Contract: 027617

SPICE

Service Platform for Innovative Communication Environment

FP6 - Integrated Project (IP)

Priority T2 – Information Society Technologies

WP N°: 1
Deliverable D1.1:
SPICE User, enterprise, technical and open market business requirements

Due date of deliverable: 31/03/2006
Actual submission date: 02/05/2006

Start date of project: 01/01/2006
Duration: 30 Months

Project coordinator name: Christophe Cordier

Organisation name of lead contractor for this deliverable: NOKIA, C. Riva

Editor: Claudio Riva

Revision: v1.4

PROPRIETARY RIGHTS STATEMENT
THIS DOCUMENT CONTAINS INFORMATION, WHICH IS PROPRIETARY TO THE SPICE CONSORTIUM. NEITHER THIS DOCUMENT NOR THE INFORMATION CONTAINED HEREIN SHALL BE USED, DUPLICATED OR COMMUNICATED BY ANY MEANS TO ANY THIRD PARTY, IN WHOLE OR IN PARTS, EXCEPT WITH THE PRIOR WRITTEN CONSENT OF THE SPICE CONSORTIUM THIS RESTRICTION LEGEND SHALL NOT BE ALTERED OR OBLITERATED ON OR FROM THIS DOCUMENT
TABLE OF CONTENTS

1. INTRODUCTION 5

1.1. PURPOSE OF THIS DOCUMENT 5
1.2. DOCUMENT OVERVIEW 5
1.3. EDITORS 6
1.4. CHANGE HISTORY 6

2. SPECIFICATION OF THE SPICE SYSTEM REQUIREMENTS 8

2.1. THE RATIONAL FOR SPICE 8
2.2. SPICE HIGH-LEVEL OBJECTIVES 10
2.3. SPECIFICATION OF THE REQUIREMENTS 10
2.4. THE BODY PARTS FOR THE USE CASE TEMPLATE 13
2.5. BODY PARTS FOR THE REQUIREMENTS TEMPLATE 16
2.6. TERMINOLOGY FOR THE REQUIREMENTS 17

3. USE CASES 18

3.1. SERVICE HANDOVER (CONTINUITY OF SERVICE) 21
3.1.1 UC1.1 - COMMUNICATION SESSION FOLLOWING USER 21
3.1.2 UC1.2 - WATCHING MOVIES 22
3.1.3 UC3.10 – CHANGING DEVICE (SPLIT CONTENT STREAMS TO DIFFERENT DEVICES) 22
3.1.4 UC7.2 - CONTENT ADAPTATION ACCORDING TO CAPABILITIES OF OTHER AVAILABLE TERMINALS 23

3.2. DEVICE MANAGEMENT AND CONFIGURATION 24
3.2.1 UC1.3 - BUYING A NEW PHONE 24
3.2.2 UC3.5 - DEVICE MANAGEMENT 25

3.3. SUBSCRIPTION AND CONFIGURATION OF SERVICES 26
3.3.1 UC3.1 - END USER INSTALLS A DYNAMIC ADDRESS BOOK APPLICATION 26
3.3.2 UC3.6 - USER SUBSCRIBES TO A NEW SERVICE. 28
3.3.3 UC7.5 – SPECIFY CONTENT DELIVERY 28

3.4. CONTEXT AWARENESS 29
3.4.1 UC3.2 - AUTOMATIC UPDATED OF REACHABILITY 29
3.4.2 UC3.3 – COST OPTIMIZED CONNECTION ESTABLISEMENT 31
3.4.3 UC3.4 - ENHANCING SERVICE DELIVERY USING REACHABILITY 33
3.4.4 UC1.4 - FINDING ATTRACTIONS 33
3.4.5 UC7.1 CONTENT ADAPTATION ACCORDING TO USER POSITION 36
3.4.6 UC7.4 – PRIORITY OF SPICE PLATFORM IN DANGEROUS SITUATIONS 37

3.5. SERVICE LIFECYCLE 39
3.5.1 UC 1.7 - APPLICATION DESIGNER BUILDS A NEW APPLICATION 39
3.5.2 UC3.7 - SERVICE PROVIDER DEPLOYS A NEW SERVICE 39
3.5.3 UC 3.11 – SERVICE SENDS A MULTI-MODAL NOTIFICATION TO USER 40
3.5.4 UC7.6 – TERMINALS ACTUALIZATION AND RECONFIGURATION
3.5.5 UC 5.4 – SERVICE DEVELOPER CREATES A NEW SERVICE
3.5.6 UC 5.5 – END USER CREATES A NEW SERVICE
3.6. SERVICE EXECUTION MODEL
3.6.1 UC5.1 INVOKE A SESSION COMMUNICATION SERVICE
3.6.2 UC5.2 SERVICE USING THE IT/WEB SERVICE EXECUTION MODEL (CLICK-TO-DIAL)
3.6.3 UC5.3 SERVICE USING THE MESSAGING EXECUTION MODEL
3.6.4 UC5.14 PUSH SERVICE EXECUTION MODEL
3.7. SERVICE DISCOVERY
3.7.1 UC1.5 - ASSIST THE TRAVELER
3.7.2 SPICE DISCOVERS A 3RD PARTY SERVICE WITH SEMANTIC DISCOVERY (TO WRITE)
3.8. PAYMENT
3.8.1 UC 1.6 - PURCHASE A MOVIE ON-LINE
3.8.2 UC7.3 – PURCHASE CONTENT FOR A THIRD PARTY
3.9. AUTHENTICATION
3.9.1 UC1.8 - SINGLE SIGN-ON
3.10. GROUP AND SOCIAL SERVICES
3.10.1 UC3.8 – GROUP FORMATION AND MANAGEMENT
3.10.2 UC3.9 - END USER USES A GROUP EMERGENCY SERVICE
3.10.3 UC7.7 – USE OF DIGITAL CONTENT

4. USER REQUIREMENTS

4.1. PROFILES AND POLICIES
4.2. IDENTIFICATION AND AUTHENTICATION
4.3. CONTEXT AWARENESS
4.4. SERVICE HANOVER
4.5. DYNAMIC DESKTOP
4.6. TERMINAL USAGE
4.7. SOFTWARE UPDATE
4.8. SERVICE CREATION AND USAGE

5. ENTERPRISE REQUIREMENTS

5.1. INTEROPERABILITY
5.2. MOBILITY AND ROAMING
5.3. BILLING AND CHARGING
5.4. MULTI-MODALITY AND MULTI-Terminal
5.5. SERVICE AND SERVICE PLATFORM
5.6. CONTENT DELIVERY
5.7. POLICY

6. TECHNICAL REQUIREMENTS

6.1. SYSTEM / BASIC ENABLERS
6.1.1 R8.1.1 - SYSTEM / BASIC ENABLERS INTEROPERABILITY 64
6.1.2 R8.1.2 – SYSTEM / BASIC ENABLERS DEPLOYMENT, ACQUISITION AND DISCOVERY 64
6.1.3 R8.1.3 – SYSTEM / BASIC ENABLERS AAA 65
6.1.4 R8.1.4 – SYSTEM / BASIC ENABLERS OPERATION & MAINTENANCE 65
6.1.5 R8.1.5 – SYSTEM / BASIC ENABLERS RUNTIME EXECUTION 66
6.1.6 R8.1.6 – SYSTEM / BASIC ENABLERS NON-FUNCTIONAL 66

6.2. INTELLIGENT AND CONTENT ENABLERS 66
6.2.1 R8.2.1 - INTELLIGENT AND CONTENT ENABLERS INTEROPERABILITY 66
6.2.2 R8.2.2 - INTELLIGENT AND CONTENT ENABLERS DEPLOYMENT / ACQUISITION / DISCOVERY 67
6.2.3 R8.2.3 - INTELLIGENT AND CONTENT ENABLERS AAA 67
6.2.4 R8.2.4 – INTELLIGENT AND CONTENT ENABLERS, OPERATION & MAINTENANCE 67
6.2.5 R8.2.5 – INTELLIGENT AND CONTENT ENABLERS RUNTIME EXECUTION 68
6.2.6 R8.2.6 – INTELLIGENT AND CONTENT ENABLERS NON-FUNCTIONAL 68

6.3. SERVICE CREATION 68
6.3.1 R8.3.1 – SERVICE CREATION INTEROPERABILITY 68
6.3.2 R8.3.2 – SERVICE CREATION DEPLOYMENT / ACQUISITION / DISCOVERY 68
6.3.3 R8.3.3 – SERVICE CREATION AAA 69
6.3.4 R8.3.4 – SERVICE CREATION OPERATION & MAINTENANCE 69
6.3.5 R8.3.5 – SERVICE CREATION RUNTIME EXECUTION 69
6.3.6 R8.3.6 – SERVICE CREATION NON-FUNCTIONAL 69

7. OPEN MARKET REQUIREMENTS 71

7.1. CHARGING, BILLING AND REVENUE SHARING REQUIREMENTS 72
7.2. USER PROFILE REQUIREMENTS 72
7.3. TERMINAL REQUIREMENTS 73
7.4. BUSINESS PROCESS SUPPORT REQUIREMENTS 73
7.5. INTEROPERATION REQUIREMENTS 73
1. Introduction

The objective of SPICE (Service Platform for Innovative Communication Environment) is to research, prototype and evaluate a scaleable overlay architecture and framework for rapid creation and deployment of intelligent and personalised services.

We intend with SPICE to develop an overlay structure that will enable large-scale service introduction through its support of multiple heterogeneous execution platforms. The SPICE platform will, on the development and operation side, manage the inherent complexity of the diverse communication systems while on the users side, it will assist the end user to control and enjoy the same communication technologies.

The rationale and the key objectives of SPICE are described in Chapter 2.

1.1. Purpose of this document

The purpose of this document is to present the key requirements defined for the SPICE platform in terms of user requirements, enterprise requirements, technical requirements and open market requirements. The requirements have been derived by analysing the different aspects of the SPICE service platform and by analysing a set of use cases relevant for the SPICE project.

The SPICE project is split into several work packages each addressing specific aspects of the SPICE service platform. Moreover, a set of service scenarios have been defined to guide the work on the requirements.

Within each work package a set of use cases has been defined which highlight the main functionality addressed by the work package. In addition, the work packages also produced a list of detailed requirements linked to the use cases and to the scenarios. The contributions of the work packages have been analysed and merged addressing the user, enterprise, technical and open market business requirements.

The SPICE project has identified three representative service scenarios (I-Portal, E- Tourism, E- Emergency) which together illustrate the whole set of indented SPICE features. Guided by the three scenarios, we have identified the basic requirements for the SPICE platform that will be fed into the architecture work.

1.2. Document Overview

The document is structured in the following way.

First, in Chapter 2 we describe the rationale and overall objective with SPICE. We will also introduce the method and the templates used for specifying the requirements.
In Chapter 3, we summarize the main use cases that we have extracted from the three SPICE scenarios and how they are mapped to the scenarios.

In Chapter 4, 5, 6, and 7 we respectively present an overview of the main user requirements, enterprise requirements, technical requirements and open market requirements.

In addition to the present document, five appendixes contain more detailed information on the requirements and terminology. The appendixes are located in a separate document organised as follows:

- Appendix A contains the list of the definitions of terminology and acronyms that we use in the SPICE project.
- Appendix B contains the detailed user requirements.
- Appendix C contains the detailed enterprise requirements.
- Appendix D contains the detailed technical requirements.
- Appendix E contains the detailed open market requirements.

1.3. Editors

The authors of this document are: Michaël van Bossuyt, Silvia Cabanillas, Jos Hendriks Claudio Riva and Robert Seidl. Input to this document has however been provided by many contributors in the project stemming from their work on specification of requirements within the different tasks and work packages.

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Update description</th>
<th>Doc. Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.3.2006</td>
<td>Claudio Riva</td>
<td>Version submitted for review</td>
<td>0.7</td>
</tr>
<tr>
<td>21.3.2006</td>
<td>Claudio Riva</td>
<td>Integrated changes from: Erwin Postmann (chapters 1-4), user requirements and technical requirements</td>
<td>0.8</td>
</tr>
<tr>
<td>21.3.2006</td>
<td>Claudio Riva</td>
<td>Fixed the document overview, moved the terminology to appendix</td>
<td>0.81</td>
</tr>
<tr>
<td>22.3.2006</td>
<td>Ernö Kovacs</td>
<td>WP5 UC and Req were missing. Added and sort into the right sections. Summary tables updated</td>
<td>0.83</td>
</tr>
<tr>
<td>22.3.2006</td>
<td>Claudio Riva</td>
<td>Small fixes</td>
<td>0.84</td>
</tr>
<tr>
<td>23.3.2006</td>
<td>Ernö Kovacs</td>
<td>Checked that all changes were still included in 0.84. Very minor fixes</td>
<td>0.85</td>
</tr>
<tr>
<td>24.3.2006</td>
<td>Claudio Riva</td>
<td>Included comments form Oddvar and Walter (only for the use cases section). Use cases moved back to Chapter 3. Appendixes will be taken away in the final section</td>
<td>0.86</td>
</tr>
<tr>
<td>30.3.2006</td>
<td>Claudio Riva</td>
<td>Updated all sections with the comments from reviewers</td>
<td>1.0</td>
</tr>
<tr>
<td>2.4.2006</td>
<td>Claudio Riva</td>
<td>Fixed some errors in the use cases section.</td>
<td>1.1</td>
</tr>
<tr>
<td>Date</td>
<td>Author</td>
<td>Change Description</td>
<td>Version</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
<td>----------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>27.4.2006</td>
<td>Claudio Riva</td>
<td>Moved the appendixes to a separate document</td>
<td>1.2</td>
</tr>
<tr>
<td>2.5.2006</td>
<td>Claudio Riva</td>
<td>Included changes from Oddvar and Michael</td>
<td>1.3</td>
</tr>
<tr>
<td>28.5.2006</td>
<td>Christophe Cordier</td>
<td>Fixed consistency of the document and added Odvvar’s changes</td>
<td>1.4</td>
</tr>
</tbody>
</table>
2. Specification of the SPICE System Requirements

In order to understand the system requirements for the SPICE platform we have identified three service scenarios which illustrate the expectations on the set of features that SPICE should handle. In addition we have defined a set of use cases that are described in Chapter 3.

In this chapter we first present the rational for SPICE in order to understand the context of the work addressed in this deliverable. Next, we describe the objectives set out for SPICE and how the defined scenarios maps to the strategic objectives of SPICE. Finally we define the templates used for specifying the use cases, the requirements and the basic terminology.

