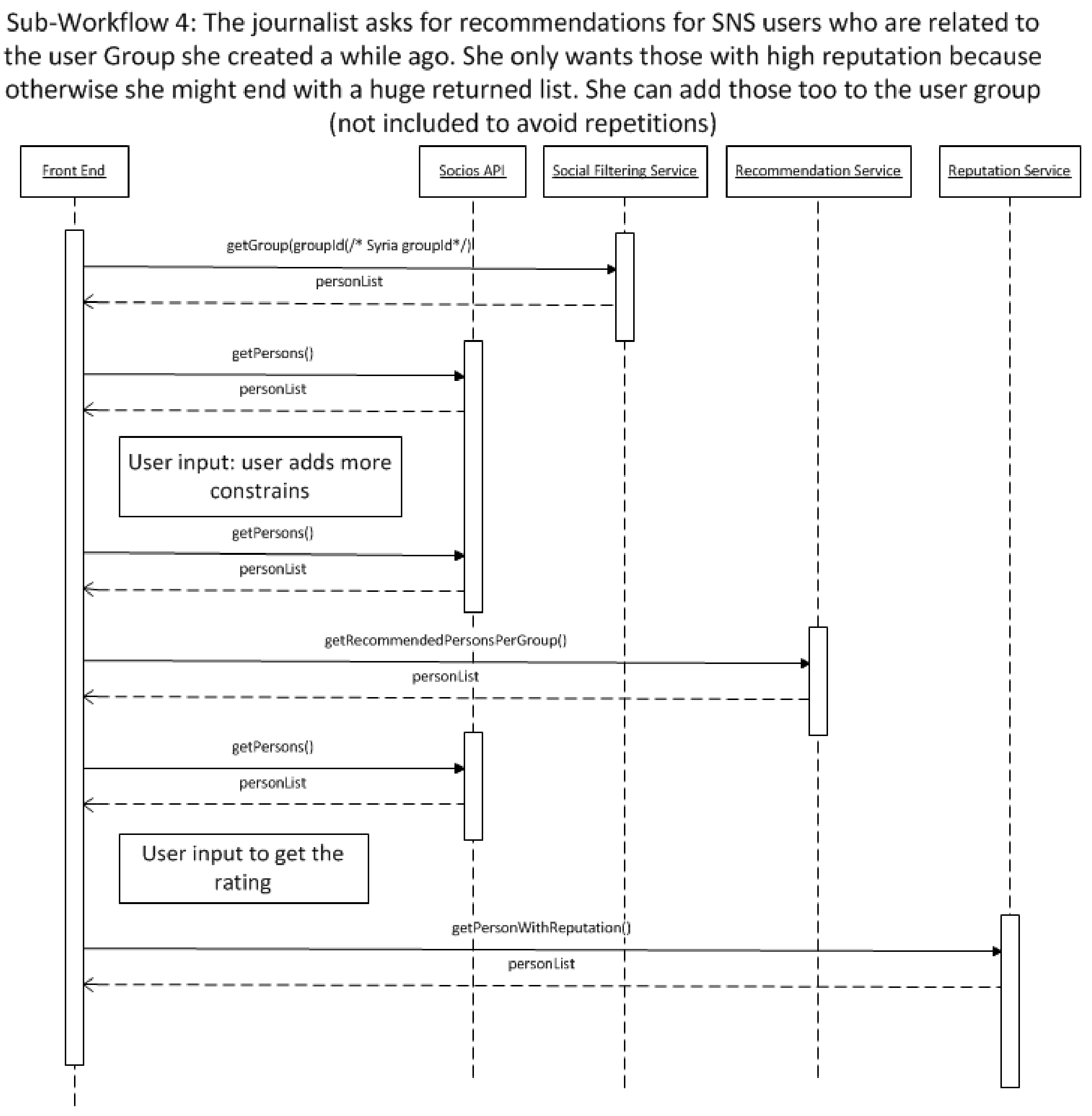

Sub-Workflow 4: The journalist asks for recommendations for SNS users who are related to the user Group she created a while ago. She only wants those with high reputation because otherwise she might end with a huge returned list. She can add those too to the user group (not included to avoid repetitions)

Figure 11. Sub-workflow 4, "Journalist utilizes SocIoS recommendation and reputation services"

