
Deliverable D2.4
FP7-ICT-2009-5 257103
April 2012

# D2.4: Updates on Scenarios and Use Cases

| | | |
|---|---|---|
| **Project Number** | : | FP7-ICT-2009-5 257103 |
| **Project Title** | : | Secure WebOS Application Delivery Environment (webinos) |
| **Deliverable Type** | : | Report |

| | | |
|---|---|---|
| **Deliverable Number** | : | D 2.4 |
| **Contractual Delivery Date** | : | Mar, 31st, 2012 |
| **Actual Date of Delivery** | : | May, 20th, 2012 |
| **Title of Deliverable** | : | Updates on Scenarios and Use Cases |
| **Contributing Work Package** | : | WP 2 |
| **Nature of Deliverable** | : | Report |
| **Editor** | : | Fraunhofer |
| **Authors** | : | Oxford, Fraunhofer, BMW F+T, NTUA, TUM, IBBT, Polito, VisionMobile |

**Document History**

| Version | Date | Author (Partner) | Remarks |
|---|---|---|---|
| 0.1 | 30/03/2012 | Fraunhofer | First draft |
| 0.2 | 19/04/2012 | Fraunhofer | First complete export from webinos redmine and GIT repository |
| 0.23 | 23/04/2012 | Fraunhofer | Final review of chapter 4 and 5 |
| 1 | 15/05/2012 | VisionMobile, Fraunhofer | Included webinos cross-screen competition |
| 1.01 | 20/05/2012 | VisionMobile, Fraunhofer | Final review of chapter 3 |

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

**Abstract**

Deliverable D2.4 describes an updated set of user stories and use cases as input for the subsequent work packages. The goal of this deliverable is to define the scope of the webinos project; this allows readers to get an idea about what type of applications can be created using webinos. The deliverable also contains use cases giving an overview about what really is part of the webinos platform; this allows developers to better understand the benefits they can get out of webinos. The scenarios and use cases described in this document also cover the mobile, PC, automotive and home media domains.

As the webinos platform is now openly available to 3rd parties, it becomes more and more important to start capturing developer mindshare. Therefor the deliverable also describe the current status of the planning of webinos cross-screen competitions that will be targeted during 2012.

Updating, refining, and potentially creating new scenarios and use cases is based on the experiences and knowledge the consortium obtained during the first half of the project. With this in mind, scenarios were updated based on design and scope decisions made in the first year and a half. This also includes the creation of new scenarios to better address certain issues which were either not covered at all, or were not clearly described in the first set of scenarios. The same is also true for the use cases. Several use cases were updated or removed, while other use cases were added or completely refocused.

This deliverable also contains the outcome of collaboration activities by describing how the different selected projects could benefit from each other.

To improve traceability between scenarios, use cases, and requirements and to better understand the context of people, involved in scenarios, the personas developed in D2.7 were added to the scenarios. The original scenarios and use case categories were also modified to match the functional areas identified in the requirements deliverable (D2.2).

The use cases provide an input to the update of functional and architectural requirements described in Deliverable D2.5 and to WP3 – Specification of the webinos system. Scenarios and use cases also provide input to WP5 in order to define and implement applications that highlight specific features provided by the webinos platform.

**Keyword list**

webinos, requirements, landscape, app completion, cross screen

# Contents



This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 1 Introduction

This document provides scenarios and use cases as a foundation of the webinos project. This allows developers of the webinos system to further develop the requirements and derive the specifications in WP3. It also contains the outcome of collaboration activities that describe how related projects and webinos could benefit from one another. In addition this deliverable contains the initial webinos cross-screen completion plan.

## 1.1 Intended Audience

This document provides scenarios and use cases that form the foundation of the webinos architecture to be used for requirements engineering and specification work in WP3. The updated document also includes mappings between use cases and the webinos architecture so platform developers can see where use cases steps are realized in the implementation. It also addresses third parties interested in webinos by describing what type of scenarios (applications) can be fulfilled (made) by using webinos.

## 1.2 Background

After roughly one and a half years of project progress, the webinos platform architecture is stabilising. As a result, it has become clearer what webinos is, and what the scope of the webinos platform should be. With this in mind, the first batch of use cases and scenarios were updated to reflect the project progress. In addition to updating existing scenarios and use cases, new scenarios and use cases were created when appropriate and superfluous use cases and scenarios were deleted.

Since scenarios and especially use cases evolve over time, for example based on feature requests from in- or outside the project, this deliverable is a snapshot. The project maintains a GIT repository for scenarios, use case, and requirements; this always contains the latest version of these artifacts, and is available to all project members for inspection and, where authorized, update.

Also as the webinos platform is now openly available it becomes important to start capturing developer mindshare. To support this webinos will run cross-screen developer contests wherefore the initial plan is described in this deliverable.

## 1.3 Structure of the Deliverable

- Chapter 2 explains and recaptures the methodology used for collecting scenarios, use cases and involved stakeholders.
- Chapter 3 introduces the webinos cross-screen completion plan
- Chapter 4 provides the updated, restructured and new scenarios.
- Chapter 5 contains the use cases based on the scenarios.
- Chapter 6 summarizes the contributions of this deliverable.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 2 Methodology and Terminology

Prior specifications to be defined in work package 3 webinos uses three classic levels of abstraction to define the webinos system. Namely scenarios, use cases and requirements. The basic meaning and separation between the three levels are defined as follows:

- A scenario describes in a narrative way of envisaging how a person directly, or indirectly, interacts with webinos. Scenarios are used as a basis to define the features that should be implemented. The scenarios are written by stakeholders of the webinos project and are their main instrument to influence the scope of webinos. Scenarios are also candidates for application development in order to demonstrate the potential of webinos and are one of the ways to show users what webinos can do.

- A use case is a behavioural specification which captures a contract between stakeholders of a system and its behaviour (COCK2001). In many cases within webinos, this is typically an application on behalf of the user. Occasionally, this is a user that interacts directly with webinos while installing or configuring webinos-enabled application software. In this deliverable, use cases define a subset of the functionality of webinos. They are primarily used to define the behavior of a system without specifying its internal structure. Although use cases are useful to developers because they describe what functionality is available without going directly into a specification. If a developer does wish to consult these specifications, use cases are also useful because they describe how these specifications are operationalised.

- System requirements are optative descriptions which describe what the system has to do, or what properties that system functions must have (RORO2007). These statements are described as precisely as natural language text can allow such that they are both readable and usable by all webinos project stakeholders. For these requirements to be useful for software architects and developers, the origin of these requirements should be traceable, as well as how these requirements have been operationalised and implemented. Scenarios provide some measure of a requirements origins, as do use cases which may be derived from scenarios or, equally, operationalise how a requirement is implemented in architectural terms. System requirements are not part of this deliverable. An updated version of webinos system requirements (Deliverable D2.4) will be described in Deliverable D2.5.

Specific methodologies for scenarios and use cases are given in the related chapters below.

## 2.1 Stakeholders

The stakeholders of webinos are described below:

User

- Of a webinos application,

- Discovers availability of application,

- Selects Application for use and installation (including pre--conditional commercial steps),

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Configures application according to his personal preferences,
- Uses application to perform intended actions,
- Share knowledge about application availability with other users (e.g. invite/encourage to install).

Developer of webinos Applications

- Creates software packages,
- Develops applications using various programming languages and toolkits (IDE, SDK, etc…)

Developer of webinos Platform

- Creates functional specification and reference implementation of the webinos Platform,
- Provides developer community support where application developers are engaged (e.g. the webinos project team).

Application Service Provider of installable webinos Applications

- Offers applications for consumption (by means of installation) on devices under a business model of their choice (e.g. retail, direct, free, etc.),
- Provides application lifecycle management,
- Manages application developer relations.

Application Service Provider of hosted webinos Applications

- Runs webinos Applications „in the Cloud“ on behalf of a webinos Application User,
- Offers hosted instances of webinos platform components.

Device Manufacturer

- Offers execution environment for the webinos runtime,
- Supports webinos discovery functionality for devices and applications,
- Demands webinos to add tangible user benefit by means of making webinos applications available on their devices.

Network Provider

- Of Internet Connectivity (not local / short--range connectivity),
- Provides and manages access to the internet for users as well as for application service providers. 3rd Party Service Provider
- Supplies functionality outside of the webinos platform and functions set to webinos App Developers which they can use in their Apps.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 3 webinos cross-screen competition

The following section is work in progress and is subject to change based on review/feedback from legal consultants, requirements of and negotiations with event organizers and finally on the experience that we will gain during the first competitions. The final terms and conditions as well as the expected prize amounts will be submitted to the European Commission for consideration and approval separately from this document. The final terms and conditions, processes and results of the competitions will be included in deliverable D02.9.

## 3.1 Description

### 3.1.1 Summary

Developer competitions with generous cash prizes have become commonplace in the developer community building strategy of technology & developer tool providers.

As the webinos platform is now openly available to 3rd parties, it becomes more and more important to start capturing developer mindshare. This is particularly important for the early adopters; those that will turn to platform experts and eventually platform evangelists, a core ingredient for a thriving developer community.

webinos will organize a series of developer contests aiming to:

1. Get insight and feedback from 3rd parties about developing on webinos platform
2. Tap into innovative ideas / use cases.
3. Generate marketing buzz.
4. Stimulate media traction.
5. Attract competent developer teams and individuals to use and become involved in webinos.

European Commission (EC) will support the competitions by offering a number of cash prizes in order to increase the appeal to developers and strengthen the brand messaging.

The following sections describe the current status of the planning, in terms of, overall framework & processes, terms & conditions and physical events that will be targeted during 2012. As several of the items are still under revision, for example due to the on-going negotiations and discussions with event organizers, the content of these sections is subject to updating. Webinos is aiming to build the contest framework in a way that can be easily replicated and used in several occasions, whenever webinos identifies an opportunity for growing the webinos developer community and raise awareness around the project.

Other competition formats (e.g. a hackathon) may be considered at a later stage when webinos platform has reached a certain level of maturity and popularity.

### 3.1.2 Framework and outline

The overarching aim is to keep the process simple and repeatable in order to be able to run more than one contests in the duration of the project. At the same time we want to be flexible enough in order to

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

easily run a competition in conjunction with physical events such as courses, workshops and conferences in order to leverage co-marketing, tap into greater audiences and also gain from the marketing assets created as part of a physical event.

A competition has five main phases:

1. Contest announcement
2. Submissions open
3. Submissions closed & filtering based on evaluation for eligibility
4. evaluation/selection of runner-ups
5. selection & announcement of winner(s).

If the competition leads to a physical awards event (e.g. part of conference or workshop) where there is opportunity for long airtime (e.g. 1hr) then phase 4 may be split in two as follows:

4a. A set of 5-10 runner-ups are selected before the awards event and invited to the event to do a 5 minutes presentation/demo of the application. Judges panel decides and votes during the event.

4b. For online-only competitions, winner will be announced online and notified by email.

#### *3.1.2.1 Effort*

Estimated between 0.5-2PM per competition depending on involvement and awards event complexity.

#### *3.1.2.2 Time-line/format*

Two different formats will be followed based on two different models: a short competition which will fit events with a monthly cycle and a longer cycle for competitions that require better marketing build-up and more logistics for the awards event.

The short competition timeline is shown in the diagram below. Note that the exact same format can be used when there is no awards event.

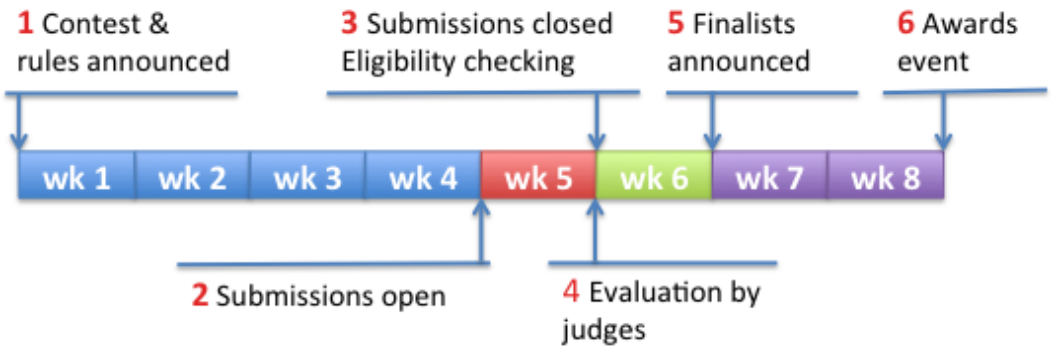



**Figure 1: short contest timeline**

The long competition timeline is shown in the diagram below. The timeline is based on the assumption that the competition is linked to a major event (e.g. LeWeb) therefore expecting much higher visibility. To this end, the early phases of the timeline are extended to allow for greater number of submissions

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

and better marketing build-up, and the later phases are also extended to accommodate longer processing due to more expected entries and logistics preparations for the physical awards event.

Note that in this diagram the boxes indicate months. More importantly, due to the more complex logistics and resource intensive nature of a larger scale competition this will ideally be done in collaboration with an external partner with relevant know-how.