2.1. The Rational for SPICE

Mobile communications and Internet services have become an essential part of everyday life of European citizens. However, most end-users today consider mobile communication and Internet as totally separated ICT facilities. Voice telephony and SMS are the dominating communication services in mobile communication. Despite the good availability of broadband radio technologies, the business of mobile services has not yet taken off for the following reasons:

- Time to market for new services developments is too long due to a lack of suitable service creation environment and to a vertical design approach.

- Integration and deployment cost are too high due to the inherent complexity and heterogeneity of service execution environments.

- Service provisioning involves more and more parties - Telco, content / service providers, third party networks and service providers, and even end-users – increasing the complexity of the environment in which services must live.

- Users own many different communication devices and are surrounded by many access technologies but they usually cannot handle the complexity of accessing their services via several of these devices. In these cases, they look to the access and service providers for help.

- Specific constraints of mobile service provision (devices with small screen, limited computational capabilities and impractical keyboard) impose limitations to the type and size of content that can be delivered, thereby limiting the range of available services.

- Services and applications are currently responsible for performing their own content management functionality, and for obtaining and interpreting their own user profiles and context information. This not only results in incoherent functionality across different services, but also increases the time to market of intelligent applications and services because wheels are invented over and over again.

- Continuity of service from fixed to mobile access and seamless roaming of services across operators and networks is far from being a reality.
The SPICE project will act to these shortcomings in current service technology. SPICE will be developing a method for rapidly generating new services that hides the complexities of the communications environment and allows commercial services to be developed and deployed efficiently and economically.

The following short story illustrates the use of SPICE features for creation and deployment of mobile services by service providers:

2010 Chronicle at the “Just-a-click.com” service provider company.
John, an authority in mobile service creation, with a quite creative mind, is just starting a new working day in his company “Just-a-click.com”. He starts-up his SPICE mobile service creation graphical environment and the SPICE ontology service browser that provides him with a catalogue of services and components. He is hatching on profitable mobile applications with a wide user appreciation (remember, John is an expert). His strategy is to use special Telco components from the SPICE service platform, add some of his previously developed components, and maybe reuse some open-source components from the Internet. The Telco’s SPICE enablers at “competitor.com” are available to John because Just-ac-click.com has a commercial agreement on sharing components with “competitor.com”.

His SPICE service creation tool allows him to manipulate and configure the boxes (that represent the different building blocks from the service catalogue) and combines them into a new service component. After an hour of work this results in a complete service. The resulting product is quite impressive considering it has been created by reusing a couple of basic components and by putting in an hour of creative work to express his ideas. He knows that the basic components will be dynamically tied to real services, thanks to SPICE semantic service discovery mechanisms and orchestration features, taking into account what is available locally to the user. The service will follow the user seamlessly wherever he goes and it will optimize the user’s communication means, making use of SPICE service components that have these capabilities.

The rest of the creation process is facilitated by the SPICE platform. John is going to start the deployment of his component onto the “just-a-click” SPICE mobile service platform using the “deploy it!” button.

The SPICE platform automatically registers the new components to the “just-a-click.com” service repository and they can now be discovered automatically by any user or be distributed by any TelCO sharing components under the SPICE model. Once the new service is deployed on the SPICE platform, the end-user will benefit from intelligent context-aware and personalised service provided by SPICE platform enablers.
It was a good day for John!

The story illustrates some immediate benefits of the SPICE platform when applied to the service creation process: easy and quick service creation, and dynamic service instantiation and deployment. It also highlights some of the tangible results that will come out of the SPICE project in the service creation area such as integrated service creation environment, graphical service creation tool, and advanced service composition mechanisms. Other tangible results from SPICE implied by the story are seamless service and information roaming, opening-up to 3rd parties for sharing of service components, and automatic discovery of new services.
2.2. SPICE High-Level Objectives

The SPICE project will turn today’s confusing complexity and heterogeneity into a managed, rich service environment by exploiting the diversity of device capabilities and fostering service adoption. In addition, the SPICE approach will broaden business opportunities in the communications and associated industry sectors. Simple and low-risk service introduction features will support SME demands.

The key SPICE project objectives are to:

- **Provide an easy and simple way to create and roll out innovative services to reduce development time and introduction costs and risks.**

- **Provide a unified way to seamlessly deliver services over heterogeneous execution platforms, networks and terminals.**

- **Enrich the service landscape, through an overlay structure supporting the users and offering a personalised user experience anytime, anyplace.** The service platform will provide appropriate service enablers distributed onto service platform components and mobile devices allowing the delivery of smarter context-aware services, providing the user with meaningful access to personalised services anytime, anyplace and delivering content adapted to the user’s communication sphere, access technology and personal situation:
  
  - **Create a trusted and open platform that will simplify the use of services, devices through personalisation & customisation.** SPICE will revisit communication paradigms and promote communication with persons as well as authorised devices. Services will be available on a multitude of terminals, allowing the user to choose the most appropriate communication means.

- **Enrich current service platform functionality** with content management and distribution features and intelligent service-controlled context-information processing.

- **Open-up to new business models and value chains.** The flexibility of the SPICE platform environment will specifically support the demands from TelCos, SMEs and 3rd party service developers and providers for open service platforms.

- **Enabling Pan-European service provisioning.** Mobility-aware capabilities in the service platform environment will enable applications and services to roam seamlessly across commercial and country boundaries

- **Promoting the uptake of innovative IT software technologies in a telecommunications grade service platform environment.**

SPICE will deliver technical solutions that provide simplicity and convergence at the service layer to enrich the service landscape and to ensure a faster deployment and adoption of new services.

2.3. Specification of the requirements

We have identified three representative scenarios with the intention that these together illustrate the set of expected SPICE features and that the scenarios can be used to guide the
specification of the system requirements and later on act as reference for the work undertaken in the different work packages of the project. In addition selected sub-sets of these scenarios will be used for the probes and demonstrations tasks during the project.

To illustrate how the scenarios maps to the expected SPICE features, we have positioned the scenarios onto a diagram depicting the main strategic objectives of SPICE. The following diagram in Figure 1 depicts the main strategic objectives of SPICE.

![Figure 1: SPICE Strategic objectives](image)

The three scenarios identified will be described in detail in the SPICE Deliverable D1.2. The illustrations below show how each scenario addresses the main strategic objectives of SPICE.

**Intelligent Portal: i-Portal**

In this scenario users will be assisted by an “Intelligent Portal” that proposes services adequate to the location, the context and user preferences. It mainly focuses on service
roaming, seamless delivery, security and content adaptation as indicated by the diagram below.

![Intelligent Portal Diagram](image1)

**E-tourism**
E-tourism actually consists of a set of scenarios that illustrate how users may be assisted when travelling. The scenario mainly focuses on intelligent service enablers, user assistance and service creation tools as shown in the diagram below.

![eTourism Diagram](image2)

**E-emergency**
This scenario shows how the user’s Distributed Communication Sphere can be optimally used in order to handle intelligently emergency situations. Focus is on Multi-modality, dynamic desktop, service push and attentive services.
In order to design the SPICE platform as a system, we must collect and analyze the basic requirements that this system must fulfil (system requirements). The system requirements can be derived by analyzing the functionality that the system will have to fulfil. We capture the main functionality of the system through a set of use cases. Use cases specify the behaviour of the system under various conditions as it responds to a request of one stakeholder. A use case mainly describes the interaction of the system with the stakeholders and how the various interests are preserved. The use cases represent and describe the main **functional requirements** of the system.

Not all the requirements can be expressed with use cases. **Technical or quality requirements** describe additional constraints on the behaviour of the system that must be considered during its design. Each additional requirement must be linked to the main functionality of the system (to one or more use cases) and needs to be clearly justified.

In this document we first define the main functionality of the system through a set of use cases focused on the user/enterprise/open market requirements. Then, we define the additional requirements (technical or quality) and link them to the use cases. The outcome will serve as a basis to derive the system requirements.

### 2.4. The body parts for the use case template

A use case (UC) describes the **behavioural contract among the stakeholders** of a system under discussion. It describes what a system should do under various conditions as the system responds to a request of one stakeholder, called the **primary actor** who has one goal to achieve. The primary actor initiates the UC and the system responds protecting the interests of all the stakeholders. The UC contains all the different sequences of behaviour that can unfold. The **main success scenario** describes the typical situation when there are no failures. The **extensions** handle the failures or the alternative paths. Whatever the UC succeeds or fails, the **minimal guarantees** are satisfied. The system under discussion can range from a component to a subsystem, to a system or the whole organization.

The following template is used to represent use cases:
UC <WP>,<Use Case number> – Title of the use case

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
<th>System</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
<td>User</td>
<td>Subfunctional</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stakeholders:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preconditions:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Success guarantees</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Main Success Scenario

Extensions

Contributor:

**Use Case ID:** the naming convention has the following syntax:

UC<WP>,<Use Case number>

Where <WP> is the work package that created the use case and <Use Case number> is a progressive number that is unique within the workpackage. (Example: UC7.5 is the 5th use case created by the WP7).

**Use Case Title:** an active verb phrase describing the goal of the primary actor (e.g. Buy a book online)

**Scope:** The boundaries of the system under discussion. Anything inside the boundaries is considered a black-box. The primary actor is outside of the scope. We define three types of scope:

- **Enterprise:** for UCs discussing the behaviour of the entire organization;
- **System:** for UCs discussing the behaviour of the SPICE platform or a network infrastructure;
- **Subsystem:** for UCs discussing the behaviour of some components with SPICE

**Level:** the level of details for describing every sentence of the UC. We define three levels:
• \textit{Summary}: describing the life cycle sequence of multiple user goals (e.g. get paid for a car accident);
• \textit{User}: a significant single goal for the primary actor (e.g. register into the system);
• \textit{Subfunctional}: detailed steps for achieving the user goals.

\textbf{Tips}: Raise the goal level by asking the “why” question and lower the goal level by asking the “how” question.

\textbf{Primary Actor}: the person, organization or computer system with behaviour that has a goal on the system under design (the primary actor is outside of the system in the scope). Example: clerk, sales manager, customer.

\textbf{Stakeholders}: who has interests to protect or cares about the outcome (primary actors are stakeholders with clear goals to achieve). Examples: network operator, bank, service provider.

\textbf{Preconditions}: what the system ensures is true before the letting the use case to start.

\textbf{Postconditions}: what can be assumed to be true after the use case has been successfully completed.

\textbf{Trigger}: what business situation or even causes this UC to start.

\textbf{Minimal guarantees}: the fewest promises that the system makes to the stakeholders when the goal cannot be achieved (e.g. Order will be initiated only if payment received).

\textbf{Success guarantees}: what interests of the stakeholders are satisfied after the successful conclusion of the UC (e.g. the claimant will be paid, the claim is closed).

\textbf{Main Success Scenario}: description of one easy-to-understand and typical scenario where the actor’s goal is achieved and the interests of the stakeholders are preserved. The scenario consists of \textit{3 to 10 numbered steps} or actions of the primary actor. Verbs are in the active form (e.g. actor1 \textless{verb}\textgreater{} actor2 with some data). No UI descriptions are provided. The scenario continues without any failure till when all the interests of the stakeholders are satisfied. Example:

1. Customer selects the books on the web
2. Bookstore confirms that the books are on stock
3. Customer enters the credit card details
4. Bookstore validated the credit card number with the credit card company
5. Customer confirms the order
6. Bookstore creates the delivery order
7. Customer received the goods

\textbf{Extensions}: describes failures or alternative paths to the main successful scenario. For every branch you must specify the extension condition (what went wrong and the system takes a different behaviour than the successful one) and the extension handling (the sequence of steps
to deal with the condition). The extension either terminates the UC or rejoins the main scenario. The extension takes care that the interests of the stakeholders are preserved in the case of failure. Example:

2a. Not in stock
   2a1. Bookstore informs the purchaser
4a. Wrong credit card number
   4a1. Bookstore ask to re-enter the CC number

Additional requirements: list of all additional requirements including: quality requirements, protocols, performance, User interface design, business rules, data formats.

2.5. Body parts for the requirements template

<table>
<thead>
<tr>
<th>ID</th>
<th>R&lt;WP#&gt;,&lt;Task&gt;,&lt;Req number&gt; - &lt;title&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert:</td>
<td></td>
</tr>
<tr>
<td>WP Source:</td>
<td>Links to use cases or scenes in the scenario</td>
</tr>
<tr>
<td>Priority:</td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td></td>
</tr>
<tr>
<td>Rationale:</td>
<td></td>
</tr>
<tr>
<td>Dependency (optional):</td>
<td></td>
</tr>
<tr>
<td>Environment (optional):</td>
<td></td>
</tr>
<tr>
<td>Other (optional):</td>
<td></td>
</tr>
</tbody>
</table>

**Requirement ID:** the naming convention has the following syntax:

R<WP#>,<Task>,<Req number> - <title>

Where <WP> and <Task> are respectively the work package and the task that created the use case, <Req number> is a progressive number that is unique within the task. The <title> is the title of the requirement.

(for example, R6.4.5 is the 5th requirement identified with the task 6.4 in WP6).

**Expert:** the contact person that created the requirement

**Priority:** the importance of the requirement. The choices are: low (nice to have), medium (would support other parts a lot), high (it will be difficult, but possible, to implement other parts if this requirement is not satisfied) and mandatory (this requirement must be implemented and without it, the whole cannot be implemented).

**Description:** the detailed description of what the actual requirement is.
**Rationale**: the reason why the requirement is necessary.

**Dependency**: a reference to another requirement, or a set thereof, together with a possible short explanation.

**Environment**: external dependencies, outside of the system. In the case of technical requirements this may be "Java Virtual Machine" or "object-oriented database management system". In the case of user requirement this may be "user is member of a family" or similar. If necessary, also a short explanation should be given here.

**Other**: other remarks or notes concerning this requirement.

### 2.6. Terminology for the requirements

The keywords "must", "must not", "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" in this document are to be interpreted as described in RFC 2119.

1. MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

2. MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

3. SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

4. SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

5. MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)
3. Use Cases

In the following table we summarize the use cases and their mappings to the scenes in the scenario. Each use case represents a piece of functionality that is necessary to implement some parts of the scenarios. Each use case represents a requirement that the SPICE platform shall implement.