Sub-Workflow 5: At some point later, the journalist asks from the event detection service to detect if there were noteworthy activities that the SNS users from the user Group. The event detection service returns a list of activities (tweets, video uploads, etc) of these SNS users that spurred intense activity,

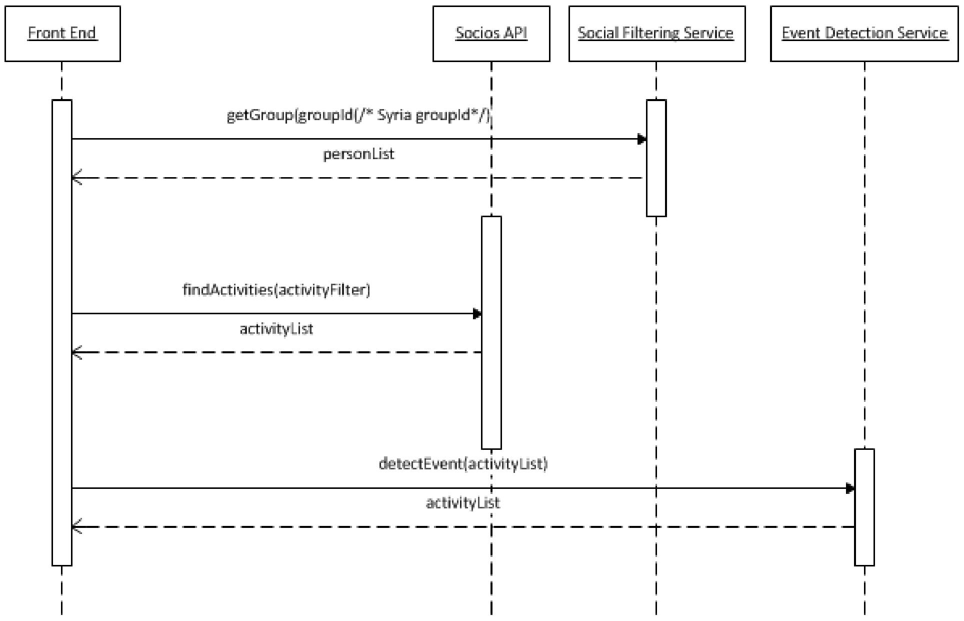


Figure 12. Sub-workflow 5, “Journalist utilizes SocIoS event detection service”

Sub-Workflow 6: At any point the journalist asks to see the activities of the user Group of the last 5 hours

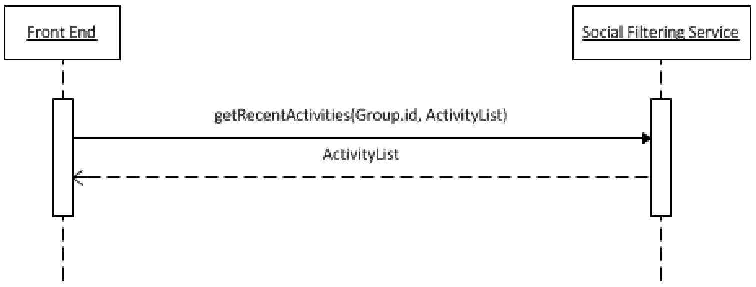


Figure 13. Sub-workflow 6, “Journalist asks to see user group activities”

Sub-Workflow 7: At any point the journalist may request to contact an SNS user from the group in order to make a transaction