**Figure 2: long contest timeline**

Below is a breakdown of the phases that are described above:

- Phase 1: Competition announcement (media, website etc.) for marketing staging & interest build-up).
- Phase 2: Application form/web-page is open and accepts submissions.
- Phace 3: Submissions are checked for eligibility.
- Phase 4: 5-10 finalists are selected by webinos judge panel and invited to the awards event.
- Phase 5: Finalists announced via media & website and invited to the awards event
- Phase 6: Awards event: where finalists present their apps and final winners are selected by judges or public (attendants) voting. Winners are announced and prizes are awarded.

#### *3.1.2.3 Selection criteria/guidelines*

Winner or winners will be determined by voting by a panel of judges selected by the webinos consortium.

The judges will evaluate the apps based on three criteria

1. Originality/innovation of the use case
2. Design / usability
3. Functionality & Level of integration with webinos (number of APIs used) & number of domains/screens spanning

All criteria carry equal weight. For each eligible application, Judges will give one mark for each of the above criteria in the range of 1 (worst) to 5 (best). Applications will be ranked based on the cumulative total of the marks. In case of a draw, the judges have the final saying in the decision of who the winners are.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

*Example:*

*App A (originality/innovation: 5, functionality/design/usability: 3, level of integration & multi-domain coverage: 2) - total = 5+3+2 = 10 points*

*App B (originality/innovation: 1, functionality/design/usability: 5, level of integration & multi-domain coverage: 3) - total = 1+5+4 = 9 points*

*Therefore App A (10 points) is ranked higher than App B (9 points).*

#### *3.1.2.4 Public voting*

For marketing and community strengthening webinos will use public voting in two ways:

a) The "Popularity Award": The general public is allowed to vote for their favourite entry by twitter. The winner (the app that collects the highest number of votes) will be featured on the competition website as the winner of the "Popularity Award". There is no cash or other prize involved in this award. The aim is to create buzz, capture public sentiment about the entries and produce marketing material. webinos has the right to disqualify any entry if there is suspicion for unfair practices to vote-up.
b) Allow public voting for the attendees of the awards event: In cases where this is feasible by the venue organizer, webinos may chose to allow the attendants of the awards event to vote in order to chose the final winners. The public voting process will depend on the format of the event (e.g. vote for your single favourite app, rank all the finalists or do a virtual crowd-funding game where the attendants get fake money and invest to the ideas they like).

#### *3.1.2.5 Judges selection process*

Three judges are selected and voted for by the consortium. Any member of the consortium can become a judge. If necessary, a set of names will be proposed and the consortium will have the opportunity to vote. People outside of the consortium can also be invited to act as judges, subject to the same approval process, in order to raise awareness and attract influencers to get involved with the project.

#### *3.1.2.6 Eligibility criteria / Rules:*

- Use of at least one webinos API & at least two screens.
- Open only to participants outside the consortium. No consortium partners are eligible for participation.
- Applications must be submitted and accepted at the webinos application portal in order to be eligible.
- By submitting an app to the contest, contestants agree to make the application source code available to webinos. The IPR of the entire application remains entirely with the developers.
- Apps must be accompanied by a clear description, screen-shots, and optionally (but we recommend to) with a video and/or screen-capture demoing their usage. The developer at his discretion may submit additional documentation and/or instructions.
- If an application requires special hardware (e.g. fixed appliances/machinery or very expensive equipment) or proprietary/confidential/sensitive data sets in order for its full functionality to be tested, the Judges may request additional proof of the functionality claims (e.g. arrange a live

demo) and agree on how this will be publicly demonstrated / show-cased during the awards event before they approve the submission. It remains to the Judges' discretion to disqualify an entry if they believe that it does not meet the criteria.
- Developers agree to complete an online survey at the end of the application form in order to be eligible.
- Developers agree to give at least one interview for quality/technical feedback and/or marketing purposes.

#### *3.1.2.7 Prizes*

Prizes are structured with respect to the size of the associated event and the anticipated media traction. At the moment two events have been identified as primary targets. The following prize structures are brought to the consideration of EC for approval:

Small competitions:

**1st prize € 1,000**

**2nd + 3rd prize € 500 each**

The Mobile Monday Athens competition (see details in Section 2.1) is the first competition to use this prize structure.

Another small-scale competition (online only) is currently under discussion in collaboration with the W3C/MobiWebApp team as part of an upcoming W3C online training course (within the W3DevCampus program).

Large competitions:

**Option A: 1 prize: € 10,000**

**Option B: 2 prizes: € 5,000 each**

At the moment, the prestigious LeWeb Paris event is the main target for the first webinos large-scale competition (see details in Section 2.2).

Where applicable and subject to approval by the consortium, webinos may enrich the above prizes with additional offerings (e.g. mobile phones, gadgets, trips) sourced by the industry partners or external sponsors in order to increase the appeal of a competition for the developers.

#### *3.1.2.8 Leveraging the relationship with the finalists*

- winner/runner-ups need to agree on producing/be included in marketing (e.g. photos, app screenshots, names in PR, blog posts).
- Selected winners may be invited to join webinos and have their ideas implementation "supported" by WP5
- Winners may be invited to follow-up events
- webinos "gurus" web-page on webinos website where finalists get permanent display of their achievement as webinos experts.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

### 3.1.3 website

For uniform branding, ease of administration and common look and feel a micro-site under the main webinos website can be used in order to host the webinos competition related information. Applications will be handled in the form of a simple web-form including the contestants contact info, application name, link to the webinos appstore, additional optional documentation and link to youtube/vimeo etc. video demoing the application, and questionnaire that will be prepared by the consortium.

### 3.1.4 marcoms

All current webinos social media channels will be used to perform outreach activities for the competitions, including twitter, facebook, linkedin. In addition to the webinos channels the competition will be posted to a number of relevant websites including:

http://www.f6s.com

http://www.contestwatchers.com

http://www.ycompetition.com

Other candidate target seed websites are:

http://reddit.com contest & giveaways

http://digg.com news/technology/media/recent

http://stumbleupon.com Programming, Web development

http://slashdot.com

http://clipmarks.com

http://delicious.com

http://news.ycombinator.com

Furthermore, subject to time and resources allowing, seed efforts could be extended to conversations that are relevant to the topic on the following:

http://quora.com

http://answers.onstartups.com

http://youngentrepreneurs.com

Forums, Blog comments

The competition will also provide content for an upcoming press-release.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

### 3.1.5 Terms and conditions

A first iteration of the terms and conditions has been prepared and is currently under revision (see Appendix I)

### 3.1.6 Open issues, dependencies & next actions – timeline

The following issues need to be in place before the competition can be announced:

a) Education material for developers new to webinos in order to get started quickly (to be available by May 14th)
b) Get approval by the EC for the prize amounts (week of May 14th)
c) Get legal feedback/support on the exact Terms and Conditions (to be reviewed by May 28th)
d) Appstore: add a "friendly" URL and "webinos" branding (to be available by June 4th)
e) Set-up webpage and submission form (webpage to be available by May 14th – submission form by June 11th)
f) Compile questionnaire for the contestants (to be available by June 11th)

## 3.2 Venues

Following discussions and research during March & April two venues will be the primary targets for webinos contests in 2012:

a) Mobile Monday global event series
b) LeWeb Paris 2012
c) co-organized competition with the W3C/MobiWebApp as part of the W3C "Mobile Web 2: Applications" online training course.

Webinos had been in touch with the following event organizers:

- http://leweb.net/
- http://www.mobileworldcongress.com/
- http://mobilemonday.com/
- Future of Web Apps
- Future of Mobile
- Future of Web Design
- Future Insights Live
- Future of Social Business
- http://www.apps-world.net/europe/
- W3C

### 3.2.1 Mobile Monday Athens – July 9th, 2012

#### *3.2.1.1 Introduction*

Mobile Monday Athens (MoMoAth) is a unique opportunity for running the 1st webinos competition as it combines a global brand, broad reach and a medium size (circa 200 attendants). It is also an extremely attractive venue at that time of the year. Equally importantly, the theme of MoMoAth July 9th is going to be "**Developing for Multiple Screens**" which perfectly fits the scope of webinos. Note that this is an ideal test-run of the process, which can be then easily replicated in any similar event in the future.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

MoMoAth will host a "**webinos cross-screen prize**" competition. Webinos will have full control of the submission and winner selection process.

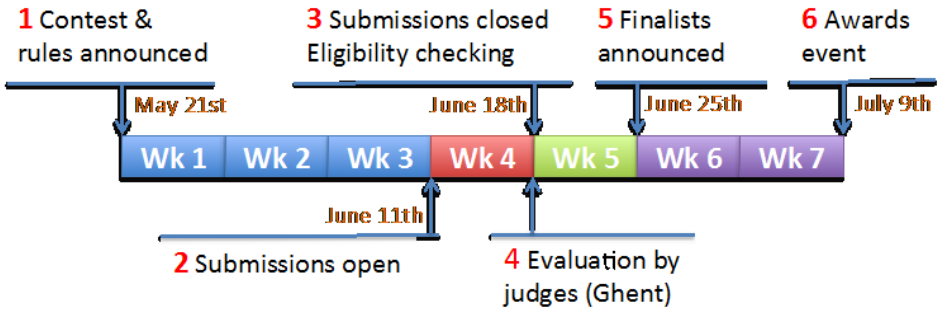



**Figure 3: MoMoAth contest timeline**

#### *3.2.1.2 Details*

- Because of VMs involvement with MoMoAth, webinos can run an own-branded "**webinos cross-screen prize**" competition hosted by MoMoAth and added as competition sponsor on the website without sponsorship cost beyond the prize money.
- The competition will run according to the timeline of figure 3.
- The rules, terms and conditions are defined by webinos (as per Section 1.2.5)
- webinos will also get a presentation slot during the normal session equal to all other speakers
- webinos will be "competition organizer" and "big prize sponsor"
- Up to 10 runner-ups (depending on number/quality of submissions) get 3 minutes presentation of their app and 5 mins Q&A.
- webinos will have control over who's winning the prize
- **Webinos prizes: 1st prize 1,500 EUR, 2nd prize 500 EUR, 3rd prize 500 EUR.**
- MoMoAth may add another prize to the best app by an external sponsor in the form of a mobile device (handset, tablet).
- MoMoAth may be able to provide hotel accommodation for overseas runner-ups subject to confirmation.

#### *3.2.1.3 Effort:*

Estimated 0.5-1PM

### 3.2.2 Le Web – Dec 4th-6th, 2012

#### *3.2.2.1 Introduction*

VisionMobile will run a multi-sponsor app competition at LeWeb Paris 2012. Webinos gets an opportunity to receive media coverage (http://www.salesforce.com/fr/events/leweb/leweb-infographic.jsp) in the prestigious LeWeb conference by becoming the big prize (€10K) sponsor for the best webinos-enabled application without extra sponsorship cost, subject to finalization of the sponsorship terms & conditions which are expected to be finalized within May.

The competition will be announced at LeWeb London in June. The exact format of the contest will be available early June. Below is some additional information about the specific event:

- Last year there were circa 700 submissions.
- Heavyweight influencers attending (we want to be on their radar),
- Organization and logistics to be taken care entirely by VM (i.e. no webinos overhead)
- It will be a multi-sponsor competition. More details about the specific tiers will become available late-May / early-June.

#### *3.2.2.2 Effort:*

Between 1-2PM

#### *3.2.2.3 Winners/prizes:*

- 1 webinos cross-screen prize of €10000 cash

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 4 Scenarios

The first half of this chapter details the phase I scenario update process and provides a snapshot list of scenarios (as mentioned before, scenarios and use cases which result in architectural decisions can change over time). In the second half of this chapter we look at several related projects to use them either as additional scenario or use case source or to look at problems they are facing which webinos technology might possibly address.

## 4.1 Update of Phase 1 Scenarios

In Phase I, WP2 concluded with a selection of 23 scenarios (previously called user stories) delivered in D2.1. The aim of D2.4 was to revisit the existing scenarios to align them with the actual webinos architecture. With this, this deliverable contains 15 scenarios that describe webinos from a user's point of view.

## 4.2 Summary of Changes

Scenarios have been revised in a number of ways. First, they have been re-written to better envisage webinos as we now understand it. As such, scenarios are now an accurate showcase of webinos either currently does, or will do based on the perceived timetable of delivery. Second, we have aligned the usability research carried out as part of D2.7 and D2.8 by incorporating elements of the context of use description into these scenarios In particular, the personas developed for D2.7 have been introduced into the scenarios, which have also been updated to better reflect their needs and expectations. Tasks and misusability cases from D2.7 and D2.8 have also been incorporated into these scenarios where this was also felt to be appropriate.

Scenarios which were deemed to fall outside the current scope of webinos were either deleted or, where appropriate, behavioural elements of interest were incorporated into other scenarios.

Finally, some modifications emerged from discussion with other projects or external topics (Smart Cities, Cloud4All).

## 4.3 Scenario Template

The scenario template of phase one was extended to also include a short overview of the described scenario and a usability breakdown. The following template was used for describing our scenarios.