We have identified 45 use cases categorized in 12 groups. Below are listed the 12 categories of use cases with a brief summary:

1. **Service handover**: the SPICE platform must provide the functionality for handing over a service from one device to another one and for maintaining the continuity of service. Examples are: an active video call that is transferred from a mobile device to the home device (UC1.1), or a streaming service that moves across PC/mobile device/media centre. The service handover is also used for obtaining the highest benefits from the available devices around the user (UC7.2).

2. **Device management and configuration**: the SPICE platform must provide the basic functionality for backing-up/restoring the users’ data (profiles and preferences) and reinstalling the services in the case of device change (UC1.3 and UC3.5).

3. **Subscription and configuration of services**: the SPICE platform must provide the ability of installing/subscribing to new services (UC3.1 and UC 3.6) and to configure the format for the delivery of services (UC7.5).

4. **Context awareness**: the SPICE platform must provide a set of intelligent functionalities for understanding the context of the user (e.g. location; UC 3.2) and delivering targeted services (UC 3.3, UC3.4 and UC1.4). The context information can also be used for adapting service delivery (UC7.1) or for emergency situations (UC7.4).

5. **Service lifecycle**: the SPICE platform must provide the functionality to allow network operators (UC5.04), 3rd parties (UC5.10) and end-users (UC5.05) to build and deploy new applications (UC 1.7) and new services (UC 3.7) and for the automatic update of services (UC7.6). The platform must also support the multi-modal delivery of services (UC3.11). A special case is composing a new service from existing ones (UC5.13) Deploying of a service is related to the Service Lifecycle (UC5.06) and the provisioning of content used in the service (UC5.12).

6. **Service discovery**: the SPICE platform must allow the end-user to discover available services (UC1.5).

7. **Profiles and preferences management**: the SPICE platform must provide the necessary mechanisms for the management of the different user / service profiles and the preferences of the user for the subscribed services.

8. **Payment**: the SPICE platform must provide a payment mechanisms for online content (UC 1.6 and UC 7.3)
(10) **Authentication**: the SPICE platform must provide a single sign-on authentication mechanism (UC1.8).

(11) **Group and social services**: the SPICE platform must provide functionalities for the creation and management of groups for social applications (UC3.8) and emergency situations (3.9) and the use of digital content (7.7).

(12) **Service monitoring**: the SPICE platform must provide mechanisms for the monitoring of the actual status of the offered services, to inform the administrator, service provider or user about the actual status, e.g. in case of exceptional situations.

The table below lists the use cases headers and the mapping to the scenarios. The use cases not tagged with an identification number have only been identified but their definition has not yet been filled in this document as further work in needed to address them properly.

<table>
<thead>
<tr>
<th>Use case</th>
<th>I-Portal</th>
<th>e-Tourism</th>
<th>E-Emergency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(1) Service Handover</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC1.1 - Communication session following user</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC1.2 - Watching movies</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC3.10 - Changing device</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC7.2 - Content Adaptation According to Available Terminals</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(2) Device Management and Configuration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC1.3 - Buying a new phone</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC3.5 - Device Management</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>(3) Subscription and configuration of services</strong></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>UC3.1 - End user installs a Dynamic Address Book application</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC3.6 - User subscribes to a new service</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC7.5 - Specify Content Delivery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(4) Context Awareness</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC3.2 - Automatic updated of reachability</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>User context used by 3rd SP service</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC3.3 - Cost optimized connection establishment</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC3.4 - Enhancing Service Delivery using Reachability</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>UC1.4 Finding attractions</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC7.1 Content Adaptation According to User Position</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UC7.4 – Priority of SPICE Platform in dangerous situations</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ID: SPICE_WP1_D1_1_v1.4_clean.doc**

**Revision: 1.4**  
**Date: 02/05/06**  
**Security: PUBLIC**  
**Page 19/74**
<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC7.6</td>
<td>Terminals actualization and reconfiguration</td>
</tr>
<tr>
<td>UC5.4</td>
<td>Service Developer creates a new service</td>
</tr>
<tr>
<td>UC5.5</td>
<td>End user creates a new service</td>
</tr>
</tbody>
</table>

**Service Discovery**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
<th>Requirement Met</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC1.5</td>
<td>Assist the traveller</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Provide most appropriate service to end-user</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>SPICE discovers 3rd party service with semantic discovery</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Service Execution Model**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
<th>Requirement Met</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC5.1</td>
<td>Invoke a session communication service</td>
<td>✓</td>
</tr>
<tr>
<td>UC5.2</td>
<td>Service using the IT7Web Service execution model (Click-to-Dial)</td>
<td>✓</td>
</tr>
<tr>
<td>UC5.3</td>
<td>Service using the Messaging execution model</td>
<td>✓</td>
</tr>
<tr>
<td>UC5.14</td>
<td>Push Service execution model</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Profile and preferences management**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
<th>Requirement Met</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Profile and preferences used by 3rd SP</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Preferences management</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Payment**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC1.6</td>
<td>Purchase a movie on-line</td>
</tr>
<tr>
<td>UC7.3</td>
<td>Purchase Content for a Third Party</td>
</tr>
</tbody>
</table>

**Authentication**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC1.8</td>
<td>Single sign-on</td>
</tr>
<tr>
<td>UC3.8</td>
<td>Group formation and management</td>
</tr>
<tr>
<td>UC3.9</td>
<td>End user uses a group emergency service</td>
</tr>
<tr>
<td>UC7.7</td>
<td>Use of Digital Content</td>
</tr>
</tbody>
</table>

**Group and social services**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
<th>Requirement Met</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sharing and posting to a group of interest</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Finding recommendations</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Service monitoring**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Monitoring that service is always running</td>
</tr>
<tr>
<td></td>
<td>Acknowledgment by end-user</td>
</tr>
<tr>
<td></td>
<td>Service events log</td>
</tr>
</tbody>
</table>

The table above summarises the present identified use cases for SPICE. In general, the number of use cases may increase as we study requirements in more detail in the different tasks and workpackages. We have therefore defined a process for capturing new use cases and requirements and these will be documented in an update of the SPICE requirements (deliverable D1.5 due M18).

The Description of Work also identified the need to investigate the impact of multi-language support on the architecture of the platform. At present the requirement on multi-language support has not been fully investigated, i.e. what impact this will have on the architecture of the SPICE platform. This requirement will also be investigated in connection with service
roaming which is defined as one of the key objectives of SPICE (see Section 2.2: enabling pan-European service provisioning). Service roaming will be investigated in several WPs and will be addressed in forthcoming deliverables such as D1.3 on SPICE architecture. Thus, some new use cases may be identified in the process of dealing with the multi-language support and service roaming requirements.

3.1. Service handover (continuity of service)

3.1.1 UCL.1- Communication session following user

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Service Provider (SP)</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Mobile user</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>A mobile user makes a communication session (e.g. audio call, video call)</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>A mobile user decides to make use of devices available in his vicinity for continuing his communication session (e.g., audio call, video call)</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

Main Success Scenario

1. User enters his home while having his video call with his friend.
2. Service platform detects his presence in his home
3. Service platform authenticates user
4. Service platform sends the list of available devices in his home, which are have the required capabilities and not yet occupied
5. User selects device(s) for continuing his video call
6. Service platform authorizes whether he is allowed to make use of device(s) requested by the user
7. Service platform transfers the call to selected local device(s)
8. User continues talking with his friend on the local device

Extensions (ctrl+tab for indenting the actions)

3a. User is not the authenticated person for making use of device(s) at the location
   3a1. Service platform does not send the list of available devices
   3a2. User continues talking on his mobile device with his friend
5a. User does not select any devices from the list
   5a1. Step 6-8 will not be processed
   5a2. User continues talking on his mobile device with his friend
6a. User is not authorized to make use of device(s)
   6a1. Step 7 and 8 will not be processed
6a2. User continues talking on his mobile device with his friend
7a. Service platform could not transfer the call to the target device(s)
   7a1. Step 8 will not be processed
   7a2. The call is returned to the mobile device
   7a3. User continues talking on his mobile device with his friend

3.1.2 **UC1.2 – Watching movies**

<table>
<thead>
<tr>
<th><strong>Scope:</strong></th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level:</strong></td>
<td>Summary</td>
</tr>
<tr>
<td><strong>Primary Actor:</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Stakeholders:</strong></td>
<td>Spice operator, user, movie provider</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>User has Spice account</td>
</tr>
<tr>
<td><strong>Postconditions:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Minimal guarantees:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Success guarantees</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. User enters the Internet shop with movie viewing facilities
2. User logs in to the Spice-portal
3. Portal authenticates user
4. User navigates to movie service
5. User selects movie to watch
6. User selects the device on which he/she wants to watch the movie
7. User starts watching the movie at the Internet shop
8. User leaves the Internet shop and continues to watch the channel using his/her handset
9. User arrives at home and continues viewing the movie on his home TV
10. User gets bill from mobile operator

**Extensions** (ctrl+tab for indenting the actions)

The same extensions as for UC1.1. applies also for this use case

Contributor:

3.1.3 **UC3.10 – Changing device (split content streams to different devices)**

<table>
<thead>
<tr>
<th><strong>Scope:</strong></th>
<th>System</th>
</tr>
</thead>
</table>

ID: SPICE_WP1_D1_1_v1.4_clean.doc  Date: 02/05/06
Revision: 1.4  Security: PUBLIC
Page 22/74
<table>
<thead>
<tr>
<th>Level:</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Actor:</strong></td>
<td>SPICE Platform</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>users (Philippe and Karin's kids, in this case), operator of the SPICE platform, service and content providers</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>kids are viewing a movie on a TV screen in an Internet shop. The movie is being streamed by a content provider.</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>kids are viewing the same movie on in-car displays and use their wireless headsets for the movie's audio component</td>
</tr>
<tr>
<td>Trigger:</td>
<td>context change (kids moving from the Internet shop into their parents' car)</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. Kids move from the Internet shop into their parents' car  
2. The SPICE platform authenticates the kids and authorizes them to use the on-board displays in the back of the front seats  
3. The SPICE platform discovers the devices, networks, and services that are available in the car  
4. The SPICE platform seamlessly redirects the video part of the movie from the TV screen at the Internet shop to the on-board displays  
5. The SPICE platform seamlessly redirects the audio part of the movie from the TV screen at the Internet shop to the kids' wireless headsets

**Extensions** (ctrl+tab for indenting the actions)

The same extensions as for UC1.1. applies also for this use case

| Contributor: | Cristian Hesselman, Telematica Instituut (TELIN) |

### 3.1.4 UC7.2 - Content Adaptation According to capabilities of other Available Terminals

<table>
<thead>
<tr>
<th>Level:</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Actor:</strong></td>
<td>SPICE Platform</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Users</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The user is using a SPICE service on his mobile terminal when he arrives home and is presented the possibility of translate such service to another detected device</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>The user can go on using the desired service without interruption and in an user friendly way</td>
</tr>
<tr>
<td>Trigger:</td>
<td>The user arrives in a place where other terminals that can give best</td>
</tr>
</tbody>
</table>
performances are detected.

Minimal guarantees: The user goes on using the service on his mobile terminal

Success guarantees The user passes the service on a terminal that gives him best performances

### Main Success Scenario

1. The End User, using her mobile equipment, asks for and obtains (from SPICE system), a general service (video, audio and so on).
2. He is using such service when he arrives in a place where exists the opportunity of using another terminal, that can give best performances.
3. The user is advised about this possibility and is prompted to take a decision about passing or not to the new mobile terminal.
4. The SPICE service is redirected to the new terminal and his characteristics are adapted to the new environment (screen resolution, sound quality and so on).

### Extensions (ctrl+tab for indenting the actions)

2a. The mobile terminal advises the user that exists the possibility of using a new terminal that can give him best performances.
   2a1. the SPICE platform detects the new terminal according to the position of the user.
   2a2. the mobile terminal detects new terminals.

3a. The user is prompted to decide about passing or not to the new device
   3a1. the user decides to pass and go on using the service on the new device detected.
   3a2. the user rejects such possibility and goes on using the service using the mobile terminal.

Contributor: Telefónica I+D

---

### 3.2. Device Management and Configuration

#### 3.2.1 \textit{UC1.3} - Buying a new phone

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Content Provider, Service Provider, Operator</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The user is a registered user with a service profile. Some components of the service portfolio requires some specific features on the mobile phone to work. The user has only a subset of the available services due to an older mobile phone model. Aspects like</td>
</tr>
</tbody>
</table>
billing are omitted for now.

| Postconditions: | The existing services will work as before on the new device, and the user can now use more of the offered service portfolio |
| Trigger: | An ADD (Automatic Device Detection) trigger notifies the system that the user has upgraded his/her mobile phone |
| Minimal guarantees: | The user will always be in full control of what to install and what not to install. The user will control when the operation takes place, and can abort or postpone the operation |
| Success guarantees | The content and service providers are sure that the user has access to all their services, and there will be less downtime after device change. The operator will also not lose traffic during the time that the user has lesser or malfunctioning services on the device. |

**Main Success Scenario**

1. Fred inserts his SIM card into his new mobile phone and turns it on
2. Fred receives a notification that asks if he wants to transfer his existing services to his new phone
3. He selects 'OK' and receives all relevant software and configurations that he had on the previous phone, now customized to fit his new phone
4. He then receives a notification that says that he can now use more advanced features from his Intelligent Portal, and a question if he wants to install and configure those as well
5. Fred selects 'OK' to install and configure the new features, that are adapted to both his phone and his user profile
6. Fred is now happy because there was no hassle related to changing phone, and he now can use some new and exiting features from the Intelligent Portal

**Extensions** (ctrl+tab for indenting the actions)

3a. Fred selects 'Cancel'
   3a1. Operation is aborted
3b. Fred selects 'Postpone'
   3b1. Notification will be resent the next day or after a selected period
5a. Fred selects 'Cancel'
   5a1. Operation is aborted
5b. Fred selects 'Postpone'
   5b1. Notification will be resent the next day or after a selected period

| Contributor: | Anne Marte Hjemås, Telenor |

### 3.2.2 UC3.5 - Device Management

| Scope: | Enterprise |
| Level: | Summary |
| Primary Actor: | End User |
| Stakeholders: | Communication Service Provider |
Preconditions: 

Postconditions: 

Trigger: The End User changes mobile terminal and Operator must configure the new device.