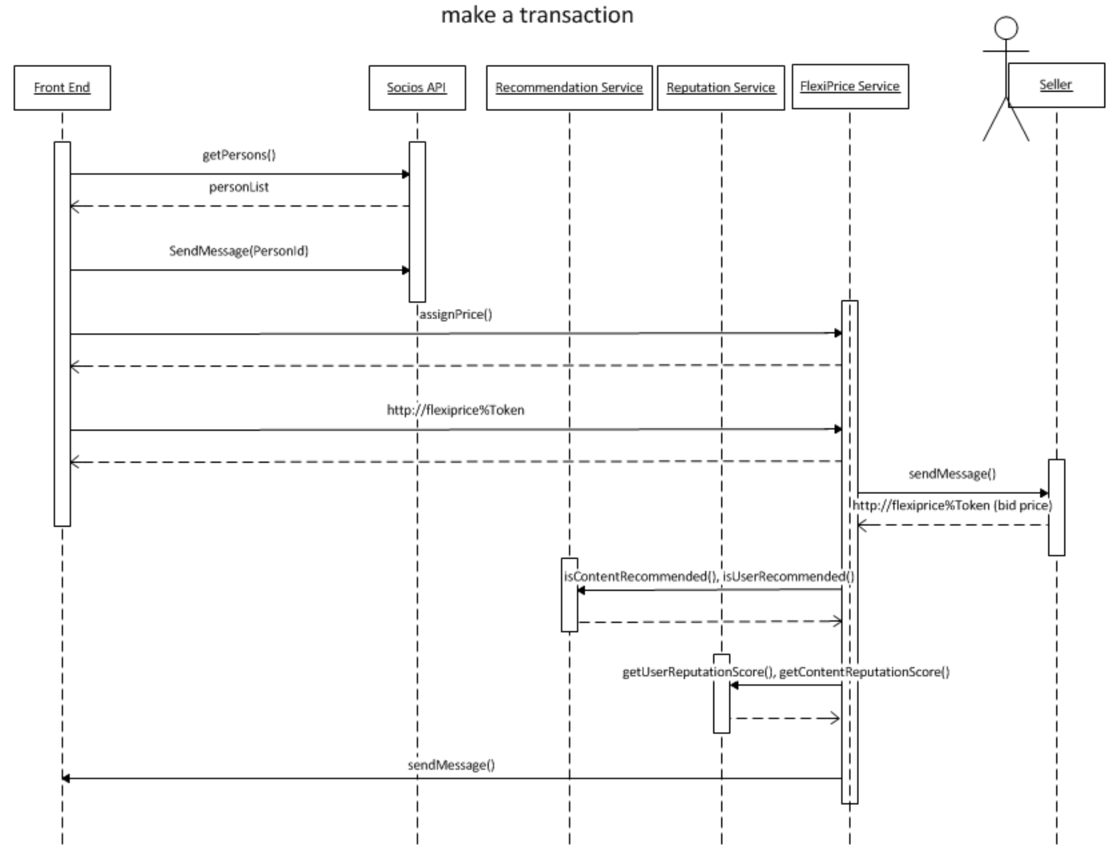


Figure 14. Sub-workflow 7, “Journalist contacts SNS user and makes a transaction”

6 Time-line Integration Activities and Responsibilities

This section defines the responsibilities that each partner of the integration team has undertaken and the internally steps of the technical coordination focusing on the time line integration activities.
The following table (Table 4) is an adjustment of Table 2, presenting the SocIoS components, the responsible partner, the Workpackage that it is allocated to, the date of its finalization and the version that will be integrated to.

Table 4. SocIoS responsibilities and dates of finalization

SocIoS component	Responsible partner	Relevant WP	Planned date	Integrated in version
Service Manager	NTUA	WP2	31/08/2011	Intermediate
Workflow Manager	NTUA	WP2	31/08/2011	Intermediate
WorkflowGUI	NTUA	WP2	31/08/2011	Intermediate
Recommendation Service	IBM	WP3	31/10/2011	Intermediate
Reputation Service	IBM	WP3	31/10/2011	Intermediate
Topic-Specific Community Detection	NTUA	WP2	21/10/2011	Intermediate
Event Detection	NTUA	WP2	21/10/2011	Intermediate
Social Filtering - BounceIt	Cognium	WP3	21/10/2011	Intermediate
Translation Service	ATC	WP3	21/10/2011	Intermediate
SLA Manager	NTUA	WP3	31/08/2011	Intermediate
FlexiPrice	UH	WP3	21/10/2011	Intermediate
IPR Clearance Service	NTUA	WP3	24/08/2012	Final
Crowdsourcing Service	IBM	WP3	24/08/2012	Final
SocIoS Front-End	ATC	WP2	21/10/2011	Intermediate

The integration activities for each one of the two iterations consist of the following phases:

1. **Compilation of the integration plan**. It will be compiled by the integration leader and it will describe the main integration activities that must be performed in the two integration phases. The current report is SocIoS integration plan.

2. **Aggregation of the incorporated components**. All the technical partners must send their components or the respective interfaces to the integration leader. Their obligation is to follow SocIoS communication's framework principles [3] and to stabilize the components in a way that they will be bug-free prototypes. The latter must comply with the functionalities and objectives defined by SocIoS consortium.