**Title**: The title of the scenario. Also contains the abbreviation of the functional area it belongs to the most.

**Author**: The original author(s) of the scenario.

**Overview**: This is a short introduction to the scenario. It might describe the Persona's objectives when engaging in the activity, or introduce some consequence arising from the use of webinos.

**Description**: This is a short vignette describing what happens when a Persona uses webinos. The narrative need not be long; a few paragraphs should suffice. It just gives an overview, how a user could

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

interact with the webinos features in "normal" language which could be direct as with using an application that is doing the direct interaction with webinos.

**Issues**: Specifying the important matters in the story for the different actors, i.e. what the different actors 'desire' out of the story and/or what limitations they have today (e.g. a car may not always be online).

**Benefits**: Specifying what the different actors get out of the story to clarify the value for webinos users in this story and in comparison to the situation as it is today.

**Usability breakdown**: This describes the usability aspects of the involved personas. There is one table for each persona involved in the scenario.

| Attribute | Description | Possible values |
|---|---|---|
| Persona | The persona involved in the scenario. | The persona as defined in D2.7 used in the scenario. |
| Duration | The time taken by a persona to complete the scenario. | None, Seconds, Minutes, Hours or longer |
| Frequency | The frequency a persona carries out the scenario. | None, Hourly or more, Daily - Weekly, Monthly or less |
| Demands | The mental or physical demands on a persona. | None, Low, Medium, High |
| Goal Conflict | The degree to which the task interferes with the persona's work or personal goals. | None, Low, Medium, High |

**Required Use Cases**: Lists derived use cases which must be provided by the system to enable the story in general.

## 4.4 List of Scenarios

### 4.4.1 S-TMS2: Single Sign-On, Multiple Devices

Author: Anders Isberg, Shamal Faily

#### *4.4.1.1 Overview*

Before going to work, Georg interacts with several personal devices.\n

#### *4.4.1.2 Description*

Georg has a mobile phone. When Georg reaches for his phone first time in the morning, he uses his credentials to authenticate against his OpenId identity provider; this enables access to all services he is authorised to use with his identity. During breakfast Georg decides to read the latest news and the latest Facebook status updates on his tablet computer. To avoid manually re-authenticating, Georg puts the Mobile internet device and the tablet together; the tablet then transfers the authorisation from his phone. After reading the news, Georg leaves home and drives to work. As Georg enters the vehicle, he puts his phone in the built-in device holder; this transfers authorisation from the phone to the in-car.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.1.3 Issues*

How can authorisation can be transferred across devices without introducing new security threats and making the user experience less intuitive?

#### *4.4.1.4 Benefits*

The end user should be able to use the personal services without needing continually authenticate across multiple devices, while still enjoy personalised services with a reasonable level of trust.

#### *4.4.1.5 Usability breakdown*

| | |
|---|---|
| Persona | Georg |
| Duration | Minutes |
| Frequency | Daily – Weekly |
| Demands | Low |
| Goal Conflict | Low |

#### *4.4.1.6 Required Use Cases*

- TMS2: Data Synchronization
- PS3: Authentication

### 4.4.2 S-CAP2: Report Disorder

Author: Shamal Faily

#### *4.4.2.1 Overview*

Peter wants to report an act of civil dis-obedience to the police.\n

#### *4.4.2.2 Description*

It was 2315 and, after a night at the opera with some friends in Frankfurt, Peter was on the tram back to his apartment. Peter was about to doze off when was stirred by the sound of shouting outside. The tram had stopped for a few minutes at Luisenplatz; this is fairly common given it's position as an interchange for several tram and bus routes. Peter looked out of the window towards the square and the source of the disturbance. An intimidating group of youths with a strong penchant for black clothing and Alsatians was sat at the foot of the statue of Ludo. Based on the slurring of their speech, their excessive volume, and the several half-empty crates of Schneider-Weisse, it is clear to Peter that everyone in the group was quite drunk.

As he surveyed the scene, a large, shaven-head member of the group stumbled over to the Ludwigsmonument with a spray-paint can. As the sounds of an impromptu rendition of "Feuer Frei" rang out over Luisenplatz, the youth began spray painting graffiti onto the base of the statue. While Peter found this wanton act reprehensible, other passers-by in the square were acquiescent. With no police nearby, no-one wanted to challenge the behaviour of a large, clearly inebriated group of youths with dogs.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

Peter quickly entered "vandal graffiti" into the text field beside the photo, assigned the event to a "Graffiti" category, and clicked on the Send button. Meanwhile, the youth had returned to the group, who had now determined that the flash came from Peter's tram. As the group advanced menacingly towards the tram, a bell sounded and the tram slowly pulled away. Relieved, Peter returned the phone to his pocket, and hoped that his action had made a difference.

#### *4.4.2.3 Issues*

- The politics of web applications. Might an EU funded project facilitating civil order applications provide the impetus for a dystopian, Big Brother society?
- When users provide authorisation to web applications at install time, do we expect their expectations to be the same when an application is used?

#### *4.4.2.4 Benefits*

None

#### *4.4.2.5 Usability breakdown*

| | |
|---|---|
| Persona | Peter |
| Duration | Minutes |
| Frequency | Hourly or more |
| Demands | Medium |
| Goal Conflict | None |

#### *4.4.2.6 Required Use Cases*

- DA1: Virtual Device
- PS7: Interpreting policies and making access control decisions
- LC1: Installation and update of webinos applications

### 4.4.3 S-CAP1: Creating Applications for webinos

Author: Victor Klos

#### *4.4.3.1 Overview*

Jimmy wants a rapid development environment of cross platform web applications\n

#### *4.4.3.2 Description*

Jimmy loves to develop programs. He has created several good-selling apps for the iPhone. Also, his blog is well visited and he makes quite some cash with the corresponding add income. His expertise is creating well working mobile apps in the category of ‘life hacking’.

He has heard of webinos and decides to take it for a spin. Luckily, it is open source as he is getting more and more frustrated with the closeness of the iStore. Better still, after download he can do a local install. After only 15 minutes he is running his own development webinos cloud.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

He soon gets himself familiarized with the webinos concept and APIs and ports his latest iPhone app ZenSearch to webinos. That application searches the web for photos (in the public domain or with a certain CC licence) which are usable in Zen-style presentations. It consists of a back-end and a front-end. The back--end is a server that does the searching and 'scoring' of the photos. The front-end is the aforementioned iPhone app with its excellent navigational interface. For the porting effort the server, of course, is fine as it is. The mobile app however needs to be rewritten to HTML + CSS + JavaScript. This is okay, because the website of the ZenSearch service and the mobile app will have more in common and are thus easier maintainable. Finally, because webinos bridges to the DLNA home network, his app cannot only suggest beautiful pictures but also show them on the End Users' TV. All that for just a couple of JavaScript lines!

#### *4.4.3.3 Issues*

- Jimmy wants to learn new technology but is often overwhelmed before he begins. For example a simple experiment with a scripting language once required him to do a make and install with system administrator rights.
- Jimmy learns quickly. However, new paradigms he often finds hard to comprehend. (His favorite learning style is just to start experimenting.)
- As a professional and agile developer Jimmy likes his development, test and deployment environments clearly separated while highly manageable.
- Jimmy would very much like to create apps that use the television, but he is too unfamiliar with UPnP, DLNA and what else.
- Being forced in a single direction usually puts off Jimmy. He has an allergy for notions like 'the one platform that rules them all' or 'the mother of all platforms'. He likes to be able to make his own choice which abstraction level to design his applications on.
- Jimmy likes to control his own stuff. His NAS is securely reachable from the internet using public/private keys and he runs his own mail server for all his domains.

#### *4.4.3.4 Benefits*

- webinos provides a smooth installation experiencing, running out of the box on all supported platforms. (When a new technology advertises ease of use it should behave like that itself, even during installation.)
- webinos provides a great learning experience that caters for all learning styles. Developers that take it for a test drive are in no way hindered to choose for webinos.
- Having formal and separated environments for development, test and deployment is considered to be good practice.
- Having full access to webinos functionality using HTML + CSS + JavaScript makes the target developer audience as large as possible.
- Being able to access webinos functionality using native APIs attracts the more advanced developers which can in turn create more advanced solutions.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Integrating with other devices, especially if they use lesser known protocols like UPnP requires a lot of work. Having webinos do the bridging to its familiar environment both scales up webinos functionality quickly and makes it easier for developers to create compelling apps.
- Allowing federation between (i.e. seamless interworking of independent instances of) different webinos clouds maximizes the number of people that will actually use it in their daily life.

#### *4.4.3.5 Usability breakdown*

| | |
|---|---|
| Persona | Jimmy |
| Duration | Hours or longer |
| Frequency | Daily - Weekly |
| Demands | Medium |
| Goal Conflict | None |

#### *4.4.3.6 Required Use Cases*

- NM1: Subscribe and Publish Events
- TMS2: Data Synchronization

### 4.4.4 S-DA2: Input Method Dictionary Sharing across Devices

Author: NTUA, IBBT, ISMB, Samsung, Polito

#### *4.4.4.1 Overview*

Have a relaxing trip thanks to smart input methods.\n

#### *4.4.4.2 Description*

When Georg starts a trip in his car, he is using the webinos Enabled Tweeting application on his mobile to "get-in" his car. This allows the app to retrieve the destination and the estimated time of arrival. The car is equipped by a navigation service, it can accept commands by touching physically the console, through vocal command, and can also accept command from the other enabled webinos device, like the mobile of Georg (which itself can accept different kind of inputs). The accuracy of these methods benefits from or even depends on populating a dictionary of non-standard words and phrases which each user employs. In this case, all input methods can benefit from the dictionary previously created by Georg using all his webinos enabled device, and shared among all of them. The Georg's in-car webinos system looks for available dictionary application among its reachable devices. But his webinos-enabled laptop is switched off, and his webinos-enabled TV (at home) has been instructed to not synchronize data when Georg left his house, to save unnecessary bandwidth consumption. Anyway, the smart-phone dictionary is ready and up-to-date.

So, Georg can pilot the navigation system of the car accurately, and instructs it to posts to Georg's Twitter account "George is on the way to the Alps, be there in 4 hours" while safely driven, and edit/complete that default phrase with "he's ready for a beautiful week-end"

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.4.3 Issues*

- All the input methods working on the user devices would have to use the same dictionary, which means that all the input methods would have to be rewritten for the webinos platform.
- Devices may be offline when the dictionary is needed, so local synchronized copies of the dictionary will still be required.

#### *4.4.4.4 Benefits*

- Users will have access to their dictionaries from each of their devices. They will need to define a new entry in a dictionary only once, which can happen automatically in most cases and from this time on the new word will be automatically recognizable on all of their devices. Users can choose whatever input method they like at the time, which may vary according to the device and context, without having to teach new words to each input method separately.
- By capturing anonymized word lists from the Cloud (where permission is granted) the default dictionaries can be frequently updated so that newly popular words do not need to be added by each new user.

#### *4.4.4.5 Usability breakdown*

| | |
|---|---|
| Persona | Georg |
| Duration | Minutes |
| Frequency | Monthly or less |
| Demands | Low |
| Goal Conflict | None |

#### *4.4.4.6 Required Use Cases*

- DA4: Discovering the application
- TMS2: Data Synchronization

### 4.4.5 S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files

Author: Claes Nilsson, Dieter Blomme

#### *4.4.5.1 Description*

As the lifespan of people is growing, the number of people having at least one chronic disease is increasing. Personal medical sensors and mobile communication could play a big role for the treatment of chronically ill persons and also improve their quality of life. Furthermore, sensors and mobile communication makes remote monitoring possible, which is convenient for the patients and saves money for healthcare providers.

In the following story, Clara is a teenager with diabetes. She uses her favorite blood glucose meter to measure her blood glucose levels. This meter is also able to communicate with internet connected

computing devices such as her laptop and her mobile smart phone through a standardized local communication method (USB, Bluetooth, BLE, ANT+ etc).

Clara has found several interesting cloud-based web applications for diabetics. These applications help Clara analyzing her blood glucose levels so that she can control her disease. The applications could for example display all her glucose levels for a certain time period, display diagrams showing average values throughout a day, show statistics on glucose levels within target/below target/above target, etc. She runs these web applications not only on her laptop but also on her mobile smart phone as well as on the family's internet connected TV. When she runs the applications they discover that there is connectivity with her blood glucose meter and all values stored in the meter are transferred to the computing device and are processed by the diabetes web application. Clara's blood glucose meter can be used by any diabetes web application that fulfills any standardized manner to retrieve values from the blood glucose meter.

The most common case is that the blood glucose meter is connected to the computing device and values are transferred when Clara starts the diabetes web application. It is also possible to have a continuous connection so that each measured value is instantly sent to the computing device. This is especially applicable for future non-invasive (checking blood glucose levels without puncturing the skin for a blood sample) continuous blood glucose meters. For elderly and possibly demented diabetics who are not capable of understanding and acting on blood sugar values by themselves continuous blood sugar measurement with instant transmission to a wireless connected computing device and further direct uploading to a health care provider web service may be necessary.