Minimal guarantees: 

Success guarantees 

Main Success Scenario 

1. The End User changes her mobile terminal. 
2. The SPICE system restores all needed Service configuration (for example, access network configuration, communication service configuration, dynamic desktop configuration, etc.). 
3. The SPICE system restores all needed applications the user may have installed on her device (for example, Games, or mobile portals like Yahoo! Go, etc.). 
4. The SPICE system restores all needed End User’s personal data (for example, address book, ring tones, music, videos, events, etc.). 

Extensions (ctrl+tab for indenting the actions) 

2a. The End User sets rules about a partial restore of non standard services. 
   2a1. The SPICE system restores only the services that are relevant, according with user’s rules and context. 

4a. The End User sets rules about a partial restore of personal data. 
   4a1. The SPICE system restores only the data that are relevant, according with user’s rules and context. 

Contributor: Armelle Clech, France Telecom 

3.3. Subscription and configuration of services 

3.3.1 UC3.1 - End user installs a Dynamic Address Book application 

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>End User is just authenticated and attached to wireless access network.</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>The End User manages an advanced personal address book, using her advanced mobile equipment.</td>
</tr>
</tbody>
</table>
Minimal guarantees:  | New client application is installed on End User’s mobile terminal. The End User can configure the dynamic address book in off-line mode.
---|---
Success guarantees | The address book is synchronized and backed up to the SPICE system. End User and her contact can mutually (and dynamically) see their availability.

### Main Success Scenario

1. The End User, using her mobile equipment, downloads (from SPICE system) a new client application providing an “advanced personal address book”.
2. The End User installs and launches that client application on her mobile equipment.
3. The End User creates some address book “groups” (for example, family, colleagues, friends, etc.), allowing contacts categorisation.
4. The End User inserts her “contacts”, filling her address book.
5. The SPICE system synchronizes and backs up the address book to the Communication Service Provider’s network.
6. The End User sets her personal availability (or willingness to communicate).
7. End User’s availability is shown to her contacts (using the same “advanced personal address book” client application or using other compatible compliant IMS SIP/SIMPLE client).
8. Reciprocally, the End User can see the availability of contacts in her address book.
9. Any “availability” information change is dynamically updated into the SPICE system and refreshed at the watcher’s display.

### Extensions (ctrl+tab for indenting the actions)

3a. Dynamic Address Book client application just executed.
   3a1. A reference version of address book is retrieved from the SPICE system and synchronized with the local information.

5a. Mobile equipment is out of coverage
   5a1. Data synchronisation will be triggered when the Mobile equipment will be in network coverage

7a. Client application not installed to contacts/watcher’s mobile terminal
   7a1. Potential watchers are notified (for example, by sms) about new client application.
   7a2. The contact’s reference (for example, telephone number, e-mail) is updated considering the End User localisation and context.

**Contributor:** Antonio Vitale, Alcatel
3.3.2  UC3.6 - User subscribes to a new service.

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The End user uses 2 kinds of mobile terminals a PDA under Windows Mobile 0S and a mobile supporting J2ME</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>On her mobile terminal, the End user selects a new application she wants to use and acknowledges the update of her profile</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The end user subscribes to a notification service in order to be aware of the availability of new application in one of her domain of interest (weather forecast for example)
2. When notified the end user acknowledges and customizes the access to the application (preferred city for the weather for example)
3. A new dedicated icon is presented on both of her terminals desktops (for one click access)

The end user profile is updated

**Extensions** (ctrl+tab for indenting the actions)

1a. The end user selects and customizes the access to a new application by using a repository browsing service. The browsing is based on requests using semantic and ontology. The end user profile is updated

Contributor: Patrick Fontaine, Alcatel

3.3.3  UC7.5 – Specify Content Delivery

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>SPICE Platform</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>A SPICE user can configure his profile specifying the way the</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>The information is always delivered according to user profile</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Trigger:</td>
<td>A SPICE user asks for a service whose delivery format is different from the one established in the user profile</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td>In emergency situations the content must be adapted according to user necessity</td>
</tr>
<tr>
<td>Success guarantees</td>
<td>Every kind of content is successfully adapted to user necessity (e.g. if a user has configured “no audio” option he will be able to see a film using subtitles automatically)</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. A SPICE user asks for a general service, which can be video, audio or text and images.
2. The User specified in his profile that he wants to receive every content in a determined format (that can be video with subtitles, audio and so on).
3. The SPICE platform analyzes the user profile when the user asks for a service and negotiates with the content provider the format of the content that will be delivered or analyzes if the same platform can translate the content format.
4. The user receives the information according to the profile he configured.

**Extensions** (ctrl+tab for indenting the actions)

3a. The content provider can’t deliver the information in the desired format or the SPICE Platform is unable to translate the contents.
   3a1. another content provider with the specified characteristic is searched.
   3a2. the user is notified about the impossibility of delivering the content in the required format.

| Contributor: | Telefónica I+D |

### 3.4. Context awareness

#### 3.4.1 UC3.2 - Automatic updated of reachability

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td><strong>Primary Actor:</strong></td>
<td>End User</td>
</tr>
<tr>
<td><strong>Stakeholders:</strong></td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Postconditions:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>The SPICE system automatically updates the End User’s</td>
</tr>
</tbody>
</table>
reachability information (availability, context, cost and bandwidth of access network) when she moves through heterogeneous wireless networks (GPRS, 3G, WLAN, etc).

<table>
<thead>
<tr>
<th>Minimal guarantees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success guarantees</td>
</tr>
</tbody>
</table>

### Main Success Scenario

1. The End User, using her mobile equipment, downloads (from the SPICE system), installs and runs a new client application providing an “advanced personal address book”.
2. The End User, using her mobile equipment, downloads (from the SPICE system), installs and runs a new client application that manages her “reachability” information.
3. The End User moves through heterogeneous wireless networks (GPRS, 3G, WLAN, etc) using her advanced PDA terminal.
4. The End User enters into a relevant and preferred wireless access network coverage.
5. The End User is transparently authenticated and attached to the access network.
6. Personal and system information (user’s availability and context, outgoing call cost and access’s bandwidth) associated with that access network are automatically updated, according to user’s rules and settings.
7. Current data values are refreshed for the End User herself, synchronized with the SPICE system and shown to the End User’s contacts (as Watchers).
8. Personal and system information are used (by the End User, the Watchers or the SPICE system) to track the End User, to take optimal decisions or to trigger services.

### Extensions (ctrl+tab for indenting the actions)

**a. Mobile equipment is out of coverage**

   **a1.** Data synchronisation will be triggered when the Mobile equipment returns to network coverage

**4a.** The newly discovered network is not relevant (the End User is not interested in)

   **4a1.** The new access network is ignored and putted in a “blacklist”.

**4b.** The newly discovered network is not preferred to current network (following user’s criteria)

   **4b1.** The user’s terminal stays attached to the preferred access network.

**5a.** The network is discovered for the first time (access network is “unknown” to the End User)

   **5a1.** The Client application discovers the new network and opens a pop up to the End User to configure the new access.

   **5a2.** The End User sets the access network’s parameters (for example, hot spot SSID, WEP key, network’s relative preference, cost, bandwidth) and personal parameters (like context).

   **5a3.** The new access network is put in a list of relevant and “known” networks.

   **5a4.** The Client application synchronizes the data with the SPICE platform.

   **5a5.** The user’s terminal attaches to the preferred access network.
5b. The SPICE platform proposes to the user to install (or update) a Client application to bypass future potential network’s ambient hostility (e.g. network constraints like NAT, Firewall, etc.)
   5b1. The End User acknowledges the software download.
   5b2. The SPICE platform remotely configures the terminal and downloads necessary software/components to solve ambient hostility and to efficiently protect communications.

7a. End User configures privacy rules about her reachability information
   7a1. End User enters rules to show reachability only to a select number of contacts from her buddy list.
   7a2. End User enters rules to show reachability to “watchers” as function of contact’s group membership (declared in the Address Book).

8a. Client application not installed on contacts/watcher’s mobile terminal
   8a1. Potential watchers are notified (for example, by sms) about new client application.

Contributor: Antonio Vitale, Alcatel

Additional Requirements:

# - <Advanced mobile terminal>
   Description:
   As mobile terminal, a nomadic End User MUST use, at least, an advanced mobile PDA terminal with multiple access capability (for example, GPRS, 3G, WLAN, Bluetooth, etc …) using a mobile OS (for example, Windows Mobile, Symbian, etc …).

   Rationale:
   To allow the nomadic End User to move through heterogeneous wireless networks using an advanced, smart and light terminal.

   Reference:

# - <Reference mobile terminal>
   Description:
   SPICE system SHOULD use End User’s personal mobile phone terminal as reference to retrieve her context.

   Rationale:
   To track the End User’s reachability in a natural way, using the personal terminal that is always in her pockets.

   Reference:

3.4.2  UC3.3 – Cost optimized connection establishment
<table>
<thead>
<tr>
<th>Level:</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td></td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>The End User wants to establish a cost optimized communication.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees</td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The End User, using her mobile equipment, downloads (from the SPICE system), installs and runs a new client application providing an “advanced personal address book”.
2. The End User, using her mobile equipment, downloads (from the SPICE system), installs and runs a new client application that manages her “reachability” information.
3. The End User moves through heterogeneous wireless networks (GPRS, 3G, WLAN, etc) using her preferred advanced PDA terminal.
4. The End User verifies that the person (contact) that she would like to call is “available”.
5. The SPICE system checks the possibilities for a cost optimized connection between user and person (contact) i.e. the cheapest route and informs the user about the outcome.
6. The End User decides to call the contact using the proposed connection.

**Extensions** (ctrl+tab for indenting the actions)

4a. The contact is not available (for example, in “off line ” or “do not disturb” status)
   4a1. The End User decides not to make the phone call.
   4a2. The End User decides to postpone the phone call, waiting for the contact’s availability.
   4a3. The SPICE system will trigger an automatic call back (on “avaibility” basis).

5a. The calculated costs of the connection is expensive (“high”)
   5a1. The End User decides not to make the phone call.
   5a2. The End User decides to postpone the phone call, moving in a cheaper network coverage.
   5a3. The SPICE system will trigger an automatic call back (on “minimum cost” basis).

**Contributor:** Antonio Vitale, Alcatel
3.4.3 UC3.4 - Enhancing Service Delivery using Reachability

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>e-mail delivery Service</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td></td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>An e-mail delivery Service wants to deliver an e-mail with a large attachment, according to user preferences.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The End User, using her mobile equipment, downloads (from the SPICE system), installs and runs a new client application that manages her “reachability” information.
2. The End User moves through heterogeneous wireless networks (GPRS, 3G, WLAN, etc) using her preferred advanced PDA terminal.
3. The SPICE system monitors the current access network bandwidth.
4. An e-mail delivery Service asks the SPICE system the permission, according to users rules, to deliver a pending e-mail with a large attachment.
5. An e-mail delivery Service delivers the e-mail with the large attachment.

**Extensions** (ctrl+tab for indenting the actions)

4a. The current bandwidth is “low”.
   4a1. The e-mail delivery Service delivers “small” e-mails and only the subject of “large” email.
   4a2. The e-mail delivery Service will deliver the attachments of large e-mails only when:
      1) the SPICE system notifies that the current bandwidth is updated to “high”.
      2) The End User forces the attachment download.

**Contributor:** Antonio Vitale, Alcatel

3.4.4 UC1.4 - Finding attractions

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
</tbody>
</table>
Primary Actor: Traveller  
Stakeholders: Network operator, Service provider  
Preconditions:  
Postconditions: Traveller has found desired attractions nearby  
Traveller is able to use the same service next time by clicking the service icon  
Trigger: User wants to find attractions nearby  
Minimal guarantees: Traveller gets list of attractions into his phone. Not necessarily near where he is currently located.  
Success guarantees: Spice platform helps traveller to find selected attraction.  

### Main Success Scenario

1. Traveller opens Spice enhanced yellow page application on his phone.  
2. Traveller selects topic from a predefined list of topics (e.g. night clubs)  
3. Traveller selects preference criteria from a predefined list of criteria  
   (e.g. **Distance** from his hotel must be less than 1km from where he is now  
   Club is popular among **30-40 year old** people) For more details see use case requesting attractions.  
4. Spice enhanced yellow page application lists attractions that meet the criteria with hyper links, for more details see use case listing attractions.  
5. Traveller clicks one of these hyper links, which provides more information on the attraction.  
6. Traveller wants to go to one of the attractions and selects option “guide me to this this place”  
7. Spice enhanced yellow page application provides instructions how to find the place while traveller walks to the attraction (nightclub). For more details see use case locate attraction.

### Extensions (ctrl+tab for indenting the actions)

1a. Spice Platform is not capable of locating traveller  
   1a1.Traveller can still make simple yellow page search  
7. Service asks if user wants to create an icon of the service

Contributor: Sampo Sovio, Nokia

---

### UC 1.4.1 - Listing Attractions

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Sub System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Subfunctional</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Network operator</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Network operator, Service provider</td>
</tr>
</tbody>
</table>
Preconditions:
Postconditions: Network operator and service provider has agreement, which provides coordinates of mobile subscribers to third party server, which provides Spice enhanced yellow pages application.

Trigger: User has requested attractions via yellow pages application.

Minimal guarantees: Traveller gets list of attractions into his phone. Not necessarily near where he is currently located.

Success guarantees Traveller gets list of attractions that are in desired range.

Main Success Scenario
1. Third party server sends travellers IMSI to location server that is an enabler provided by network operator.
2. Location server checks if it is entitled to provide coordinates of this IMSI to third party service provider. The policy of providing coordinates to third party may involve legal agreement between local operator and travellers home operator.
3. When third party server gets coordinates of IMSI it will use them to find attractions that are located in the desired range and meet the other criteria as well.

Extensions (ctrl+tab for indenting the actions)
1a. Third party server does not get the requested coordinates.
   1a1. Third party server provides information that meets other criteria made by traveller.

Contributor: Sampo Sovio, Nokia

---

UC 1.4.2 - Locate attractions

| Scope: | Sub System |
| Level: | User goal |
| Primary Actor: | Traveller |
| Stakeholders: | Network operator, Service provider |
| Preconditions: | Traveller has performed attraction search. |

Main Success Scenario
1. Yellow pages application will give visual or audio signal to user whether he is moving towards to location or not.
   1.a Headset that is connected to mobile phone constantly beeps in different way whether distance between traveller and his destination increases or decreases.
   1.b Mobile phone can blink green and yellow light depending whether distance between
traveller and his destination increases or decreases.