3. **Set up activities** – Integration of the components to the SocIoS platform. After aggregating all the SocIoS prototypes, the integration leader will follow the installation guidelines provided for each one of the component and will install them (if needed) to the respective physical machines. Furthermore, the integration leader must assure that all communication channels between them are active and all the interfaces are following the predefined specifications.

4. **Integration tests – first execution**. All the technical partners of the consortium must participate in the first execution of the integration tests. It is foreseen that technical workshops will be organized (one for each iteration). As an alternative, the integration leader will organize a series of conference calls in order to bring together the people that are responsible for SocIoS components. According to the integration test that is executes each time, the respective people will participate. Finally, as already mentioned the integration plan defines the integration tests and each one of them will be executed more than once.

5. **Adjustments**. The first execution phase will provide crucial feedback and input to the technical team of SocIoS. The adjustments activity will give the opportunity to make any needed corrections to the components of the platform. Furthermore, a refinement in the framework configuration will be made accordingly to those indications coming from the previous experience. After fine tuning, the necessary preparation for the final execution will follow. This period will be led by the integration leader in cooperation with the technical team of the project.

6. **Final Execution – Prototype released**. The final execution will be organized as the first one. It will be based on conference calls and on the same integration tests. Considering that the technical partners will have done the necessary refinements, the result of the final execution phase will be a stable, bug-free 1st SocIoS integrated prototype that will be used in the 1st SocIoS pilot trial. The prototype release will be accompanied by the respective software documentation. Furthermore, the final execution is also when the evaluation of the integrated version of the platform will take place. This activity will be carried out by the consortium as a whole and especially by the end-users (people that belong to the project's team and not outsiders) who will instantiate the test cases. Their observations on the reactions of the platform will be the basis of the evaluation.

7. **Verification and validation** – Compilation of the report. Following the final execution, the integration leader will aggregate all the comments/suggestions of the consortium and will distribute the respective validation report with suggestions for further improvements of the software.

The following table (Table 5) presents SocIoS integration time-plan activities.

Table 5. SocIoS integration time-plan activities

Activities description	M12 August 2011	M14 October 2011		M15 November 2011				M24 August 2012		M25 September 2012			
		Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
Compilation of the integration plan	■												
Aggregation of the incorporated components		■						■					
Set-up activities			■						■				
First execution				■						■			
Adjustments					■						■		
Final execution						■	■					■	■
Verification and Validation							■						■

7 Conclusions

This document presented the integration plan of the SocIoS project. The main purpose of the report is to provide instructions for the software integration phases that will follow. The ultimate goal of the integration activities is the release of two prototypes (intermediate and final) that will be used from the two pilot trials in order to evaluate the functionalities and capabilities of SocIoS platform.
Starting from the user requirements, this report presented the SocIoS functionalities and the respective use cases that the prototype must comply to. The non-functional requirements (such as security, availability, scalability, etc) were also described.
The components of the platform were mentioned and according to their specifications a physical diagram with the actual machines that will be utilizes was produced. Furthermore, the component's interfaces defined the core communication principles and several dataflow diagrams have been created. For the 1st integrated prototype, some of these diagrams will be the test cases that the integration team will use to evaluate the prototype.

During the compilation of this report, the deployment of the components hasn't been ended yet. Most research modules, as well as the system itself, will be updated. For the next two months, the goal is to enrich the existing functionalities, finalise the research work and integrate the respective components, in order to deliver the intermediate prototype version of the SocIoS integrated prototype.
The final version of the SocIoS integrated system is foreseen for M25 of the project, providing enough time to the target stakeholders to evaluate the system capabilities and provide their perception on the proper adjustments for the commercialisation of the system post project end.

8 References

[1] SocIoS deliverable D2.1b. "Requirements Management Guide".

[2] SocIoS deliverable D5.1.1. "Pilot Implementation Plan: Scenarios and Evaluation"

[3] SocIoS deliverable D2.5.3. "SocIoS Overall Architecture"

[4] SocIoS deliverable D2.2. "SN Platform APIs analysis"

[5] SocIoS deliverable D2.5.1. " SOA Infrastructure design"

[6] SocIoS deliverable D2.6.1. "SOA Infrastructure Prototype and Report"

[7] SocIoS deliverable D3.2.1. "Business Toolset"

[8] SocIoS deliverable D3.3.1. "User Created Content Management Services"

[9] SocIoS deliverable D3.4.1. "Social Dynamics Services"