A webinos enabled diabetes web application also has the possibility to store Clara's blood glucose values in a diabetes file in the cloud. It is important that this file is stored in a secure manner and that no unauthorized third parties have access to it. The file could be shared with her health care provider based on Clara's decision. For example, all values handled by the diabetes web application could continuously be stored in her diabetes file in the cloud and be made available to her endocrinologist, her nurse, her dietitian, etc. This would for example allow the doctor to spot persistent out-of-normal-range episodes early. The doctor and other medical parties could also update the diabetes file with information from the laboratory e.g. HBA1c (which reflects the average blood glucose level during the last 2 - 3 month period), provide recommendations on modifications to the diet or medication and remind Clara of important upcoming events such as the expiration of prescriptions.

Alternatively, Clara can make her results available when she visits her diabetes doctor for her three-monthly check-up. A scenario can be that Clara takes out her mobile phone and starts her webinos diabetes web application. She goes to the Sharing my file page. The application automatically searches for webinos enabled devices in the vicinity. The doctor's mobile phone shows up on the list and webinos automatically provides Clara with the necessary data to check the security of sharing the file. Clara can then choose to share her file with the doctor using this mobile phone. Now the doctor gets a notification on his mobile phone and his PC that Clara has shared her file with him and he can now check the results and update the file where applicable.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.5.2 Issues*

- Users are not able to run web applications such as a diabetes web application that automatically retrieves data from a sensor such as a blood glucose meter in the proximity of the device.
- There is currently no easy way for web application developers to create applications that access sensors in the proximity of the device. Only a few specialized APIs to access native functionality exist, e.g. the W3C Geolocation API.
- New types of sensors are invented all the time. There is no system for providing web developers with easy access to sensor data in a generic manner.
- Users want medical data to be secure. This is a major issue for data stored in the cloud. No unprivileged access should be given to third parties.
- The sharing of the medical file should be authorized only by the end user, who is the owner of the file. No third parties should be able to edit the sharing and access settings.

#### *4.4.5.3 Benefits*

- The User is able to use her favorite sensor-aware web application, e.g. a diabetes web application, with any favorite sensor of the type that the web application needs, e.g. a blood glucose meter.
- Data from new sensors that are invented will be available to web applications.
- The user is able to share sensitive data, e.g. medical data, in a secure manner with all relevant parties.

#### *4.4.5.4 Usability breakdown*

| Persona | Clara |
|---|---|
| Duration | Minutes |
| Frequency | Daily - Weekly |
| Demands | Low |
| Goal Conflict | Low |

#### *4.4.5.5 Required Use Cases*

- DA1: Virtual Device
- NM1: Subscribe and Publish Events
- CAP1: Sensors and Actuators
- PS7: Interpreting policies and making access control decisions
- DA3: Virtual Device Ownership

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- TMS1: Linking device to a user account
- PS3: Authentication
- DA2: Network Independent Virtual Device
- NC1: Content Adaptation

### 4.4.6 S-DA3: Converging Applications within and across Devices

Author: Hans Myrhaug, Dr Ayse Goker, Anders Isberg, Andrea Paul, Andrea Atzeni

#### *4.4.6.1 Description*

Peter is considering buying a new car of the BMW 3--series. He likes the style of it and especially that he can travel to Munich to personalize for his individual interior choice, exterior, but also for infotainment, engine, tires, and safety choices.

Justin accepts to go to the BMW museum with Peter, since interested in cars, music and fashion.

When Peter arrives to the Museum, at the reception desk He may get a museum guide for mobile with free connection to the wireless network. He accepts and installs the application.

The museum guide allows to be shared on different devices and users, and Justin happily ask to share it while in the museum.

Through the guide, Peter discovers that he can literally hold the phone close to each artifact, and relevant information is automatically presented on his mobile. Meanwhile Justin discover the possibility to also view videos describing the history of some BMW series.

Those videos comes with extra audio and subtitle streams for different languages. The videos has also a special stream for hearing impaired users where the environmental noise is reduced in order to make the actors speech clearer.

It is a huge advantage for the friends that the movie has these different audio streams. Justin is not as good as Peter in understanding German and prefers video in the English language anyway.

#### *4.4.6.2 Issues*

For some users it is uncomfortable to consume the same content together. Especially language or physical restrictions can influence the experience. One issue here is to provide the content in an appropriate format while using an appropriate device to the user's private device in order to improve the user's content consumption experience.

The scenario requires that the devices involved have wireless, contactless, and camera capabilities. It also requires from the converged webinos applications that data can be instantly and seamlessly transferred between the devices and applications through open standard interfaces. Applications can additionally run in parallel. Applications and data can also be added automatically to the cloud in the background while applications are being used.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.6.3 Benefits*

An easy integration of private “output streams” (which could be audio, video, or text maybe even with real-time translation) allows for an improved social life style. webinos allows to deliver content in a format that can be handled by the user’s device.

Providing converged applications enables application developers to incrementally roll out applications as an evolving package consisting of several smaller applications. These can be glued and interlinked together with each other. They can start/pause/resume/stop each other from within the application. Additionally the applications can run in parallel. Devices instantly connect to each other to enhance aspects of the information experience for the end users. The story also involves the use of contactless and wireless transfer of data between devices, and in addition to this, the use of the camera on the mobile to obtain object recognition and visual search for information

#### *4.4.6.4 Usability breakdown*

| | |
|---|---|
| Persona | Justin |
| Duration | Hours or longer |
| Frequency | Hourly or more |
| Demands | None |
| Goal Conflict | None |
| Persona | Peter |
| Duration | Hours or longer |
| Frequency | Hourly or more |
| Demands | Low |
| Goal Conflict | None |

#### *4.4.6.5 Required Use Cases*

- DA4: Discovering the application
- DA6: Finding devices in close physical and social proximity (alternative)
- NC1: Content Adaptation
- NM4: Proximity detection of Services

### 4.4.7 S-DA1: Smart Device Integration

Author: Andre Paul, Andrea Atzeni

#### *4.4.7.1 Overview*

Alice wants to enhance her experience of watching movie thanks to new device\'s technological opportunities.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.7.2 Description*

Alice is sitting in front of her Internet enabled TV and is browsing the web. She install a new application which allow access to a community of eighty's romantic movie fans (providing clips and lots of information about them) which was recommended by Bob, a friend of Alice. As usual, there is some profile information that must be provided to complete the registration procedure and become a member of the community.

Because webinos has build in user identity features, the application can retrieve most of the needed information to pre-fill required data. From the PZP, the application may retrieve first name, surname, address, preferred nick name, etc. Additionally, the application credentials can be stored in the Alice personal zone, and provided to the application when required. Thus, if webinos is running there is no need for manual logins later.

The webinos identity mechanism provided most of the needed information but there are still some fields left which should be filled out by Alice. Additionally, Alice may want to change some pre--filled values which should be unique to the application or at least different to the proposed values.

Alice decides to change and add some values. Due to common remote control restrictions, such as the absence of easy to use text input facilities on the TV, this is not an easy and comfortable task if a TV remote control is used. Completing profile information could be time consuming tasks.

With webinos each webinos-enabled device that provides text input capabilities could be used to insert text for an application running on a TV set in a user friendly manner. Alice has several text input sources in her environment. A common desktop computer with an attached keyboard, a smart phone, and a tablet. Thanks to the "webinos-enabled keyboard" application running on each of those devices, all of them can be used as input media for another device in the personal zone. In this way, there is no need for yet another keyboard for yet another screen; each keyboard can be used for each screen. Or, even better, only one keyboard is needed to control all the devices.

The social community application needs some information from Alice and therefore the application presents alternative input methods. Alice decides to use her tablet instead of the TV's primary input source (the remote control). Additionally she selects to use her tablet also in the future without asking.

To complete the registration Alice wants to attach a current picture of hers to her profile. The social community app can query the available photos of Alice in the personal zone, but no one of them seems appropriate to Alice. She wants to make a new photo right now. The social app retrieves a list of usable cameras, and due to a policy, all but the smart phone one are filtered out since are not in close proximity to Alice, which selects her smart phone device, anyway equipped with a pretty good camera. She also decides to use her smart phone any time in the future if a camera is requested by the application. After the picture was taken is attached to the profile and the registration is complete.

Because Bob proposed to join the community, Alice wants to get in contact with Bob who is currently also online in the community. Alice initializes a video chat with him to tell him that she is now also a member. She clicks on "video chat with Bob" and this time Alice is asked to use her smart phone which also notifies access to the camera. After the connections are established both videos streams, Bob's and Alice's, are shown in real time on the TV.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

While both are talking together Bob receives a message from another friend of both whose name is Charlie. Charlie is currently on the way sitting on the backseat while his father is driving. The application allows multi user video chats while all chat participants are shown in case that the application is shown on the TV screen, but will only show the current speaker if shown on small screens.

Bob invites Charlie to join the video chat and to talk about what to do next weekend. Charlie's connection does not allow for video streaming, due to limited in-car's bandwidth but instead allow for an audio-only one. The application can adapt to the local availablity, and by default choose the audio-only option when it runs in bandwidth -constrained devices.

The social community app offers upcoming events which are collected from different external sources and presented to the user. The three friends are clicking a while through the events and later they decide to watch a movie that is currently running in cinemas. Bob clicks on "participate on event" and selects all video chat participants as attendees. With this click the event is added to Alice's, Bob's, as well as Charlie's mobile phone calendars.

#### *4.4.7.3 Issues*

The issues addressed by the story are directly related to the "one virtual device" idea of webinos. Each hardware or software based service of any device could be used by any of the other devices. For example using a mobile phone's camera to take a picture and directly present it on the TV screen without any direct user interaction for the transfer of the data. In general, Alice wants to use any feature or service of any of her devices in a common integrated way realized by webinos applications, here the TV as primary application execution environment, to not rely on multiple devices with the same functionality for each device that executes applications.

#### *4.4.7.4 Benefits*

With webinos it is unnecessary to have each service or device needed by an application directly attached to the device that is executing the application. This also covers the aspect that even uncommon components for a specific device can be used in a transparent and integrated way.

#### *4.4.7.5 Usability breakdown*

| | |
|---|---|
| Persona | Alice |
| Duration | Minutes |
| Frequency | Daily - Weekly |
| Demands | Low |
| Goal Conflict | None |

#### *4.4.7.6 Required Use Cases*

- DA1: Virtual Device
- NC1: Content Adaptation

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

### 4.4.8 S-LC1: Designing Policy-aware webinos Applications

Author: Ajit Jaokar, John Lyle, Shamal Faily, Ivan Flechais, Hans Myrhaug, Ayse Goker

#### *4.4.8.1 Overview*

Unlike the web apps he usually develops, Jimmy has a number of additional considerations when developing for webinos.

#### *4.4.8.2 Description*

This story is described from the perspective of an application developer who is writing and supporting a webinos application with privacy and security concerns that require policy management.

Jimmy is developing a webinos application to support the social network site Twitter, and wants to develop a user interface than the website, including bigger input fields, location--based services and offline features such as local storage of tweets. Jimmy starts looking at the webinos API and, while the UI development seems straightforward, location-based function involve making an access-control request to the underlying system, and handing access denied requests. Jimmy designs his application to handle these access denied responses gracefully.

After developing an initial tablet-based prototype with resources he has access to, and making note of what resources his application needs, he checks his application against a set of sample end-user privacy policies. These samples are representative of real policies and, by developing with these in mind, Jimmy can predict when calls might require approval by end-users. In two cases, Jimmy notices that privacy policies request more detail about why access to a piece of personal information – stored in the user's profile – has been requested by his application. He decides to modify the application such that the user's date of birth is not collected, and a better rationale is provided for why location data is needed.

Before installing the updated prototype to an Android handset, Jimmy tweaks the installation settings for the reasons that the application needs, and the reasons why they are needed. During the installation itself, the installer provides an explanation for the resources that need access, and whether or not this conflicts with the end-user privacy policy that Jimmy is currently working with. After installing the app, Jimmy tests that the application still continues to provide what he considers a satisfactory user experience.

As the application design becomes more apparent, Jimmy decides to use webinos' analytics feature to find out more about the services used by the customer, and where they live. After experimenting with the webinos context module, the policies that control it, integration with other analytics APIs, and the impact that privacy policies might have on all of this, Jimmy decides that collecting analytical information is easier said than done. Consequently, he decides to think more carefully about what data he needs to collect, and how to collect this would impacting runtime performance or the expectations of his end-users.

#### *4.4.8.3 Issues*

- Recent web services have a record of encroaching on the privacy expectations of customers. This can affect the trust that users have in a system or an infrastructure.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Developers are not always given the tools to work with user privacy requirements, and end up making applications which require access to as much data as possible, purely to simplify the development process.
- Users have their own opinions of what data usage is acceptable. Applications currently dictate the terms in which they are used rather than negotiating data disclosure versus functionality offered.
- Application developers want their program to be used, and want to take advantage of device features. However, they also do not want to scare potential users, and therefore need to request only the capabilities that their applications need to have access to. Should these still be unacceptable, it is better to provide limited access than none at all. This will encourage users to be more adventurous and adopt new applications.