**Extensions** (ctrl+tab for indenting the actions)

| Contributor:  | Sampo Sovio, Nokia |

---

### UC 1.4.3 - Requesting attractions

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Sub System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>User goal</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Traveller</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Network operator, Service provider</td>
</tr>
</tbody>
</table>

**Preconditions:**

**Postconditions:** Traveller has launched Spice enhanced yellow pages application

**Trigger:** Traveller has launched Spice enhanced yellow pages application

**Minimal guarantees:**

**Success guarantees:** Traveller is able to request list of attractions.

**Main Success Scenario**

1. From the User Interface of yellow pages application user selects distance as search criteria.
2. Yellow pages application will warn user that making this search may reveal his location and other valuable information to third party server.
3. User will make decision whether he wants to perform the search by clicking yes or no.

**Extensions** (ctrl+tab for indenting the actions)

| Contributor:  | Sampo Sovio, Nokia |

---

### 3.4.5 UC7.1 Content Adaptation According to User Position

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
</tbody>
</table>

**Preconditions:** The user is using a SPICE service to discover a city zone so he’s moving throughout the network
Postconditions:

Trigger: The network manages the movement of the user adapting the contents about the zones visited according to user movements, localizing him and sending him all the information that he can activate at the moment he asked for such service

Minimal guarantees:

Success guarantees

Main Success Scenario

1. The End User, using her mobile equipment, asks for and obtains (from SPICE system), a service providing a tourist explanation about a city zone.
2. The service installs a personalizing icon on the user terminal desktop.
3. The user can configure such system, specifying his interests (monuments, museums, restaurants, bar and so on...), so the service at every time can deliver the related information.
4. The SPICE network deals with mobility of the user, preparing for the end user at every moment the information that he configured.
5. If the SPICE platform detects that switching platform or content provider can be useful for end user, SPICE platform can execute such switch in a seamless way, e. g. if a high speed wireless is detect and the user is connected to a lower speed connection, or if a content provider has overload or crash problem the platform will change of content provider.
6. A mechanism of discovery of content provider is available and can be used by SPICE platform for automatic switches or by users to find services they may need.
7. The user can consult his mobile equipment every time he needs information about a place

Extensions (ctrl+tab for indenting the actions)

4a. User can decide if the information is showed automatically or if he can select manually the appropriate information using a general menu.
   4a1. if the user selects the option of showing automatically the information, the service every time an interesting place is detected, automatically visualizes the information on the user terminal.
   4a2. if the user wants to select manually the information, the service will only show a menu and give the possibility to the user to select the desired information.

5a. Switching decision can be delegated from SPICE platform to user.

Contributor: Telefónica I+D

3.4.6 UC7.4 – Priority of SPICE Platform in dangerous situations
<table>
<thead>
<tr>
<th>Level:</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>SPICE Platform</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>In an emergency situation, the system must be able to communicate with absolute priority with SPICE users</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>The SPICE users are advised on time about a dangerous situation</td>
</tr>
<tr>
<td>Trigger:</td>
<td>Sensors or agents trigger an alarm about a dangerous situation</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td>The SPICE user is advised about a dangerous situation</td>
</tr>
<tr>
<td>Success guarantees</td>
<td>The SPICE user is updated about the situation and about meeting points and hospitals available</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. Sensors or agents detect a dangerous situation (that can be a natural catastrophe or a war situation), and trigger an alarm through the SPICE platform.
2. SPICE users in the dangerous zone are localized and an advice is sent, according to user profile (voice message, text message and so on).
3. If the SPICE user is holding a call at the alarm moment, his communication is interrupted or put in stand by, to be sure that the user has received the alarm.
4. The SPICE users receive the alarm and can use a SPICE service to be updated at every moment about the situation in the area, about available hospitals and meeting points.

**Extensions** (ctrl+tab for indenting the actions)

1a. The alarm must be transmitted to the system, and acknowledged by the last, to be absolutely sure that the alarm process has started.
   1a1. if the system doesn’t acknowledge the alarm, the sensors or agent must retry the transmission until an acknowledgement is received.

2a. A SPICE user that is localised in a concrete danger zone and that at the moment of alarm is out of coverage must be advised about danger. At this purpose, the alarm will be sent until the SPICE user acknowledges the correct reception.

3a. If the user is holding a call, the possibility of resuming such call should be considered.

4a. In such cases, the possibility exists that the platform cannot reach the user properly depending on the damages that he has suffered. In this case cannot go on giving service to users.

Contributor: Telefónica I+D

UC 1.9
3.5. Service Lifecycle

3.5.1 UC 1.7 - Application designer builds a new application

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Application designer</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The building activity is performed on a PC or laptop under Windows XP OS</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>The application designer wants to simply create and deploy a new application linked to an available service.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The application designer selects an existing service by using a repository browsing service based on semantic and ontology requests.
2. The application designer defines the GUI (interaction between the end user and the service) by defining the action to be done for each input, output linked to the service or to the notification sent by the service (entry, display, selection in list ..).
3. The application designer provides a description (semantic and ontology) of the new application and defines the publication rules (profile of users that will be allowed to access this application).

**Extensions** (ctrl+tab for indenting the actions)

| Contributor: | Patrick Fontaine, Alcatel, E. Postmann, SAGO |

3.5.2 UC3.7 - Service provider deploys a new service

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
</tbody>
</table>
### Preconditions:
A new service/application was built (be the application designer) and shall be installed and deployed.

### Postconditions:

### Trigger:
The Service Provider wants to install and deploy a new application linked to an available service.

### Minimal guarantees:

### Success guarantees

### Main Success Scenario

1. The service provider installs the application, i.e. the application/service software is imported into the SPICE execution environment including all parts that are needed for correct execution.
2. The service provider deploys the service/application, i.e. the installed software is initialised and configured to default values and is then ready to run under control of the execution environment.
3. The relevant subscribed end users are notified that a new application/service is available.

### Extensions (ctrl+tab for indenting the actions)

### Contributor:
Patrick Fontaine, Alcatel, E. Postmann SAGO

### 3.5.3 UC 3.11 – Service sends a Multi-Modal Notification to User

<table>
<thead>
<tr>
<th><strong>Scope:</strong></th>
<th>Sub-System</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level:</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Primary Actor:</strong></td>
<td>SPICE’s Attentive Service</td>
</tr>
<tr>
<td><strong>Stakeholders:</strong></td>
<td>End User, service engines</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>End User is logged in.</td>
</tr>
<tr>
<td><strong>Postconditions:</strong></td>
<td>User has received notification</td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>SPICE’s Attentive Service wants to send multi-modal notification</td>
</tr>
<tr>
<td><strong>Minimal guarantees:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Success guarantees:</strong></td>
<td>Notification is sent and received by user</td>
</tr>
</tbody>
</table>

### Main Success Scenario

1. SPICE Attentive Service creates “Multi-Modal Notification” message (e.g. sets the text field, creates an audio object, creates a video object) and defines the associated delivery conditions (i.e. service’s policy).
2. SPICE Attentive Service retrieves from the SPICE platform (i.e. from functional
3. SPICE Attentive Service selects for “text” modality available option.
4. Notification is sent as “text” message to end user’s terminal.
5. “Text” content is displayed on end-user terminal.

<table>
<thead>
<tr>
<th>Extensions (ctrl+tab for indenting the actions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5a. Because of user behavior model, “text” is displayed on a “device” in the surrounding of user (e.g. in the Head-Up display of the car)</td>
</tr>
<tr>
<td>3a. SPICE Attentive Service selects for “audio” modality</td>
</tr>
<tr>
<td>3a1. Notification is send as “audio” object to end-user terminal</td>
</tr>
<tr>
<td>3a2. “Audio” object is played on end-user terminal</td>
</tr>
<tr>
<td>3b. SPICE Attentive Service selects for “audio” modality</td>
</tr>
<tr>
<td>3b1. Notification is sent as “audio” object to end-user terminal</td>
</tr>
<tr>
<td>3a2. Because of user behavior model, audio object is played on a different device in the surrounding (e.g. the stereo player)</td>
</tr>
<tr>
<td>3b. SPICE Attentive Service selects for sending the “Audio &amp; video” notification to terminal</td>
</tr>
<tr>
<td>3b1. Notification is sent as multi-modal object to end-user terminal</td>
</tr>
<tr>
<td>3a2. Because of user behavior model, multi-modal object is played as “audio and video” output on an end-terminal</td>
</tr>
</tbody>
</table>

Contributor: Ernő Kovacs, NEC
Antonio Vitale, ACIT

This use case can be linked to “ET1: Assist the traveler”, Step 6 Inform the traveler

### 3.5.4 UC7.6 – Terminals actualization and reconfiguration

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Content provider</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>SPICE Platform</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>A SPICE content provider can update his services and applications</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>The terminals must be updated in a seamless way for end user</td>
</tr>
<tr>
<td>Trigger:</td>
<td>A SPICE content provider has a new version of an application that users are currently using. The content provider can program an upload of his software according to an established profile.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td>If it wasn’t possible to upload the software version, the old version of the software will work until an update is possible</td>
</tr>
</tbody>
</table>
Success guarantees | The software is automatically downloaded and updated in a seamless way
---|---

**Main Success Scenario**

1. A SPICE user buys a service from a content provider
2. The user can download and install on his mobile terminal the icon of the service, through such icon he can access the service using a graphical environment especially developed by content provider to access such service
3. When the user downloads the software for the first time, authorization for automatic update is proposed to the user, and if he accepts all the updates would be automatic and without user interaction
4. If a new version of a software is available, the content provider, according to his defined policies, provides the update of the software
5. A content provider can change his policies according to his necessity or to follow a planning that can be better for end user and less interfere in his device use
6. The end user follows using his service without any interruption

**Extensions** (ctrl+tab for indenting the actions)

3a. The user can avoid automatic update of software.
   3a1. An authorization message appears to the user every time a new software version is available and the user can accept or decline the update.

| Contributor: | Telefónica I+D |

### 3.5.5 **UC 5.4 – Service Developer creates a new service**

<table>
<thead>
<tr>
<th><strong>Scope:</strong></th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level:</strong></td>
<td>Summary</td>
</tr>
<tr>
<td><strong>Primary Actor:</strong></td>
<td>Service Developer</td>
</tr>
<tr>
<td><strong>Stakeholders:</strong></td>
<td>Communication Service Provider (A Telecom Operator)</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>The SCE is available and configured. In particular, all basic enablers provided by the SEE are &quot;ready&quot; to use. The service developer has good knowledge on the service description language formalism.</td>
</tr>
<tr>
<td><strong>Postconditions:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Trigger:</strong></td>
<td>Developer gets order for a new service.</td>
</tr>
<tr>
<td><strong>Minimal guarantees:</strong></td>
<td>The defined service is consistent: the SCE warns about any incompleteness.</td>
</tr>
<tr>
<td><strong>Success guarantees</strong></td>
<td>The defined service is installed and ready for being executed (possibly in a testbed environment).</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The Service Developer selects from a list of service categories the one that better fits
the new specific service he wants to create.

2. The Service Developer visualises graphically the pre-defined structure corresponding to the kind of service it has selected in step 1.

3. The Service Developer updates the pre-defined structure and starts "drawing" the logical flow of the new service by composing service-components. In addition it defines the data and the asynchronous signals to be manipulated within this service description.

4. Some composed service components (see 3) are taken from the available SEE library, while others are directly defined using the service description language formalism.

5. The Service Developer publishes to the SEE components library the service components developed by him that he perceives as being reusable.

6. The Service Developer checks the consistency of its service description using a command of the SCE.

7. The Service Developer implements in a programming language supported by the SCE (e.g. Java) the parts of the service implementation that cannot be derived from the service definition.

8. The Service Developer fills the semantic information to be attached to this service to allow dynamic service discovery or composition when needed.

9. The Service Developer registers the new service, publishes its definition, deploys it in the test environment.

10. The Service Developer executes the service deployed in the previous step to check its behavior.

11. The Service Developer iterates as needed over all steps (from step 3) until it obtains the intended behaviour

**Extensions** (ctrl+tab for indenting the actions)

6a. Definition of the logic of the "client" part of the service

   In some cases defining the "server" part of the service logic is sufficient. However in some other cases the service developer will need to describe both parts separately: server and client side of the service, to take into account the fact that a service may access resources that are available locally in the client terminal.

4a. Population of the SEE component library with existing external components.

   The service developer may take the initiative of importing into the SEE library an external component that is absent. To this end, it will use an SCE "import" command (UC5.09 Discover Available Services). The imported component could be a WSDL Web-Service. The import command translates the original WSDL description into the SDL language formalism.

6a. Simulation of the described service

   If the SCE simulation module is available, the Service Developer can check the intended service logic by activating the graphical simulator.

| Contributor: | • Mariano Belaunde, France Telecom  
• Ernő Kovacs, NEC |
3.5.6 UC 5.5 – End user creates a new service

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>End User, Service Provider, Network Operator</td>
</tr>
</tbody>
</table>

Preconditions: A set of basics services are installed inside the SPICE service platform and the user can access those services. A Service customization tool is available to the user by the Service Provider.

Postconditions: The service has been created by the end user and is executed in the SPICE platform.

Trigger: The End User needs a service that allows him to send an SMS message to all his friends everytime his blog is updated, no such service is available on the SEE.

Minimal guarantees: The SPICE platform allows the user to customize the behavior of the Service to which he is subscribed to.

Success guarantees: The user has created the new service and can start using the service.

### Main Success Scenario

1. The user is searching a service using the discovery function inside the Service Execution Environment (SEE) of the SPICE platform to send update via SMS of its blog updates to his friends.
2. There is no such a service in the SPICE platform.
3. The User looks for available services inside the spice platform with the keywords “Web Update” and “Send SMS” inside the SPICE platform.
4. The USCE returns a list of services among those services there are a “Web Feed Update Service” provided by several service provider and network operators and a “Send Message” service provided by several network operator.
5. The user composes those services to create a service that allows him to send SMS to all the phone numbers or chat contacts that are inside his mobile phone address book under the group “blog friends”.
6. After having composed the service the user can select the providers that are more convenient for the overall composed services (this can depend for instance on the mobile network provider used by his friends).
7. The user may submit his service to a service provider, to be published in the SPICE environment, by using a wizard that allows him to add the description of the service.

### Extensions (ctrl+tab for indenting the actions)

3a. The user cannot find basic service that can be used to compose the service, for instance there is no service that reports web site updates.

   The Spice platform provides a most wanted service list that can be implemented by the operator or other business entity.

b. The User customizes the Send SMS Service to manually send updates to his friends.
5a. The user composes the service in the wrong way.