#### *4.4.8.4 Benefits*

- All access to private user data is mediated and may be controlled by the user.
- Companies developing applications can do so against known security and privacy policies, so they can improve the user experience and avoid endless pop--ups and requests for access permissions.
- Developers are encouraged to design applications to make use of as little personal data as possible.

#### *4.4.8.5 Usability breakdown*

| Persona | Jimmy |
|---|---|
| Duration | Hours or longer |
| Frequency | Daily - Weekly |
| Demands | Low |
| Goal Conflict | Low |

#### *4.4.8.6 Required Use Cases*

- PS8: Interpreting policies and making access control decisions (alternative)
- PS7: Interpreting policies and making access control decisions
- LC1: Installation and update of webinos applications
- CAP2: User Profile
- PS9: Enforcing multiple policies on multiple devices

### 4.4.9 S-NC2: Learning Assistance for Disabled Students

Author: Heiko Desruelle, Shamal Faily

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### 4.4.9.1 Overview

A bad start to the day turns out for the better for Anna thanks to webinos.\n

#### 4.4.9.2 Description

It's Friday morning 7AM and time for Anna to get up to go to class. As Anna lifts herself out of bed, she realizes she's having one of her bad days. She has immense difficulty breathing and every physical effort is exhausting. She has been having these kind of seizures her entire life, and by now she knows that the only remedy is to take a day off and sit it out. Anna hates it when she has to skip classes. Although she has lots of friends that wouldn't mind lending her their notes, she prefers to be independent from others. However, the last time she tried to ignore her condition she ended up in hospital for three days. So she decides to get back in bed and use her laptop to access the university's online learning platform instead. It is an old laptop, but it has a custom-made keyboard with extra-large buttons to improve accessibility.

The university recently extended its learning environment to support webinos-enabled devices. As she browses through the available course material, she can't help wondering what all the webinos fuss is about, because she doesn't notice any changes. However, after selecting today's presentation she gets an unexpected notification. It states that nearby devices have been found which are better suited to render the requested content. Surprised, though intrigued, Anna accepts the suggestion to access the presentation from an alternative device. Within a blink of a second, her television set is automatically woken from it energy-saving mode and a few moments later the first slide is showing up on its screen. "Great", she thinks, "how do I navigate this thing?". Because of her limited motor skills, especially at this time of the morning, she isn't too keen on using the television's remote control. But her disappointment does not last long. When Anna turns back to her laptop, she notices its keyboard can be used to control the presentation on the television set.

#### 4.4.9.3 Issues

This scenario emphasises the need for automated adaptability processes, driven by the user context and device context. Learning applications that support a variety of disabilities require a fine-grained knowledge of the end user in his environment.

#### 4.4.9.4 Benefits

webinos enabled learning applications can provide high quality user experience, regardless of the disabilities that students might have. Learning applications can handle this type of variability and accommodate end users with an optimized experience.

#### 4.4.9.5 Usability breakdown

| | |
|---|---|
| Persona | Anna |
| Duration | Minutes |
| Frequency | Hourly or more |
| Demands | Medium |
| Goal Conflict | Low |

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.9.6 Required Use Cases*

- DA5: Finding devices in close physical and social proximity
- CAP2: User Profile
- CAP5: Wake-on-Webinos
- NC1: Content Adaptation

### 4.4.10 S-NC3: Background Monitoring Services

Author: Simon Isenberg

#### *4.4.10.1 Overview*

The scenario describes how to use monitoring services in webinos which can notify applications or a user, if a specific event is detected. Furthermore it showcases how applications can use information about the device to adapt the UI to the capabilities provided by the currently used device.

#### *4.4.10.2 Description*

Peter has to visit a business partner on the next business day. His business partner has recently moved to a new office and so he looks up the new address on his smartphone and pushes the address to his in-car navigation system using the webinos travel app. A few moments later a notification pops up on his smartphone from the FuelEconomic app. The app notifies him about the cheapest gas stations along the way. His car does not have enough gas to reach the business partner without stopping at a gas station, so Peter selects one of the suggested stations and pushes its address to his in-car navigation system. On the next morning Peter is guided to the selected gas station by the in-car system and then to his business partner.

In order to suggest appropriate gas stations the FuelEconomic constantly monitors the current position of the vehicle, consumption, range and next destination which is programmed into the in-car navigation system. If the service suggests a relevant gas station, Peter can request guidance to it on his in-car navigation system. The way how notifications are presented to Peter depends on the device he is currently using. When Peter is in his car, a selection menu of available gas stations is displayed on the central display and he can choose from one of the suggested stations. FuelEconomic adapts the layout of the selection list according to the input controls and the screen size of the IVI-system. When Peter transmits a new destination to his in-car navigation system using a different device, FuelEconomic checks the distance to the new destination and determines, if he has enough gas available to reach the destination or if a cheap gas station is along the way. In this case he already gets the information on his smartphone and can plan the trip accordingly.

#### *4.4.10.3 Issues*

- Users can be able traced by such background services.
- For in-car scenarios it has to be verified that notifications and applications do not distract the driver. Certain rules and guidelines have to be followed. webinos might need to monitor what an application wants to display on the in-car screen and deny certain actions

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.10.4 Benefits*

webinos enables applications to access features from remote devices in a seamless manner. Furthermore, webinos provides the information about devices to adapt the UI to the different device capabilities.

#### *4.4.10.5 Usability breakdown*

| | |
|---|---|
| Persona | Peter |
| Duration | Minutes |
| Frequency | Hourly or more |
| Demands | Medium |
| Goal Conflict | Low |

#### *4.4.10.6 Required Use Cases*

- CAP3: Background Execution
- PS1: Automatic login for External Services
- CAP1: Sensors and Actuators
- TMS2: Data Synchronization
- PS7: Interpreting policies and making access control decisions
- DA2: Network Independent Virtual Device
- NC1: Content Adaptation
- PS9: Enforcing multiple policies on multiple devices

### 4.4.11 S-PS3: Privacy Controls and Analytics for Corporations and Small Business

Author: Ajit Jaokar, John Lyle, Shamal Faily, Ivan Flechais, Hans Myrhaug, Ayse Goker

#### *4.4.11.1 Overview*

webinos compatible analytics and privacy preferences lead to different experiences when passing a department store.

#### *4.4.11.2 Description*

A central London department store wants to attract more customers by launching a retail advertising campaign taking advantage of information about the customers' mobile device to provide a more targeted and relevant campaign to individual customers. To do this, they have decided to launch the compaign which relies on ads placed within mobile web applications. If prospective customers are near a Joe Lewis store and running an application with placed advertising then, based on their preferences, they will receive a special offer entitling them to a discount on certain products. These product discounts will be tailored based on their webinos preferences. Joe Lewis will run this campaign using a large developer platform whose APIs support webinos. Support for webinos was important to Joe Lewis

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

when selecting a platform upon which to base their advertising. Because there is a growing emphasis on consumer privacy, Joe Lewis has made the decision to only use marketing tools that conform to EU data protection regulations; they understand that their selected platform partner conforms to these regulations due to their platform APIs rely on webinos for eliciting privacy sensitive information..

Three weeks later, Peter and Gloria are walking past this store on opposite sides of the shop, and in opposite directions. Peter, who is visiting London for the weekend, receives an alert from his diary app about a special offer on sound-proof headphones, which is fortuitous as he was thinking about buying a set of these only the other day. Meanwhile, Gloria receives no messages at all because, although the platform deduced that some dietary preferences had been defined for Gloria, her policy was such that she had explicitiy asked this information not to be disclosed.

#### *4.4.11.3 Issues*

- webinos will provide unparalleled amounts of information about users, and provides a basis for monitoring behavior across all the device domains. This is an attractive target for advertisers and analytics companies, and webinos must expect this scenario. To make sure that privacy is maintained, webinos must provide an easy and controllable method for implementing, monitoring, and auditing functionality.

- Many free applications are driven by advertising and marketing campaigns. This is good for end users, who gain an application and new functionality, but webinos must support the business model of the developers too.

- webinos APIs co-existing with similar APIs on other platforms.

#### *4.4.11.4 Benefits*

- Advertising, marketing and analytics are considered from a privacy perspective and webinos keeps control in the hands of the users of the devices.

- webinos caters for a wide variety of business models, including ad-driven applications and services.

- webinos can enable advertisers to tailor adverts to the right customer and offer discounts on products they have a genuine interest in. This functionality is balanced against the need for users to maintain the confidentiality of their private information and not share too much about themselves.

- webinos can be used to keep control of important data across multiple devices, despite the new threats posed by mobile and in-car systems.

#### *4.4.11.5 Usability breakdown*

| | |
|---|---|
| Persona | Gloria |
| Duration | Minutes |
| Frequency | Monthly or less |

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

| | |
|---|---|
| Demands | Low |
| Goal Conflict | Low |
| Persona | Peter |
| Duration | Minutes |
| Frequency | Monthly or less |
| Demands | Low |
| Goal Conflict | None |

#### *4.4.11.6 Required Use Cases*

- CAP2: User Profile
- PS9: Enforcing multiple policies on multiple devices

### 4.4.12 S-PS1: Usable Privacy Controls for webinos

Author: Ajit Jaokar, John Lyle, Shamal Faily, Ivan Flechais, Hans Myrhaug, Ayse Goker

#### *4.4.12.1 Overview*

A chance alert leads to Gloria updating her webinos privacy preferences.

#### *4.4.12.2 Description*

Gloria has a webinos-compatible mobile phone and car, and owns several online identities; she is particular keen on keeping her work and personal life separate. When using her mobile phone for work she uses her LinkedIn identity, and during the weekends she moves to a Facebook-provided identity. Outside work, she enjoys shopping and, on occasion, hearing about promotions and deals at her favorite shops. For this reason, she has configured webinos to accept nearby requests from sensors and networks to inform her of relevant special offers.

Gloria visits a nearby restaurant, and is alerted on her phone that the restaurant can take any information about her allergies and tailor menus specifically for her. After a few moments thought, Gloria thought this might be useful, so she agrees to share her allergy information. However, she also specifies that this information should not be shared with any other applications. On her return home, Gloria gives more thought about how the restaurants was able to alert her about allergy-specific menus. "Sometimes", Gloria thought, "I might want a little more privacy because I'm visiting somewhere more risqué or shopping for a surprise present." As she nows feels a little uncomfortable about some of her private information becoming publicly available, she decides to edit her webinos privacy preferences. In particular, she restricts the applications which might have access to her name, online identity, current location and device information. These revised preferences won't keep track of what she is doing or where she is going, and will keep her devices hidden from nearby networks. She can switch to this policy very easily, and does so whenever she feels she would like greater “nonobservability”.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.12.3 Issues*

- People have many electronic identities which they would like to keep separate, but current systems are bad at providing this isolation.
- Users do want to share data and receive personalized services. However, they need the tools to control what services are allowed to do and what data they disclose.
- Policy and privacy decisions change over time. What was once unacceptable may change, or a new application feature might persuade someone to alter their policy decisions. However, existing applications often force the decision upfront and do not make it easy for these changes to happen.
- Privacy violations can affect the trust that users have in a system or infrastructure.
- The protection of user privacy will become more important in a world dominated by sensor networks. Governments have an obligation to protect citizens and EU policies have been forward thinking in this respect. Software infrastructures should provide the necessary tools to let users make sensible decisions about their privacy.
- Privacy policies change according to domain and should be context enabled. For example -- home, venue (e.g. the café), office, street and city have different (and in some cases overlapping) privacy frameworks. In addition, specific applications will have their own privacy policies, for example in smart grids and health care.
- Policy management, no matter how well designed, can be time consuming and difficult for even technically-skilled end users.

#### *4.4.12.4 Benefits*

- Users are given both fine--grained control as well as intuitive simple control to manage their privacy and visibility.
- webinos can scale to future technologies such as sensor networks, smart grid and healthcare scenarios by managing the changing privacy and security issues.
- Users can share data with trustworthy services and applications with more confidence when policies are enforced.
- webinos can enable advertisers to tailor adverts to the right customer and offer discounts on products they have a genuine interest in. This functionality is balanced against the need for users to maintain the confidentiality of their private information and not share too much about themselves.

#### *4.4.12.5 Usability breakdown*

| | |
|---|---|
| Persona | Gloria |
| Duration | Hours or longer |

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

| | |
|---|---|
| Frequency | Hourly or more |
| Demands | Low |
| Goal Conflict | Low |

#### *4.4.12.6 Required Use Cases*

- DA4: Discovering the application
- CAP1: Sensors and Actuators
- PS7: Interpreting policies and making access control decisions
- PS9: Enforcing multiple policies on multiple devices

### 4.4.13 S-PS2: Implementing Government-mandated Privacy Controls

Author: Ajit Jaokar, John Lyle, Shamal Faily, Ivan Flechais, Hans Myrhaug, Ayse Goker

#### *4.4.13.1 Overview*

The EU decides to take a leading role in developing privacy profiles with the aid of webinos.\n

#### *4.4.13.2 Description*

The European Commission is interested in protecting the privacy rights of their citizens, and Carlos is responsible for helping people realize and protect their right to privacy. Carlos has seen many proposed solution, but these remain on paper alone. Carlos also sees that many of the new privacy initiatives that the EU wants to implement, especially for sensor based networks, may be implemented after deployment. For Carlos, that means potentially never. This is a concern and on behalf of the Commission, he sets about finding a solution before sensors become commonly deployed (with privacy becoming an afterthought).

Carlos has heard about webinos following a discussion with one of his colleagues. He likes the sounds of webinos because it is open source and web based, and thus potentially available to everyone. Because of its cross-device scope, he thinks webinos might form part of a privacy solution across current and future platforms. At an EU sponsored workshop on privacy and design, Carlos explains his vision to a mixed audience from academia and industry. He wants to work collaboratively with vendors to implement privacy policies that empower and protect EU citizens, and make webinos one of the mechanisms for a central, secure profile controlled by users, which is agnostic of any technology and can be accessed across all platforms. webinos will also be used to provide a client side dashboard (e.g. on a phone) on multiple platforms that accesses the privacy profile and uses it to enable or disable services. The dashboard will build upon previous work on EU projects, such as PrimeLife, while also being compatible with accessibility tools and technologies being developed by related projects such as Cloud4All . The systems developed by the commission will enhance the privacy profile in such a way that the latest privacy regulation guidelines are encoded in the profiles and are accessible through the dashboard.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.4.13.3 Issues*

- The protection of user privacy will become more important in a world dominated by sensor networks and analytics platforms. Governments have an obligation to protect citizens and EU policies have been forward thinking in this respect. Software infrastructures should afford tools that enable users to take sensible decisions about their privacy.
- There are expenses associated with the implementation of privacy frameworks which is hard to implement for most companies.
- Privacy policies change and it is not easy to implement the latest guidelines into products. So many privacy policies remain on paper only.
- There are research and development collaboration challenges across projects with broadly similar agendas, but subtle differences in how security and privacy expectations of different stakeholders should be satisfied.

#### *4.4.13.4 Benefits*

Different stakeholders, from software developers to device manufacturers, can comply with privacy directives by using existing components, functionality and policies.

- Regulatory authorities have a way to push privacy policies to webinos, with a potentially huge impact across multiple devices.

#### *4.4.13.5 Usability breakdown*

| | |
|---|---|
| Persona | Carlos |
| Duration | Hours or longer |
| Frequency | Monthly or less |
| Demands | Medium |
| Goal Conflict | None |

#### *4.4.13.6 Required Use Cases*

## 4.5 Collaboration Activities

### 4.5.1 FI-Content

#### *4.5.1.1 Project Overview*

The FI-Content project (CONTENT) (Future media Internet for large-scale CONTent experimentation) is part of several Future Internet projects which aim to improve today's Internet technologies that were designed back in the 1970s. Related projects are either focused on the technological aspect or on the scenarios and use cases. FI-Content is one of the content oriented projects and defines several scenarios which can be implemented using future internet technologies.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

The project's approach focuses on early proofs of concept to facilitate technical specification to deliver requirements to the FI-ware project (Future Internet Core Platform (CORE)).

The project as a whole addresses five key areas where today's Internet can be improved: better QoS and experience, direct Internet connectivity (a server-less world), usage beyond a purely static text/plug-in/browser based approach, better tracking of content, and the ability to provide support, failover and resilience across different providers (e.g. mobile devices, crossing between countries, or moving between different access points (e.g. home to library).

Partners involved in FI-content are coming from broadcasters and content production companies, telecom operators, and research institutes. Thus, most of the scenarios are dealing with content production, delivery, and enrichment. In the following section we examine some of the scenarios proposed by FI-content in order to find gaps and intersection points with webinos.

#### *4.5.1.2 FI-Content Scenarios and Use Cases relevant to webinos*

The following content is based on work-in-progress. Final use case and scenario selection defined by FI-Content may differ from the introductions given below.

With inputs from 5 content areas, spanning future uses of AV, games, Web, metadata and user generated content, demonstrating usage beyond current state of the art, FI-Content proposes a number of novel and inventive scenarios for new forms of content. The project provides research and user story creation around 5 different content areas, which together tackle some of the highest profile content groups. These use case families have been selected to ensure a comprehensive coverage of content based services, with a strong network dimension.

The 5 usage areas in the scope of FI-Content are:

- Games and virtual environments: The delivery and the interactive utilization of content, in most cases in a social relationship with other players,
- Professionally generated content: Production and distribution of content for and by professional broadcasters, combined with end user needs,
- UGC entertainment: Production and sharing of content by regular consumers,
- High End B2B services: A major evolution in the way business stake holders will be sharing or exchanging huge volumes of content data, opening the way to new business & service models,
- Edutainment & Culture: Content consumption by end users requiring specific new features for educational and cultural purposes and using novel education methods.

**Content Area A: Games and virtual environments**

In this content area the focus is put on games and industrial virtual worlds as two key application areas. Video games are an inseparable part of everyday culture. The explosive popularity of internet-based social games on platforms in the past few years is a testament to how strongly this particular genre resonates with the general population. However, technology limitations restrict the way in which users can interact with one another and prevent game designers from realising more compelling social game

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

experiences. Mobile games have been similarly popular but mobile game technology, however, is more restrictive. Bandwidth and lack of signal ubiquity mean that multiplayer and immersive experiences are severely hindered on all current mobile platforms.

This content area explores the ways in which the Future Internet infrastructure can extend gaming and virtual environments. The central component is a massive multiplayer mobile gaming and virtual reality platform that enables users to participate in innovative collaborative online world experiences. The platform behind it should allow game designers to move beyond the traditional gaming paradigm in which a player is fixed in front of a console or display. Instead, the game mechanics will blend the virtual and real-world experiences so that location-aware game play can extend in a ubiquitous way via Internet technology to mobile devices, smart phones, and other portable technologies.

Content area A is addressed with a single use case, “Gaming in a connected world”, for which several different scenarios address specific issues of that use case.

**Scenario 1: Augmented Reality Lemmings.**
A hand-held device uses an embedded video camera to display the player’s surroundings and embed synthetic creatures into the video, called Lemmings. The player must guide the Lemmings through the real world by building make-shift bridges, ramps, or other structures on which the virtual Lemmings walk. For example, in order to get the virtual Lemmings off the desk, the player must find some object in their house such as a board and use it to build a real, physical ramp that the virtual Lemmings can then walk across onto the floor. In this way, the game seamlessly mixes the virtual environment with the real one. Parents can view their child’s play through their own hand-held device that serves as a second lens into the augmented reality world. The game can be played by many people at once who work together to guide the herd of Lemmings from one region to another. In the limit, a pan-European cooperative game could require a huge stampede of Lemmings to be guided on a tour of Europe, from Helsinki to Sicily. All citizens can take part, or simply watch as spectators via live online video streams.

**Scenario 2: Smart Toys for Game Control - Simbio**
Current video game controllers lack tactile response and feedback. Future gaming platforms can replace complex game controllers with “smart toys” that contain embedded electronics such as GPS, cameras, microphones, gyroscopes, vibrating motors, animatronics, and other sensory feedback technologies. The player then interacts with the game by playing with the toy. If a child brings their toy to school or with them on a family vacation, sensors continue to relay all contextual information and play patterns to peers and cloud computing centres so that the virtual game evolves according to the child’s real-world play pattern, their detected emotional state, and other cues.

**Scenario 3: Mysterious Parallel Historic Attractions.**
The above technology can also be used to enhance learning and knowledge acquisition in any environment. Games can be built around historic attractions such as the Tower of London so that players view an augmented version on their mobile device in which they can see, overhear, and interact with historical characters. The conversations contain both information about the location and give clues as to where they should go next to solve a mystery. During the experience, players will also be able to share information with one another, pointing each other to particular things they should search for, and engaging with the historic locations to discover new depths of knowledge and understanding. The same

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

concept could be deployed in vacation hotspots such as the Disneyland Paris Resort. Tracking of user progress and areas of difficulty is accumulated and analysed to automatically update the flow and skill level of the story and the information provided, maximising the learning results, enjoyment and discovery of the experience. A dedicated social and transmedia portal would allow family and friends to follow the game's progress on the web or even contribute by solving riddles remotely.

**Scenario 4: Virtual Real Places**

On the flip side, the social gaming platform can be used to create a synchronized and always up-to-date live model of real-world places so that the virtual embodiment of a real-world opera, sports event, library, or university campus is always online. This "Virtual Earth" provides ubiquitous telepresence so that users can participate in industrial, entertainment, leisure, and education activities irrespective of their physical location in the world. Many other usage scenarios and potential business opportunities exists in this context. University campuses can be visited remotely and classes attended in real-time by students. Healthcare applications and patient monitoring can be assisted. By integrating interactive 3D graphics as a first class object into ubiquitous Web platforms, these virtual worlds will be widely available. This scenario lends itself to exploitation by startups and SMEs providing novel and unexpected use-cases in combination with large providers for the necessary infrastructures.

**webinos relation**

Scenario 1 is about an augmented reality game that needs advanced image recognition capabilities. Since webinos specifications propose to use W3C media capture and W3C real time communication working group APIs (webinos phase II) the game could be realized on top of webinos. Access to these APIs will be also possible directly in web browsers without webinos support. For the creation of the video stream between the kids to the parents the webinos Event API could be used to set-up the Web RTC communication channels (e.g., as also used in the webinos Kids in Focus application). With this the application would not need any server-side back-end to create the connection between sender and receiver devices.

Scenario 2 relates to the webinos architecture in a way that either the "smart toy" contains a PZP for exposing its device capabilities or by seeing the toy as "super dumb" device that exposes its features via a mediating device that runs a PZP. Basically the "smart toy" can be seen as a smartphone that has several sensors on-board to be used by remote service consumers.

For scenario 3 the same assessments as for scenario 1 apply. Finally, scenario 4 focuses on 3D modeling of real world places in order to render them later in Web browsers to provide applications that allow virtual visits of the digitalised places. Up to now webinos does not touch UI rendering capabilities. While FI-Content needs to evaluate how to implement this scenario and what deeper requirements are relevant for the 3D modelling WebGL (WEBGL) developed by the Khronos Group (KHRONOS), which is already implemented in Mozilla Firefox and Google Chrome, could be a good starting point.

Andrea: Not clear the last period, I suggest to rewrite.

**Content Area B: Professional Content**

Content Area B delves into the creation of content by users and into dialogues with professional producers with the integration of social media platforms. Therewith, it also includes niche content

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

delivery which mostly is not possible in broadcast scenarios and content recommendations. At the current state of the scenarios in content area B, it is not clear if and how webinos could provide needed technologies here or if webinos could add new features in the roadmap to support FI-Content scenarios in this content area. One thing that is mentioned and relates to webinos is the device to device interaction where it should be possible for a TV to interact, for example, with a smartphone to deliver supplemental services. This device interaction is already part of the core webinos architecture, in order to allow the creation of platform independent cross device and distributed applications.

**Content Area C: User Generated Content Scenario**

This vision is that user generated content (UGC) addresses the Future Internet by reducing the digital gap and by making the digital world access easier to a large part of the population. The UGC Future Internet services should at least involve the mobile and the tablet that are allowing to enlarge access to low revenue households, children and seniors. Real time and usability will bring easier exchanges between creators or creators and their public.

The following four scenarios are used to represent the content area for user generated content for collaborative content enrichment and handling for sharing:

**Scenario 1**
Creating, manipulating and sharing a travel road book in real time, where users can extract from any created content a specific scene and distribute it to people for free or through a payment service.

**Scenario 2**
Content distribution control, where users can store content once and make it accessible to authorised people. There is no way that this content will flow through the internet and be duplicated and used by anyone else.

**Scenario 3**
Community non-linear video storytelling, where users can create together an emotional interlinked guide for fellow people visiting the city. Each participant takes pictures, videos, text, audio from their favourite site and then make it available. When visitors walk through the city, it is automatically informed when a piece of content fits to its location.

**Scenario 4**
Birthday medley preparation and distribution where users can merge different types of content in a collaborative way from any type of device and display/distribute it in a authorised community.

**Scenario 5**
Collaborative content creation, where users can create and aggregate content addressing a specific subject together, each of them enriching this content adding new information.

**webinos relation**
In scenario 1 a road book is created, maintained and shared. With webinos, sharing data between users is possible, but depending on the circumstances, it might be too complex to set it up spontaneously. To share data and services in webinos, each user must be known to each other prior sharing data and using remote users services. This can be inappropriate for spontaneous sessions, for example at special events

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

or at the airport etc. Here is room for improvement for webinos, in order to also allow spontaneous collaboration or events to switch the personal zone to another one for a given time (like home PZ, work PZ, public PZ). The second part is about getting paid for user generated content. Theoretically this could be done by using the webinos Payment API and asking the user to register for a merchant ID, e.g., to use GSMA One API or Vodafone Bluvia. But this seems to be inappropriate. Better for this would be a payment provider that is designed to transfer money between any users and not only from users to merchants (an example for this is PayPal (PAYPAL)).