The SPICE platform checks and validates the composed service before the publication.

Contributor: Francesco Moggia, Telecom Italia

3.6. Service Execution Model

3.6.1 UC5.1 Invoke a session communication service

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The End User is just authenticated and attached to the wireless access network.</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>The End User invokes the services.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td>The service was invoked.</td>
</tr>
<tr>
<td>Success guarantees</td>
<td>The service was successfully executed.</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

10. The End User, using a mobile equipment, invokes a service on the terminal.
11. The service on the terminal sends out the service request for session establishment to the network.
12. The network forwards the request to the service in the SEE.
13. The SEE executes the service logic. Accessing user specific data maybe part of executing the service logic.
14. The SEE sends the response to the network.
15. The network sends the request for session establishment to the service on the end terminal.
16. The service on the end terminal offers the service request to the destination user.
17. The destination user accepts the service request and returns a confirmation to the service on the terminal.
18. The service on the terminal returns a confirmation.

The media path is established.

**Extensions** (ctrl+tab for indenting the actions)

9. This case is only showing one general use case. Alternatives in which the confirmation is going through the service platforms or that the service receives more requests are possible.
### 3.6.2 UC5.2 Service using the IT/Web Service execution model (Click-to-Dial)

<table>
<thead>
<tr>
<th>Scope</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Service Provider, Network Operator, End User</td>
</tr>
<tr>
<td>Preconditions</td>
<td>A Web based Click-to-Dial service is deployed on the Spice Platform. The End User is a subscriber of the service and authenticated on the Platform. The called users are subscribers of the Service.</td>
</tr>
<tr>
<td>Postconditions</td>
<td>The call has been established between the two (or more) users</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user wants to establish and monitor a call using a web interface.</td>
</tr>
<tr>
<td>Minimal guarantees</td>
<td>The service has established the call or it reports errors related to unreachable users</td>
</tr>
<tr>
<td>Success guarantees</td>
<td>The call has been established between the two (or more) users</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. On the SPICE Service Execution Environment is available a service that can be accessed by the Service users through a Web Interface. The User can input two users id (either sip addresses or phone id). The logic of the service is:
   1. The service is activated by the end user request on a Web page containing two users id
   2. The service, based on the type of id/address in input, establishes a call connection between the users
   3. The service receives incoming notifications of the call status
   4. The service provides the invoker user the state of the call through a web report
2. The SPICE Service Execution Environment receives an HTTP request with addresses of the called users as parameters
3. The service is activated
4. A “Click-To-Dial” component is activated and processes the HTTP request
5. A “Call Control” component is activated
6. The Click-to-Dial component invokes the call establishment on the Call Control component
7. The Call Control component tries to establish the call between the two parties and returns to the Click-to-Dial component
8. The state/progress of the call is provided by the Call Control component to the Click-to-Dial component
9. The state/progress of the call is provided to the End User by the Click-to-Dial component by means of successive HTTP requests by the End User with a call-handle (call-id) parameter to allow the monitoring of the call

Extensions (ctrl+tab for indenting the actions)

2a. The service supports multiple users call (conferencing). The users addresses are parameters of the HTTP request

9a. The End User can stop the call. The request "stop call" is issued as HTTP request

Contributor: Andrea Parodi, IRIS

3.6.3 **UC5.3 Service using the Messaging execution model**

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Service Provider, Network Operator, End User</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>A Message Based information retrieval service, deployed and available on the Spice Platform, is triggered by the end user</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>The Service has been executed</td>
</tr>
<tr>
<td>Trigger:</td>
<td></td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td>The Service has been triggered and executed.</td>
</tr>
<tr>
<td>Success guarantees:</td>
<td>The requested information has been found and provided to the user</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. On the SPICE Service Execution Environment is available a service that can be activated by an incoming message (may be a SMS or a SIP IM). The incoming message has a specified syntax. The logic of the service is:
   1. The service is activated by an incoming message with a given syntax (eg find <who>)
   2. The service, based on the infos contained in the message, performs a search on a directory
   3. The service returns a message to the sender of the first message, containing the requested information (found contact and phone number)
   4. The service receives another message requesting to be connected to the found contact
   5. The service establishes a third party call connecting the message sender with the found entry
   6. At the end of the call the service sends a message with service usage summary information (eg call duration)
2. The SPICE Service Execution Environment, receives an incoming SMS
3. The service is activated
4. A “Receive SMS” component is activated  
5. A “Directory Search” component is activated requesting to search a contact defined in the SMS body  
6. The “Directory Search” component returns the contact information  
7. The SPICE Service Execution Environment, by means of a “Send SMS” component sends the retrieved info to the service requestor  
8. The service requestor sends another SMS requesting to be connected to the contact  
9. The SPICE Service Execution Environment, receives an incoming SMS with a given syntax (eg. call)  
10. A “Receive SMS” component is activated  
11. A “Third Party Call” component is activated  
12. The call is started  
13. When the call ends, this event is notified to the SPICE Service Execution Environment  
14. The SPICE Service Execution Environment, by means of a “Send SMS” component sends a final service usage summary to the service requestor

<table>
<thead>
<tr>
<th>Extensions</th>
<th>(ctrl+tab for indenting the actions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. The service supports multi channel user interactions (the same service may be activated by IM, SMS, MMS, EMAIL)</td>
<td></td>
</tr>
<tr>
<td>8a. If the service requestor doesn’t request the third party call, after a configurable amount of time the service is deactivated</td>
<td></td>
</tr>
</tbody>
</table>

Contributor: Alberto Baravaglio, Telecom Italia

### 3.6.4 UC5.14 Push Service execution model

<table>
<thead>
<tr>
<th>Scope:</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Service Component</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Service Provider, Content Provider, End User</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>End user has subscribed to the push service.</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>Push content has been delivered to the end-user terminal.</td>
</tr>
<tr>
<td>Trigger:</td>
<td>An external event was discovered that triggers the Push service.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td>The service executes the service push.</td>
</tr>
<tr>
<td>Success guarantees:</td>
<td>The service executes and the push content is delivered.</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The service component checks the triggering condition and finds a matching condition OR an external component triggers the service component.  
2. The service component builds the push content object.
3. The service component identifies the target user to deliver the push content to.
4. For each end user, the service component identifies the currently used terminal of the user and the delivery means to use.
5. The service component uses the adaptation engine of the SPICE platform to adapt the push content for the target terminal and the target delivery means.
6. The adapted push content is delivered to the terminal.

<table>
<thead>
<tr>
<th>Extensions (ctrl+tab for indenting the actions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6a. The push content delivered to the end-terminal contains a link to the complete push content and the user activates that link</td>
</tr>
<tr>
<td>6a1. The complete push content is accesses and displayed to the user</td>
</tr>
</tbody>
</table>

Contributor: Ernő Kovacs, NEC

3.7. Service Discovery

3.7.1 UC1.5 - Assist the traveler

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Service Provider</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Traveller (in a foreign country)</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The traveller has created a travel plan with the travel ledger</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>The traveller is visiting a foreign country</td>
</tr>
<tr>
<td>Success guarantees:</td>
<td>The traveller is routed correctly during the trip</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The traveller activates the guidance service of SPICE
2. The SP locates the traveller
3. The SP informs the traveller about the current status of the trip and shows the next planned steps
4. The traveller approves the next planned steps
5. The SP retrieves updated information about the next steps (timetables, reservations, traffic, current hazards)
6. The SP directs the traveller with notifications to his SPICE enabled device
7. The SP constantly monitors the location of the traveller and external environment (weather, traffic, delays)
8. The traveller reaches his destination

ID: SPICE_WP1_D1_1_v1.4_clean.doc
Revision: 1.4
Date: 02/05/06
Security: PUBLIC
Page 49/74
9. The SP provides a summary of the trip to the traveller

**Extensions** (ctrl+tab for indenting the actions)

*a. An unexpected situation happens (e.g. weather is changing)*  
   *a1. The SP re-routes the traveller based on his current location, preferences and environment*  
   *a2. The SP alerts the traveller and proposes a new route*  
   *a3. Continue from 3*

*b. Location information is not available*  
   *b1. SP informs the traveller that the automatic assistance is not available*

4a. The traveller rejects the proposed plan  
   4a1. The SP proposes an alternative route  
   4a2. Continue from 3

7a. The traveller's route diverges from the planned route  
   7a1. The SP evaluates the possible reason for the divergence in an intelligent way (like traffic jams)  
   7a2. The SP prompts additional information to the traveller  
   7a3. The SP proposes a new route to the traveller  
   7a4. Continue from 3

7b. An unexpected situation arises (like bad weather ahead, delays)  
    7b1. The SP analyse the situation based on the traveller's location, preferences and context  
    7b2. The SP warns the traveller and proposes a new route  
    7b3. Continue from 3

Contributor: Nokia

3.7.2 *SPICE discovers a 3rd party service with semantic discovery (to write)*

3.8. **Payment**

3.8.1 *UC 1.6 - Purchase a movie on-line*

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Customer</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>On-line retailer</td>
</tr>
<tr>
<td>Preconditions:</td>
<td></td>
</tr>
</tbody>
</table>
Postconditions:

<table>
<thead>
<tr>
<th>Trigger:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal guarantees: All the transactions are logged. Customer is not billed if the transaction fails. The on-line retailer does not ship the movie if the payment is not received.</td>
</tr>
<tr>
<td>Success guarantees: the user has purchase the movie. The retailer received the payment.</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. The Customer browses the on-line portal of the movie retailer
2. The Customer selects a final movie to purchase
3. The on-line portal asks the SPICE system to handle the on-line purchase
4. The Customer confirms the purchase
5. The on-line portal credits the purchase amount to the Customer
6. The Customer receives the digital movie on his media centre at home
7. The SPICE system sends the bill information to the Customer

**Extensions** (ctrl+tab for indenting the actions)

*a. At any point the SPICE system goes down
   *a1. The transaction is aborted
   *a2. The SPICE system informs the retailer
   *a3. The log is sent to the service unit of the SPICE system
3a. The purchase request fails
   3a1. The on-line retailer informs the Customer
   3a2. The Customer tries again and continue from 3
6a. The Customer does not receive the movie
   6a1. The Customer asks the SPICE system to block the payment
   6a2. The SPICE system informs the on-line retailer of the Customer's request
   6a3. Solve the purchase dispute

Contributor: Claudio Riva, Nokia

3.8.2 **UC7.3 – Purchase Content for a Third Party**

| **Scope:** | Enterprise |
| **Level:** | Summary |
| **Primary Actor:** | End User |
| **Stakeholders:** | Communication Service Provider |
| **Preconditions:** | A SPICE user wants to buy contents for a third party SPICE user |
| **Postconditions:** | The third party SPICE user can access and use, according to his profile, the content paid for another SPICE user |
| **Trigger:** | The third party receives a notification about the existence of a SPICE user that has paid content for him |
### Minimal guarantees:

<table>
<thead>
<tr>
<th>Success guarantees</th>
</tr>
</thead>
</table>

### Main Success Scenario

1. A SPICE user wants to buy content for a third party SPICE user.
2. The user orders content across the SPICE Platform, specifying that the content is for a third party SPICE user and giving authorization to such user to access the content after his authentication.
3. The SPICE Platform searches for a content provider, pays content and sends a notification to the third party SPICE user advising him that there is available some content for him.
4. The third party SPICE user can accept the content.
5. The content starts to flow to third party SPICE user, which can decide whether to download it or to stream it.

### Extensions (ctrl+tab for indenting the actions)

3a The SPICE Platform must analyze the third party profile and decide if, according the profile, the content that have been bought for such user can be shown to him or not.
3a1. in case the content is not compatible with third party SPICE user profile, a notification will be sent to the user that tried to do the transaction about the impossibility of concluding with exiting the operation.

| Contributor: | Telefónica I+D |

---

### 3.9. Authentication

#### 3.9.1 UCl.8 - Single sign-on

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>Spice operator, service provider, user</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Spice operator, service provider, user</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>User has Spice account; user has an account at the service provider; user has linked his account at the service provider to his account at the spice portal.</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td></td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

### Main Success Scenario

1. User navigates to the Spice portal
2. User is authenticated by the Spice-operator.
3. Spice navigates to a 3rd party service provider’s site
4. Service provider sends an authentication assertion request to the spice portal
5. Spice portal sends an authentication assertion to the service provider,
6. User is granted access to his account at the service provider.

<table>
<thead>
<tr>
<th>Extentions</th>
<th>(ctrl+tab for indenting the actions)</th>
</tr>
</thead>
</table>

Contributor: Nokia

3.10. Group and social services

3.10.1 UC3.8 – group formation and management

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End user</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>The group formation and management activity is performed on a personal device while on the move. The end user is interested in discovering new and matching opportunities during both her work and leisure time</td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td></td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. During a sports event (e.g. sailing cup), the user, a professional journalist, browses on her mobile device some statistics about the participants on the qualification sessions.
2. Prior to the event, the user had set up her SpicyGroup application to disclose some information about her: profession, hobbies and interests, duration of her staying in the sports site, favourite team, etc.
3. As additional information publicly available in her profile, she included her favourite music artist, who is expected to perform in the sports event site.
4. It’s her birthday, and thanks to the group profiling application, she gets aware that eight other people in her proximity, around her age, share both her favourite team and the music artist
5. As it would be too bad to have the dinner of her a birthday all alone, she uses the application to form a group for having dinner together with some of these persons sharing her interests.
3.10.2 UC3.9 - End user uses a group emergency service

<table>
<thead>
<tr>
<th>Scope:</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level:</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor:</td>
<td>End User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>Communication Service Provider</td>
</tr>
<tr>
<td>Preconditions:</td>
<td></td>
</tr>
<tr>
<td>Postconditions:</td>
<td></td>
</tr>
<tr>
<td>Trigger:</td>
<td>The End User exploits an emergency service that allows him to call the nearest available person that can help him.</td>
</tr>
<tr>
<td>Minimal guarantees:</td>
<td></td>
</tr>
<tr>
<td>Success guarantees:</td>
<td></td>
</tr>
</tbody>
</table>

Main Success Scenario

1. The End User, using his PC connected via ADSL, downloads (from the SPICE system), installs and runs a new client application that allows a person group to manage individual crisis situation.
2. The End User configures the emergency service (for example, the emergency list members and the order/priority/criteria of the person to be called).
3. The SPICE Service Platform creates a personalised icon (“red alarm button”) on the user’s dynamic desktop.
4. The End User pushes the red alarm button icon, to indicate a crisis requiring rescue.
5. The SPICE platform selects the current available nearest “rescue person” (for example the End User’s doctor, using an advanced PDA mobile terminal), according to the End User’s rules.
6. The SPICE system opens a voice call between the End User and his doctor.
7. The SPICE system updates the doctor’s dynamic desktop and a map is proposed to reach the End User.
8. The SPICE system sends an instant messaging notification to other contacts present in the emergency list, as a notification of an emergency call

Extensions (ctrl+tab for indenting the actions)

5a. The End User is geographically nomadic, for example, moving with an advanced PDA terminal.