Scenario 2 is out of the scope of webinos. It relies on protecting share-able user generated content (" There is no way that this content will flow through the internet and be duplicated and used by anyone else."). Scenario 3 is about an application that allows the user to take different types of media and link them to places. Here, in addition to just taking content from somewhere, webinos APIs could be used to access content stored on any of the user's devices or even to create new content using media capture APIs. Also sharing user content and rendering content on different device types is provided by webinos. Thus, also scenario 4 can benefit from using the webinos platform. But the actual merging of different content should be up to a specific application that, for example, makes use of webinos functionality. Scenario 5 is just a generalized form of scenario 4. Thus, the same assessment applies.

**Content Area D: B2B**

With the digitisation of cinema projection, new revenue streams for theatre owners have opened up. Examples that have been tested worldwide are opera or sports in the cinema. Other try-outs have included gaming, rock concerts, ballet, etc. This content is captured at the live event and streamed to international theatres, in parallel. Existing setups use satellite links to support live alternative content. This setup has some problems with logistics, flexibility and cost. Getting the content in over FI connections would lower the threshold for this use case in smaller European theatres and open them up to this new revenue stream. Two scenarios are part of content area D.

**Scenario 1: streaming live alternative content into cinema**

A live event could be a sport event, music, theatre typically with a live audience. The local camera and sound crew send the content to the local director. Local editing is typical done in an OB-van. This OB-van will be connected to the future internet and will be the source of the content stream. Connected theatres can get the stream from the internet and pull it onto their system.

**Scenario 2: Streaming from an online archive of feature film**

Movie studios or distributors have a large catalogue of recent and older movies that they are willing to distribute worldwide. Using the Internet as a distribution and storage tool, makes this process much easier and cheaper to manage. The actual delivery to the theatres will be a live stream; the delivery from the studios or distributors can be a file transfer.

**webinos relation**

Content area D is about streaming content from content providers to cinemas. This is a quite specific use case that not only requires point to point streaming capabilities but also needs content protection mechanisms. Webinos is about distributed access of services and device features which also includes different users and dynamic scenarios. It would be possible to use, for example, webinos as a signaling

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

channel between providers and consumers (e.g. using webinos Event API) and doing the streaming out of band. But this approach can be seen as too artificial for these scenarios.

**Content Area E: Edutainment & Culture**

Content Area E is about a detailed described story that focuses on learning, during a field trip and cultural events. Overall, the scenario, which is described using multiple scenes, depicts a concrete application that allows the production of video, audio, and textual content and its publication to a central repository. Of course, the scenario includes the usage of several device types and thus should be implemented in a cross-platform manner. Cross-platform is one of the main topics of webinos and, thus, this is already covered within webinos. Because of this we abstain here from printing the whole story of content area E.

#### *4.5.1.3 Possible alignment with webinos*

Webinos may provide technologies to implement some of the scenarios proposed by FI-Content, as detailed above. When FI-content scenarios are matured some of them should be revisited in order to re-assess if webinos provides a suitable platform or if webinos lacks of certain features which could be part of the platform. The other way round, as long as the supporting project FI-Ware did not decided on the architecture framework for FI projects webinos is a good candidate for creating early demonstrators for FI-Content. Thus, its open for FI-Content to use the webinos platform for creating some of its early demos.

### 4.5.2 Omlette

#### *4.5.2.1 Project Overview*

OMELETTE aims at researching on the development, management, governance, execution and conception of converged services with a specific focus on the telco domain. OMELETTE will create a sound model of mashups that follows the REST architectural style (also supported by standard widget technology), as well as a standard specification of a mashup-containing platform that may guarantee portability and interoperability among different vendors and versions. OMELETTE will foster as well the reuse of existing components and mashups, thanks to its automated service discovery functionalities. Project OMELETTE aims at developing an open platform for building convergent mashups for the telco domain to be used within several industry-driven use cases.

The key OMELETTE Contributions to Open Source Communities:

- The **OMELETTE Live Environment** will be based on the open source project Apache Rave (http://rave.apache.org/). So, it shares the same objective: "It will provide an out-of-the-box as well as an extendible lightweight Java platform to host, serve and aggregate (Open)Social Gadgets and services through a highly customizable and Web 2.0 friendly front-end. As a result of OMELETTE contributions, Apache Rave will transparently support W3C Widgets using Apache Wookie."

- The **OMELETTE Widget Repository** is an extension of the open source product Apache Wookie (http://incubator.apache.org/wookie/), in order to be able to deal with the extended widget logic proposed by OMELETTE. Apache Wookie is a Java server application that allows you to

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

upload and deploy widgets for your applications; widgets can not only include all the usual kinds of mini-applications, badges, and gadgets, but also fully-collaborative applications such as chats, quizzes, and games.

- The **OMELETTE mashup environment** is an improved version of MyCocktail (http://www.ict-romulus.eu/web/mycocktail). In few minutes you can combine information obtained of REST services, this information can be modified with operators and later presented with a wide variety of renders.

- The **OMELETTE widget descriptors** contain a description of the external interfaces of the widgets and services known to the OMELETTE platform. The purpose of these descriptors is to make widgets and services available to the widget repository and the live environment; and to give them meaning by linking each widget's/service's interface elements to the OMELETTE ontology. OMELETTE will propose an extension of the W3C widget specification, in order to turn inter-widget communication and context-awareness into a first-class concept in the specification.

The OMELETTE toolkit and workflow is described in the following steps (http://www.ict-omelette.eu/omelette_approach)

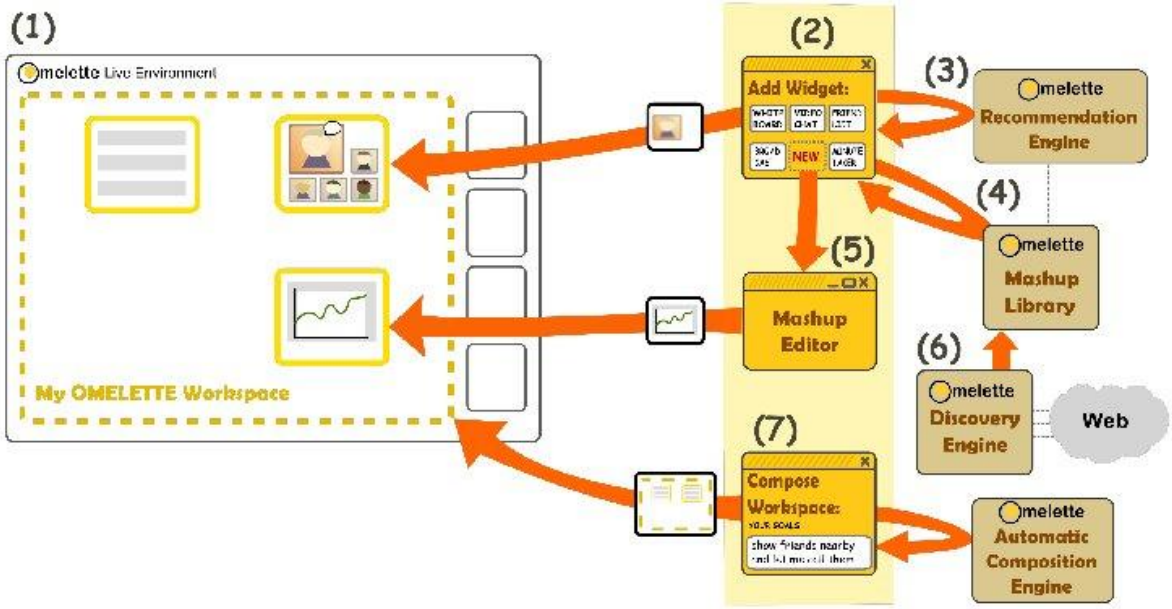



**Figure 1 OMELETTE Toolkit**

**1.** The OMELETTE Live Environment is integrated into a Web site and runs widgets within a workspace.
**2.** The intuitive and visual mashup environment allows its users to manipulate a running instance of a workspace directly - instead of models or code
**3.** During this activity, users will get recommendations regarding a composition for widgets based upon various conditions
**4.** If no suitable widget can be found in the mashup library
**5.** OMELETTE will provide a mashup editor for skilled users in order to create their own mashups in the form of deployable W3C Widgets

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

**6.** Automatic discovery algorithms crawl for components, available on the web, such as telco services, widgets or data feeds
**7.** OMELETTE will also support the automatic composition of workspaces out of a user goals, defined in a textual way

The OMELETTE partners are: LOGICA, HUAWEI TECHNOLOGIES CO LTD, SAP AG, SAP Research, TECHNISCHE UNIVERSITAET CHEMNITZ, TIE NEDERLAND B.V., UNIVERSITA DEGLI STUDI DI TRENTO., UNIVERSIDAD POLITÉCNICA DE MADRID, THE UNIVERSITY OF BOLTON, T-SYSTEMS MULTIMEDIA SOLUTIONS GMBH

#### *4.5.2.2 OMELETTE Scenarios and Use Cases relevant to webinos*

**Case study 1 - International Campus Education**

The University of Bolton is one of a number of education organisations that have an expanding network of international campuses and franchises. Currently Bolton has a campus in Ras al Khaima in the United Arab Emirates, and is opening another in Sri Lanka. One of the key challenges is to ensure that students and staff take full advantage of the international connections that are possible with the courses that are delivered in all our campuses. The university decides to enhance their existing university portal with the Live OMELETTE Environment. This enhancement will make far easier to bridge the gap between students and staff in the various campuses. The Live OMELETTE Environment will enable to run workspaces that support cross-campus student projects, collaborative games, and simulations.

In order to achieve this goal, a group of three computer science students have to develop within a 6 month project a small software product. They are supervised by a teacher who likes to introduce agile management methods and therefore needs to set up among other things a daily scrum meeting. Goal of the daily meeting is to update the teacher about the progress of their project. Besides the daily updates, they discuss problems and possible solutions within bi-weekly review meetings. Two of the students work from Ras al Khaima whereas the teacher and the third student are from the University of Bolton. To ensure a tight collaboration and knowledge exchange within such distributed teams the university uses the Live OMELETTE Environment. It allows an effortless set up of virtual conferences, provides capabilities to support discussion and collaboration and enables users to easily integrate external data and content in an ad-hoc manner. The following scenarios detailing the interaction with the Live OMELETTE Environment and demonstrate how it supports the team.

In this scenario, a **webinos-enabled Live OMELETTE Environment** could provide added functionality to the Live OMELETTE Environment by utilising webinoos APIs for resource sharing across platforms.

**Case Study 2 – Collaborative Decision Making**

Collaboration platforms similar to the one mentioned in the Case Study 1 can facilitate discussions and accelerate decision making in small companies and large enterprises. This case study demonstrates how managers and employees of a company benefit from collaboration features provided by OMELETTE. The underlying enterprise portal software is enhanced with mashup and telco features by the Live OMELETTE Environment. The existing application itself provides already basic web portal features, such as user management, content creation, change tracking, etc. In addition, it also includes basic commenting and notification features. The portal is organized in collaborative workspaces called

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

activities. Every user can create an activity and invite other users to join. The participants of an activity are able to jointly use the existing portal content together with the widgets provided by the Live OMELETTE Environment.

The product manager wants to discuss the cost data of a recently launched product. Therefore, he needs to talk to the head of sales and two colleagues from the marketing department. Goal of the meeting is to decide which pricing model the company should take in an upcoming marketing campaign for this new product. The product manager is located in Paris; the head of sales is currently on a business trip in Bulgaria and the marketing department is based in the company's headquarter in London.

In this scenario, **webinos** can be used to port the environment to multiple platforms and use hardware resources independently from the platform it runs on.

**Note :** OMELETTE provides only these two business oriented use-cases for the time being and only for the Live OMELETTE Environment, but there is an extensive description of how widgets are used to create mashups http://www.ict-omelette.eu/c/document_library/get_file?uuid=5f56096f-517c-4e3c-955f-60a78c441cb8&groupId=48739

#### *4.5.2.3 Possible alignment with webinos*

OMELETTE is relevant to **webinos** in the following areas:

- OMELETTE aims to provide definitions for mashup services and more specifically a telco mashup library that webinos can use to ease cross-platform interoperablity, which webinos might be able to use.
- OMELETTE aims to create a model for the definition of mashups that will speed up the process of adding new functionality with use of automatic discovery and description of mashup components.
- webinos can contribute its own functionality to OMELETTE, using the model OMELETTE proposes and creating collaborative environments that utilise the webinos APIs.
- webinos can extend its functionality significantly if it incorporates the broad array of third party apis from the OMELETTE project, avoiding to reinvent the wheel, i.e. re-doing existing web application functionalities.