6a1. The SPICE system updates the list of nearest rescue persons.
5a. The End User is temporarily in a different town or country.
   2a1. The SPICE system updates the emergency service considering the new relevant
       and nearest hospital (or regional call centre).
   2a2. The SPICE system uses additional user's preferences (language, specialist
       indications, ... ) to route the emergency call.

6a. The optimal rescue person doesn’t answer.
   6a1. The SPICE system tries to call the following person in the list.

6a. Any rescue person answer.
   6a1. The SPICE system tries to call the nearest hospital (or regional public
       emergency call centre).

Contributor: Antonio Vitale, Alcatel.

Special dedicated statement:
About this proposed scenario, this is a special statement agreement from France Telecom
(Christophe Cordier):

“We would like to notify that France Telecom has a pre-existing know-how about this
scenario : France Telecom filed a patent application, under n° FR 05 786, related to an alert
system which might be needed for carrying out the Project and/or for Use of Knowledge from
the Project.

In accordance with Article IV.4.3 of the IPCA Part II signed by the participants, Access
Rights to such patent application filed under n° FR 05 786 (related to an alert system ), if
needed for the execution of the SPICE project, shall be deemed granted on royalty-free basis
to the concerned participants, as of the date of the EU Contract entering into force, solely for
the purpose of the execution of the Project.”

3.10.3 UC7.7 – Use of Digital Content

<table>
<thead>
<tr>
<th>Scope</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Summary</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>User</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>SPICE Platform</td>
</tr>
</tbody>
</table>
| Preconditions: | A SPICE user can buy digital content and use it according to digital
    rights associated to it |
| Postconditions: | A SPICE user can play back, share and use the digital content
    according to digital rights |
| Trigger: | A SPICE user buys content and wants to store it to play back it later,
    or share it with his family |
<p>| Minimal guarantees: | The user can access to content according to related digital rules |</p>
<table>
<thead>
<tr>
<th>Success guarantees</th>
<th>The user can share, play back content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Success Scenario</strong></td>
<td></td>
</tr>
<tr>
<td>1. A SPICE user buys content protected with digital rights</td>
<td></td>
</tr>
<tr>
<td>2. The user accesses content through his mobile device and then stores it because he wants to reproduce it later using a different device and wants to share it with his family</td>
<td></td>
</tr>
<tr>
<td>3. The user, back home, reproduces content on his TV</td>
<td></td>
</tr>
<tr>
<td>4. The user shares with his family and forwards digital right to another component of his family profile</td>
<td></td>
</tr>
<tr>
<td>5. The other component of the family can access the content and play it back</td>
<td></td>
</tr>
</tbody>
</table>

| **Extensions** (ctrl+tab for indenting the actions) |
| 3a. The user buys a protected digital content and content can be managed according to his digital rights rules |
| 3a1. content only can be streamed once |
| 3a2. content can be stored and played back |
| 3a3. content can be shared by the user with another SPICE user |
| 3a4. content can be shared by the user with several SPICE users |
| 3a5. content can only be used on devices with determined characteristics |

Contributor: Telefónica I+D
4. User Requirements

4.1. Profiles and Policies

The SPICE Platform should allow defining user policies and profiles to guarantee the privacy rules, the possibility to make adaptations decision. In that sense, the SPICE Platform shall comply with the following:

- Be able to define rules and preferences about (at least) access network selection, communication handling and multi-modality priorities.
- Allow the user to define privacy rules and conditional access
- Allow the user to control her willingness to receive “pushed” services and/or information
- Provide a learning mechanism which learns over the history of the user’s behavior
- Define privacy rules that applies to some people that could be different ones in an emergency case
- Allow the user to inform the SPICE platform about their policies and authorize the SPICE platform to consider them in making adaptation decisions

For detailed requirements on work package level see Appendix B (separate document).

4.2. Identification and Authentication

It should be possible to identify and authenticate the users and those terminals the users have. Therefore, regarding Identification and Authentication the SPICE Platform shall comply with the following:

- Identify and authenticate the user
- Allow the service provider to know the bank account of the user
- Allow an agreement between the user’s bank and SP
- Allow the service provider to own the user credential for the bank account
- Protect the content by one DRM scheme
- Implement a rollback mechanism with the paying unit
- Implement a secure transaction system
- Use a federation of identities
- Allow the service provider to have Identity Provider functionality
- Allow the terminals, discovered by the SPICE platform, to be identified and authenticated as terminals belonging to this user identity or group identity
- Allow identifying and authenticating users without their explicit consent
- Identify and authenticate service creators
For detailed requirements on work package level see Appendix B (separate document).

4.3. Context Awareness

The SPICE platform should allow defining privacy rules to access content and allow the user to find the correct content. Therefore, it shall comply with the following:

- Store in service provider’s backend information, that may be private, for future reference
- Define privacy rules different in an emergency case
- Define several variations in privacy rules in an emergency case
- Allow that the user find the right content, having a vague knowledge of the content he/she wants
- Allow that the end user stores and plays back digitally licensed digital content on her mobile device

For detailed requirements on work package level see Appendix B (separate document).

4.4. Service Handover

Regarding Service Handover the SPICE Platform shall allow the user moving through heterogeneous access networks, using different terminals. In that sense, the SP should comply with the following:

- Authenticate and track automatically the user that is moving through heterogeneous access networks
- Allow the user to be independent from her current terminal device in case she wants to use a different one
- Be able to access the same services with different devices and in different locations
- Be able to transfer rights from one terminal to the other and delete them from previous one.
- Negotiate the SLA even during a session, to enable the roaming while consuming a content or service

For detailed requirements on work package level see Appendix B (separate document).

4.5. Dynamic Desktop

Regarding Dynamic Desktop the SPICE Platform shall comply with the following:

- Provide at least the following input modalities: Speech, Text, Camera, Pointing (on sensible display), Gesture (movement).
• Allow that the user is aware that the device is ready and attending to her actions.
• Allow the user to be able to effect meaningful actions, control its extent and possibly specify a target or targets for her action.
• Be able to support at least the following sensor information input sources: GPS information, Cell ID, Proximity, other networks & tle infrastructures such as WiFi hotspots/Bluetooth detectors, Humidity, Altitude, Air pressure, Temperature, Sound detection, 3-Axis Speed, 3/Axis acceleration, Heart rate, ECG, Barometric Pressure
• Provide at least the following output modalities: Audible feedback (e.g. pre-recorded prompts and/or Text-to-Speech synthesis), Graphics representation (raster and vector-based images, maps), Sounds
• Be able to update the available services presented dynamically, to adapt to the user's context change
• Allow that SP provides suggestions and let the user be in full control.
• Allow that dynamic desktop shows the most valuable information, avoiding to hide high priority information (e.g.: urgent warnings) with low level, informative ones
• Provide a tool for the browsing of repositories.
• Be able to communicate relevant changes to the user.
• Allow buying digital content access rights for another user
• Provide the multilingual support for the services

For detailed requirements on work package level see Appendix B (separate document).

### 4.6. Terminal Usage

Regarding Terminal Usage the SPICE Platform shall accept that the user use different terminal, complying with the following:

• Be able to accept usage by a principal that is not identified as belonging to the owner group.
• Be able to have a notion of a terminal that may be rented/owned for a short time by another user.
• Be able to authorise or forbid some groups to use a terminal.

For detailed requirements on work package level see Appendix B (separate document).

### 4.7. Software update

Regarding Software Update the SPICE Platform Infrastructure shall comply with the following:

• Be able to upgrade software of a SPICE component by secure downloading.

For detailed requirements on work package level see Appendix B (separate document).
4.8. Service creation and usage

The SP should allow informing the user about service creation, the way to access the service and usage of it. So, the platform shall comply with the following:

- The user is informed of the service available on the Execution Environment and the advertised information contains every kind of information useful to access the service.
- Multimodal terminal adaptative content consists of several sections like a small or long text, audio/video objects or a multimodal alternative. Also, it contains meta-data for describing in which situation to use which alternative.
- Graphical notation for service composition shall be adapted to the knowledge/experience of the service designer.
- Service designer shall be able to check using the SCE whether a composition of service components meets the service requirements.
- At the end of service composition process, the service shall be described semantically in terms of service function, input and output, etc.
5. Enterprise Requirements

Enterprises are facing increasing needs for mobility, simplified collaboration and seamless access to the enterprise internal information database from inside the enterprise domain, outside the enterprise domain even across networks/domain boundaries and using various types of devices. Beside this also the enterprise business processes become increasingly mobile as globalization is proceeding.

The following section provides a summary of requirements addressing enterprise domains. These requirements will ensure the smooth use of services in a specific enterprise domain as well as the inter-working of distributed service components, across various networking and enterprise domains integrating 3rd party service providers. Furthermore it will enhance enterprises’ use of ICT services and information systems as well as it will provide all types of advanced professional services the enterprise user will need in the future.

These requirements will help to derive concepts to provide new business opportunities and significant improvements to user centric services for an enterprise user. Enterprise requirements include interoperability, mobility and platform aspects, as well as content delivery and charging aspects. These requirements will help to pave the way for the architectural work done in the SPICE project.

The detailed list of enterprise requirements is located in the Appendix C (separate document).

5.1. Interoperability

**Enterprise Interoperability** deals with the communication handling between different SPICE platform infrastructures that might reside in different operator/enterprise domains. The different components of an enterprise service might run on different types of platforms that are located within the same or within different administrative domains. Consequently services have to be portable over different domains; therefore well defined interfaces are needed in order to guarantee e.g. inter-working, reliability, availability, QoS support, etc.

5.2. Mobility and Roaming

**Mobility** can be identified as the capability to move from one area to another. Different types of mobility can be distinguished: Roaming (moving from one location to another without an ongoing session), Handover (moving from one location to another with an ongoing session).

**Roaming** is a general term in wireless telecommunications that refers to the extending of service access/usage (e.g. connectivity service) in a location that is different from the home location a user was registered in (which is different from seamless mobility/handover). Concerning the SPICE project several topics have to be clarified, as there are:
- moving from one SPICE platform to another within one enterprise domain
- differentiate between services usable at different platforms and services usable only at the home platform
- agreements between enterprises to achieve roaming between enterprise domains
- differentiate between services usable at "foreign" enterprise domain and services usable only at the "home" enterprise domain
**Handover** also called seamless mobility, the ability of a user terminal to dynamically switch between access networks while remaining within the same location. Different types of mobility can be taken into account (e.g. domain handover or technology handover in conjunction with network prioritization by the user). One benefit of the SPICE platform will be to ensure staying connected while using a service. Therefore parameters like costs, QoS, security, legal agreements, etc. have to be considered.

### 5.3. Billing and Charging

One feature of the SPICE platform is an integrated billing and cost management. It should be realized by an easy-to-use mechanism; it should be understandable and transparent to the user (information before, during and at the end of the service). Furthermore the user should be able to monitor and control the actual cost of the service usage. This leads to strong trust and identity assurance mechanisms. Considering roaming, enterprises need agreements between them to facilitate charging and billing.

### 5.4. Multi-Modality and Multi-Terminal

Enterprises have to make use of the DCS to serve their users depending on their actual context. Services shall be accessible in a controlled manner from the users even if they are roaming in "foreign" networks/domains. The user shall be able to use different types of access technologies, different types of terminals to access and use the subscribed services. The user shall be served in an optimized way depending on the recent user/network context, e.g. by selecting the most appropriate technology (SMS; MMS; email, Voice message, Voice call; etc.) for informing the user in case of emergency situations.

This will help the enterprises to track their users and help them to achieve a better service (e.g. distribution transparency of system components in the dimensions of location, replication, failures and mobility management).

### 5.5. Service and Service Platform

As a part of service management and service life cycle control, the retrieval/exchange of knowledge must be possible. Exchange of information between different service platforms should be organized in such a way that the exchange is possible in an easy manner. This might be necessary during service discovery, service composition, brokering and mediation, as well as context retrieval and management. Additional contextual information that can be relevant to the end-user’s needs is gathered such that the output is even more customized to the end-user. In case of different service providers, mechanisms should be foreseen to restrict the execution of (parts of) the service to another service provider or to allow the execution of (parts of) the service only to a number of selected parties. The SPICE platform should also provide an interface for external service platforms that wish to interact with context sources.

### 5.6. Content delivery

Enterprises must support content delivery from various actors to their subscribers in a controlled manner. Future business cases will include content delivery not only from Content Providers to users but also include the delivery of personalized and adapted user contents between users. This, together with the delegation aspects, opens up new security challenges and the enterprises should be in a position to address these challenges in order to mitigate the...
illegitimate access to the content by unauthorized users. Therefore the content owners shall be able to create intellectual property licensable content, to specify access rights to that content and to choose a right owner that stores specific rights or rules of the license and implements the licensing. Digital content and licenses should be distributed and stored separately; making it easier to manage the system.

5.7. Policy
The SPICE platform must enable operators, service providers as well as users to indicate their policies, for instance regarding the reconfiguration of devices. Users and service providers must be able to dynamically configure the SPICE platform with new policies to increase the platform’s flexibility. Dynamically in the sense that users might change their policies regarding preferred modalities, whereas service providers may change their policies regarding the distribution of software updates. To accomplish this, the policies of users and service providers should be expressed in one (or more) policy specification language(s). The SPICE platform should be able to interpret these languages, some of which may be SPICE-specific.
6. Technical Requirements

It is foreseen that the technical architecture of the SPICE Platform will be divided into three main parts:

1. System / Basic Enablers
2. Intelligent and Content Enablers
3. Service Creation

Service Execution is also seen as one of the main parts, but this one is covered by the first two parts.

All three parts will have to deal with the same categories of technical requirements. These categories are:

1. Interoperability
2. Deployment, Acquisition and Discovery
3. AAA
4. Operation & Maintenance
5. Runtime Execution
6. Non-functional

6.1. System / Basic Enablers

6.1.1 R8.1.1 - System / Basic Enablers Interoperability

Interoperability has to deal with the communication between different SPICE Platform infrastructures that might reside in different operator domains. Also mobility aspects concerning device connectivity and service execution have to be addressed. In order to deal with all these kinds of interoperability SPICE components have to be defined in a well-defined way.

Regarding interoperability the SPICE Platform Infrastructure shall comply with the following:

- Build up from distributed components.
- Be able to communicate and interface to other SPICE Platform(s) outside the current administrative domain.
- Be able to deal with seamless handovers for both access and service execution.
- Be able to influence Service execution through context aware parameters and/or user preferences and profiles.
- Use a well-defined description language for defining SPICE components.

For detailed requirements on work package level see Appendix D (separate document).

6.1.2 R8.1.2 – System / Basic Enablers Deployment, Acquisition and Discovery

SPICE components being created in the Service Creation Environment have to become available to all kinds of services for execution without knowing where they actually reside.
order to do this, well-defined deployment, acquisition and discovery mechanisms have to be in place.

Regarding Deployment, Acquisition and Discovery the SPICE Platform Infrastructure shall comply with the following:

- Be able to deploy components that comply with the SPICE component definition.
- Be able to access components that comply with the SPICE component definition.
- Be able to discover components that comply with the SPICE component definition.
- Be able to influence deployment, acquisition and discovery based on context aware parameters and user preferences and profiles

For detailed requirements on work package level see Appendix D (separate document).

6.1.3 R8.1.3 – System / Basic Enablers AAA

End-users require a certain amount of privacy and security where service providers require the ability to charge for service execution and to be able to control access regarding end-users to service execution.

Regarding AAA the SPICE Platform Infrastructure shall comply with the following:

- Provide support for rating.
- Provide support for different charging models.
- Provide support for component access control.
- Provide support for settlement.
- Provide support for component licensing.
- Provide support for data protection and privacy.
- Provide support for accounting

For detailed requirements on work package level see Appendix D (separate document).

6.1.4 R8.1.4 – System / Basic Enablers Operation & Maintenance

Service Providers want the ability to introduce new versions of component without disturbing the runtime environment regarding already existing services. For this reason it is required to have several versions of the same component in the runtime environment available.

Regarding Operation & Maintenance the SPICE Platform Infrastructure shall comply with the following:

- Provide support for component versioning and being able to deal with different versions of the same component for backward compatibility.
- Provide support for component configuration.

For detailed requirements on work package level see Appendix D (separate document).
6.1.5  R8.1.5 – System / Basic Enablers Runtime Execution

Service Providers need the ability to monitor the service execution in order to anticipate on the required hardware – and software setup for providing a good quality of service towards their customers. For good quality of service also the system recovery mechanism is important. For a good quality of service it might be necessary to provide support for different communication models in different situations.

SPICE components shall have the ability to communicate with each other by means of standard communication mechanisms.

Regarding Runtime Execution the SPICE Platform Infrastructure shall comply with the following:
  • Provide support for component monitoring and tracing.
  • Provide support for component persistence in order to provide system recovery
  • Provide support for multiple communication models.
  • Provide support for a component communication mechanism.

For detailed requirements on work package level see Appendix D (separate document).

6.1.6  R8.1.6 – System / Basic Enablers Non-Functional

Service Providers require the ability to exactly dimension the system towards their needs. It must therefore be easy to increase performance when the number of end-users increases.

In order to properly debug services it must be possible to run them both in operation mode and test mode. For this components also need separate test and runtime interfaces.

Regarding Non-Functional requirements the SPICE Platform Infrastructure shall comply with the following:
  • Distinguish between ontology’s of service and domain.
  • Be able to increase performance by means of scalability.
  • Define transactional semantics and different levels of persistence.
  • Support for separate test and runtime component interfaces
  • Support for separate test and production modes of operation

For detailed requirements on work package level see Appendix D (separate document).

6.2.  Intelligent and Content Enablers

6.2.1  R8.2.1 - Intelligent and Content Enablers Interoperability

The SPICE platform infrastructure has to be able to link context awareness in the service execution in order to improve end-user experience and the quality of service outcome.

Regarding interoperability for intelligent and content enablers the SPICE Platform Infrastructure shall comply with the following:
• Provide support for context management.
• Provide support for context source aggregation.
• Provide learning mechanisms (also based on context information)
• Provide support for session mobility
• Provide support for semantic technology.

For detailed requirements on work package level see Appendix D (separate document).

6.2.2 R8.2.2 - Intelligent and Content Enablers Deployment / Acquisition / Discovery

Regarding deployment, acquisition and discovery for intelligent and content enablers the SPICE Platform Infrastructure shall comply with the following:

• Provide support for discovery of context information.
• Provide support for retrieval of context information.
• Provide support for exchange of context information.
• Provide support for aggregation of context information

For detailed requirements on work package level see Appendix D (separate document).

6.2.3 R8.2.3 - Intelligent and Content Enablers AAA

The SPICE platform infrastructure has to take care of privacy and security related to Context and Content Information of an end-user with respect to service execution.

Regarding AAA for intelligent and content enablers the SPICE Platform Infrastructure shall comply with the following:

• Provide End-user control of knowledge inference.
• Provide content management right verification.
• Provide encryption mechanisms.
• Provide content access control.

For detailed requirements on work package level see Appendix D (separate document).

6.2.4 R8.2.4 – Intelligent and Content Enablers, Operation & Maintenance

The SPICE platform infrastructure has to take care of end-user management in relation to service accessibility, configuration and execution.

Regarding Operation & Maintenance for intelligent and content enablers the SPICE Platform Infrastructure shall comply with the following:

• Provide support for User Profile and Policy Management.
• Provide support for Profile versioning.

For detailed requirements on work package level see Appendix D (separate document).
6.2.5 **R8.2.5 – Intelligent and Content Enablers Runtime Execution**

The SPICE platform infrastructure shall be able to detect context changes on an end-user level and use this context information in the actual service execution.

Regarding runtime execution for intelligent and content enablers the SPICE Platform Infrastructure shall comply with the following:

- Provide support for distributed storage.
- Provide support for concurrent data access.
- Provide support for context reasoning.
- Provide support for detecting context changes.
- Provide support for prediction based on context information.
- Provide support for adapting the service execution to the context information.

For detailed requirements on work package level see Appendix D (separate document).

6.2.6 **R8.2.6 – Intelligent and Content Enablers Non-Functional**

Service Providers require the ability to exactly dimension the system towards their needs.

Regarding non-functional requirements for intelligent and content enablers the SPICE Platform Infrastructure shall comply with the following:

- Provide a scalable solution for content distribution.

For detailed requirements on work package level see Appendix D (separate document).

6.3. **Service Creation**

6.3.1 **R8.3.1 – Service Creation Interoperability**

See section *Erreur ! Source du renvoi introuvable.*

6.3.2 **R8.3.2 – Service Creation Deployment / Acquisition / Discovery**

The Service Creation Environment of SPICE platform infrastructure shall be able to deploy and discover new components and/or services. The general idea is also that from existing components/services new components/services can be created and deployed.

Regarding deployment, acquisition and discovery for service creation the SPICE Platform Infrastructure shall comply with the following:

- Provide a list of available component/services to the service creation environment
- Provide support for deployment of newly created components/services (out of already existing components/services).

For detailed requirements on work package level see Appendix D (separate document).
6.3.3 **R8.3.3 – Service Creation AAA**

The Service Creation Environment of SPICE platform infrastructure shall provide access control to particular components/services. For the creation of components/services it shall be able to make security features available to the Service Developer.

Regarding deployment, acquisition and discovery for service creation the SPICE Platform Infrastructure shall comply with the following:

- Provide a list of only accessible component/services to the service developer
- Provide support for access control of the service developer to the service creation tool.
- Provide end-user security features to the service creation tool.

For detailed requirements on work package level see Appendix D (separate document).

6.3.4 **R8.3.4 – Service Creation Operation & Maintenance**

The Service Creation Environment shall exist of set of tools that support creation of new components/services.

Regarding operation and maintenance for service creation the SPICE Platform Infrastructure shall comply with the following:

- Provide a provisioning interface in the newly created component/service
- Provide support for component/service creation (e.g., editors, browsers, etc.)

For detailed requirements on work package level see Appendix D (separate document).

6.3.5 **R8.3.5 – Service Creation Runtime Execution**

The Service Creation Environment of SPICE platform infrastructure shall provide simulation facilities in order to test the newly built component and/or service.

Regarding runtime execution for Service Creation Environment shall comply with the following:

- Provide support for simulation of newly built components/services.

For detailed requirements on work package level see Appendix D (separate document).

6.3.6 **R8.3.6 – Service Creation Non-Functional**

The Service Execution Environment shall be able to handle mass-usage service execution and low latency.

Regarding non-functional requirements the Service Execution Environment shall comply with the following:

- Provide support for mass volume execution
• Provide support for low latency within service execution.

For detailed requirements on work package level see Appendix D (separate document).
7. Open market requirements

This section provides an overview of the open market requirements for the SPICE platform. A successful operation of the SPICE platform requires open market provisions that support different players and their role, responsibilities and position in a next generation mobile value network. The open market requirements identify a range of business constraints and objectives invoked by other providers who have access to the SPICE platform, such as (mobile) virtual network operators, (third party) content and service providers and other business actors, and also addresses requirements related to multi-partnering in services provisioning.

The overall objective of the SPICE project is the enablement of large-scale service introduction through the support of multiple heterogeneous execution platforms, in order to facilitate the diffusion of new services across different operator domains, different countries and multiple terminals. Easy and quick service creation and dynamic service instantiation and deployment are instances of benefits that could result from the application of the SPICE platform to the service creation process.

A further goal of the SPICE project is to allow a large variety of business models. To achieve this objective, seamless service and information roaming, continuity of service provisioning, opening up to third parties of sharing of service components and automatic discovery of new services are relevant. Service provisioning increasingly requires cooperation between a large number of parties – telecom operators, content and service providers, third party network and service providers, end-users – who all need to collaborate to deliver a service.

Taking the above into account, the main starting points for the composition and selection of open market requirements are:

- An open service creation and operation environment for a wide range of content and service providers (professional, semi-professional and individual)
- A managed service platform environment. This is not contradictory to the previous point, as long as the SPICE platform is accessible to any party that wishes to operate under the framework provided by the platform. For users, this requirement could be fulfilled through a secure framework that provides identification, authentication and privacy management functionalities, whereas for content and service providers this could be realised through service level agreement enforcement functionalities
- A seamless user experience, with seamless information and service roaming, as well as service provisioning continuity and technology transparency
- Clarity and trust, in order to create a positive acceptation environment for consumer adoption of innovative services
- User control over the content and service provisioning, heavily related to the previous point
- Billing and charging options for a wide range of situations and devices
Recent trends and evolutions in business models for content and service provisioning also have an impact on the open market requirements. These current trends include:

- Multi-channel approaches (e.g. triple play, quadruple play)
- New service offerings through the emergence of new technologies (e.g. personalisation, interactivity)
- Emerging mobile markets (e.g. mobile television)
- Increasing importance of portals, platforms and EPGs (e.g. (re)packaging of content, listing of content)
- Intermediation versus integration trend in the value chain (e.g. Apple iTunes that fulfils an intermediary role between content providers and customers versus BBC that operates almost all components of the service provisioning, from content creation to actual delivery of the content)

Based on these starting points and trends, the open market requirements have been subdivided in the five different categories that are discussed below. The detailed list of requirements can be found in Appendix E. A relation can often also be observed between enterprise and open market requirements, where the open market requirements mainly add the component of openness to a wide range of (third party) service providers.

### 7.1. Charging, billing and revenue sharing requirements

Charging, billing and revenue sharing is related with the need for flexible multi-actor revenue sharing models in a dynamic context and the need of offering a flexible range of payment options to consumers and merchants alike. Creating a clear and comprehensive tariff structure is also an element with high importance for user adoption of SPICE platform services, as is the granularity of the purchase process. Regarding charging, billing and revenue sharing, the SPICE should offer or support following elements:

- The steering of payment options in a checkout module
- A range of possible payment options
- An internal revenue sharing system
- A clear tariff structure
- Functionalities for content purchase and access

### 7.2. User Profile requirements

Regarding user profile requirements, the SPICE platform shall support various types of profiles and a range of complementing functionalities. Profiles not only decrease the content and service adaptation burden on users and increase the potential towards personalization and customization of services, but can also increase the flexibility of users towards changing providers if the profile is portable. User profile requirements include following elements:

- A consumer billing profile
- Management of the entity profile, portability of the user profile and interoperability of its data model
- User privacy rules
- Support for identity management and data traffic control as well as control enforcement by the end user
- Convenient access from different locations and terminals to the same services, with related authentication and authorization functionalities
- An open terminal operating system
- A user-defined set of subscribed services and support for feasible subscription combinations

7.3. Terminal requirements

Regarding terminal requirements, the open market requirements mainly reside in the area of supporting multiple devices. Users on the SPICE platform shall be able to access and consume content on a wide range of devices and even should be able to swap content consumption between devices. Requirements related to these aspects include:

- Multi terminal and multi access Dynamic Communication Sphere (DCS), multiple devices support
- The device management related to the configuration of the network access and the communication services

7.4. Business process support requirements

Regarding business process support requirements, the requirements in this chapter mainly relate to the aspects relevant for companies to operate in an open SPICE platform environment. Support of a wide range of business processes is required in an open market setting, in order to simplify the creation of content and services on the platform. Simplification of content and service creation entails business process support requirements such as:

- Access to usage data and anonymous user profiles and transparency and ease of use of tools for content and service creation
- Open access for multiple portals, support for multi-business models and dynamic and real-time creation of business networks
- Support for real-time multi-channel interactivity, user-generated content, preservation of third party look and feel, etcetera
- Aggregation from distributed domains, merchant billing profiles and the ability to find registered services
- Issues related to protected content distribution, the selection of the rights object issuer and policy updates

7.5. Interoperation requirements

Regarding requirements for interoperation between different parties, open market requirements for the SPICE platform are related to flexible, dynamic and on the spot
interoperation between actors. In order to obtain this goal, interoperation requirements shall include:

- Support for moving into heterogeneous networks and seamless hand-over between operators, devices and platforms
- Flexible operation, administration and management of the SPICE platform
- Support for third party access and discovery of and switching between content providers