### 4.5.3 Societies

#### *4.5.3.1 Project Overview*

The vision of SOCIETIES is to develop a complete, integrated Community Smart Space (CSS), which extends pervasive systems beyond the individual to dynamic communities of users. CSSs will embrace on-line community services, such as Social Networking in order to offer new and powerful ways of working, communicating and socialising. SOCIETIES will radically improve the utility and scope of future Internet services by merging social computing and pervasive computing through the design, implementation and evaluation of an open scalable service architecture and platform for self-orchestrating Community Smart Spaces.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

The goal of SOCIETIES will be achieved through four key objectives:

- To facilitate the creation, organisation, management and communication of communities via Cooperating Smart Spaces, where pervasive computing is integrated with social computing communities;
- To provide an enhanced user experience for both individuals and entire user communities, based on proactive smart space behaviour and dynamic sharing of community resources across geographic boundaries;
- To design and prototype a robust open and scalable system for self-orchestrating CSSs;
- To evaluate, through strong involvement of end-users, the usefulness and acceptance of the developed CSS software via three user trials with the following groups:
    1. **Enterprise Users**: Enterprise communities play an important role in bringing together people, goods and services within global markets, local ecosystems or large organisations. The CSS concept will bridge the gap between smart IT systems and established enterprise community activities.
    2. **Students**: Students adapt easily to new technology, and since communication and social networking play an integral role in their lives, they are most likely to adopt CSSs, using them in ways both foreseen and unforeseen.
    3. **Disaster** Relief Experts: The ability to rapidly form a disaster management community from all the closely located relief teams can help save lives, property, and the environment.

The Societies partners are:

TSSG, Intel, Heriot-Watt University, ICCS, NEC, IBM, LAKE Communications, Soluta.Net, SETCCE, Telecom Italia, TRIALOG, SINTEF, ITSUD, DLR, AMITEC, PTIN

Societies is a project that is linked with some of the webinos goals, through the implementation of context-aware applications that aim to incorporate, from a platform perspective, social and pervasive computing technologies. Societies might therefore be a significant tool for the expansion of the webinos Context API and User Profile.

#### *4.5.3.2 Societies Scenarios and Use Cases relevant to webinos*

An extensive set of scenarios and use-cases for Societies can be found at http://www.ict-societies.eu/files/2011/11/SOCIETIES_D22.pdf

**Scenario 1: Conference Scenario: Conference Organisation**

**Creating the conference communities:** The 'committee for research and innovation' agree to host a conference on their key research strategies for 2011. Peter, the elected coordinator, starts organising the conference and opens the invitations to participants by creating a community and its Community Interaction Space (CIS). Each community has a CIS where any interactions are supported. Although the

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

theme of the conference is based around the research strategies, other matters, such as: the specific topics, presenters and showcasing groupings, are not agreed at this point so Peter opens these up for discussion to the conference social network.

**Registration of Attendees:** Participants that register an interest or confirm their attendance are automatically suggested by their CSS to areas in the open discussion, where they might have an opinion, based on their expertise and previous conference engagement thus the conference communities are already forming before the conference officially starts. In addition to contributing to the conference topics, attendees supply details of their research domains, interest areas for networking and collaboration, and their arrival/departure times, to help with transport logistics. Linkages between attendees of similar interests are created and opportunities for engagement are suggested. It also allows Peter, as the coordinator, to see more information in advance, such as numbers of attendees for each session, catering numbers and transport requirements. Peter invites all support parties, such as catering and security, to join a conference organisation CIS where relevant information about the attendees can be shared so they can plan ahead (e.g. menu creation)

**Sending Personalised Attendee Info:** Closer to the event, registered attendees are provided with personalized information, via their CSS, that includes travel directions, local amenities, the event programme, a list of other attendees/represented companies, and information about presentations/sessions, so as to allow for the preparation of questions. The users will be provided with suggested personalised session agendas by their CSSs that are based on their interests, preferences and priorities for the event. This allows them to see what free time they may have for networking with other attendees.

**Organiser Feedback:** During and following the conclusion of the conference, Peter will be able to get real-time feedback from the attendees as to what worked well/poorly, opportunities for improvements and knowing who attended what sessions.

This scenario can be implemented through the use of applications created in webinos and contextual information stored through the webinos Context API. Much of the data mentioned in the scenario (e.g. attendees arrival/departure times, attendees of each session) can be automatically captured and made available by the webinos Context API and can be exchanged throughout the network, by the extended properties of the webinos User Profiles and communications.

The deliverable also contains more interactions throughout the conference, that make use of similar functionality to the one mentioned above and as such can be implemented by webinos applications.

**Scenario 2: University Scenario: Indoors scenario in University Campus**

**Student communities:** Barry is alerted to important communities that he is strongly encouraged to join, particularly the "New students" community that all starting students can join to meet other students and exchange information like tips on campus services and student societies. He also finds he has been automatically added to some communities that represent certain courses he is to be taking.

**Research Budget Distribution:** Barry has become a new student at a time when the University's budget distribution is to be decided, with many different budget proposals up for debate. Employees from the HWU staff community can comment on and debate the budget proposals via community communication channels. CSS technology is used to identify who is most interested in what proposals and who commonly communicates with whom. Based on this, the HWU CSS creates several sub-

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

communities to streamline discussions. Student opinion is also important to the discussions but rather than asking students to explicitly vote on such issues the HWU CSS implicitly gathers information from student CSSs about what resources and facilities they use most often (by monitoring resource usage, location, etc). In the end the budget is decided and the HWU CSS identifies that the sub-communities should be deleted based on inactivity.

**Event Alerts and Interactive Directions:** Barry has a lecture to go to. He is running late and hasn't had time to check emails or text messages so he doesn't know that the lecture was actually relocated to lecture room 6. His CSS knows that his intention is to attend the lecture (in line with the intent of other community members). His CSS also knows the location of the other members of the group, therefore it directs him to lecture room 6 through directional speakers and monitors in the HWU building. Based on his intent to attend the lecture and his predicted future location (lecture room 6) his CSS has also automatically transferred all the necessary files from his home PC to a workstation in lecture room 6.

**Proactive Disco:** Barry goes along to a Freshers' event called the "Proactive Disco". It is a community based disco that takes into account the music preferences of all people currently dancing on the dance floor and decides on what music tracks to play. Sensors are used to identify who is actually dancing.

**Organising a Snowboarding Trip:** Barry is a keen snowboarder. One day while bored in his dorm he tells his CSS he would like to go snowboarding. His CSS notes this intent and automatically creates an ad-hoc community to organise a trip to the dry ski slope nearby. Friends who are on campus, free and like snowboarding are identified and included in the ad-hoc community. Friends with cars are also identified. The CSSs of all ad-hoc community members coordinate a meeting time and place on the campus to suit everyone.

**Shared Cooking experiences:** After a few weeks at HWU Barry has become a member of several other communities. One is the "Student Dorm Cooking" community, which compares cooking ingredients provided by members and suggests that community members with compatible ingredients get together to make a meal between them. This community also factors in group food preferences and learns who prefers to dine with whom over time. Different incentives and awards are given to community members for various reasons e.g. the most active members or members who are considered the best cooks.

**Mood based personalization:** One day while wandering in his department building, Barry wears an earpiece which connects to his CSS. By using various bio-metrics on Barry, the CSS can roughly gauge his mood, and it records his current mood as "Contented". His CSS selects appropriate music to match his mood and plays it via his earpiece. Another student, Donald, approaches. Barry's CSS detects Donald and knows that Donald is a friend. As a result it stops the music, tells Barry that Donald is approaching and reminds him that it is Donald's birthday. It also detects Donald's mood as advertised by his CSS and informs Barry that Donald is "happy". Since the two belong to the same community, "CS Hill walkers", a CIS is automatically established linking.

The combination of biometric data and context-aware applications is also part of the webinos use-cases as well as several other data mentioned in this scenario. Webinos can be either used to implement the scenario by use of applications that follow the Societies specifications, but use webinos connectivity and the Context API, or webinos specifications can be extended to incorporate Societies proposed functionality.

The deliverable also contains more interactions throughout the student life that make use of similar functionality to the one mentioned above and as such can be implemented by webinos applications.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *4.5.3.3 Possible alignment with webinos*

The Societies scenarios and use-cases deliverable also contains several more scenarios and use-cases that are of similar relevance to webinos, including **Disaster Management**, **Entertainment** and **Health**. The common denominator between them is the interconnectivity of devices, the exchange of personal information and the multi-platform context-aware application that exchange data from device sensors or external devices through internet services. The focal point of Societies is not the same as in webinos, but there is a significant overlap between them. Both projects can see benefits from the other. Webinos can extend the Context API and the User Profiles to make use of the pervasive computing capabilities of Societies, whereas Societies can use webinos as the platform to implement/extend its specifications, by making use of the common interfaces and communications between devices that are implemented in webinos. More importantly, since Societies proposes a set of multi-platform applications that implement their models, webinos can be the framework on which these are built, facilitating the development of many of the many components that are necessary for the scenarios proposed. This facilitation is the exact goal of the webinos vision.

### 4.5.4 Cloud4All

#### *4.5.4.1 Project Overview*

Cloud4All (Cloud platforms Lead to Open and Universal access for people with Disabilities and for All) is a 4 year EU FP7 funded project; the project commenced in November 2011 and is funded until the end of August 2015. The Cloud4All consortium is composed of 24 partners and 3 collaborators across the EU and USA. The consortium is lead by Technosite, and technically co-ordinated by Raising the Floor - International. Like the webinos project, the consortium partners are drawn from research & development organisations, universities, and hardware manufacturers. Crucially, however, the consortium also contains several commercial assistive software companies.

The objective of the Cloud4All project is to carry out research and development towards a Global Public Inclusive Infrastructure (GPII). This infrastructure will make web and cloud platforms more accessible to people with special needs, such as those with disabilities or literacy and age-related difficulties. The GPII will incorporate 15 components situated around discovering features which are available, accessing preference information ubiquitously, and providing the infrastructure necessary for developers to build more accessible solutions and bring them to an international market place. The Cloud4All project will build several of these components. These will include personalisation components for capturing and storing preference information, accessible controls for supporting authentication, eliciting contextual information to support service adaptation, and the delivery of accessible web applications to different platforms.

#### *4.5.4.2 Cloud4All Scenarios and Use Cases relevant to webinos*

Because the Cloud4All project is still at an early stage, it is unclear what specific scenarios or use cases might be relevant for webinos.

#### *4.5.4.3 Possible alignment with webinos*

During a meeting held at Oxford between members of the webinos consortium and a representative of the Cloud4All project, the following two areas for collaboration were identified:

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Cloud4All is a useful source of user information about what, from a webinos perspective, is an under-represented user stakeholder: disabled users and people with accessibility problems. Consequently, the Cloud4All project has access to user communities upon which useful usability artifacts, e.g. scenarios and personas, can be created and used by webinos. These might form the basis of possible scenarios, or accessibility requirements for the webinos platform.

- Cloud4All's auto-personalisation needs appear to be closely aligned with webinos' own content adaptation requirements. Once Cloud4All's own requirements have been elicited, it would be useful to explore whether these are satisfied by webinos' own requirements in this area and, if not, what additional content-adaptation requirements might be useful for webinos.

### 4.5.5 Smart Cities

#### *4.5.5.1 Overview*

The modern city is turning into to a central hub of human life which depends on information and communication technologies (ICT). New technical solutions are being developed and perhaps most importantly, existing infrastructures can be utilized more efficiently. Therefore, ICT infrastructures of the smart city are logically connected to each other in order to exchange and use data about status, demands and capacities. Information is available wherever it is needed and the modern city thus makes itself transparent. The technology itself takes a backseat because the control concepts fit seamlessly into the everyday life and customs of the people. Energy is one of the important topics discussed in context of Smart cities: power supply system will be converted into an intelligent energy information system that will not only transport energy but also information about consumption and availability. In smart cities, a transparent data exchange for the consumer will support conscious power consumption. In the following scenarios we will describe two Smart Energy scenarios and we will also show how webinos can help by the development of such applications.

#### *4.5.5.2 Smart Cities Scenarios and Use Cases relevant to webinos*

This story is described from the perspective of a consumer who wants to manage their energy consumption, own their energy consumption data and use that data to negotiate with their energy suppliers and / or switch energy suppliers. Two scenarios under the topic Smart Energy are described below:

Scenario 1: John is conscious of energy savings and also the need to own his own energy consumption data. He knows of a third party supplier who runs a cloud based system which manages his energy data on his behalf. John signs up to the service (called WebEnergy). WebEnergy connects energy consumption from all devices in the home to a central server hosted in the cloud. The energy consumption is stored in the cloud. This consumption can be accessed by John and his family. The next time John switches providers, john is able to accurately indicate to the new provider his consumption and thereby get a better deal. The alternative is a ‘meter reading’ every six months and an approximation of energy consumption based on that reading. This does not benefit anyone. The WebEnergy service provides a benefit to John (Green awareness, better deal for negotiations, etc.), the provider (a better awareness for charging).

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

Scenario2: John and other members in his personal area (e.g. his family) are able to control the energy consumption of each connected device in real time using an application (e.g. SmartEnergyMonitor) running on his mobile phone or desktop computer. The application provides a UI where all connected devices and the energy consumption are listed. In a detailed view John can monitor the consumption of a specific connected device in real time and show the energy consumption in different time periods e.g. day, week, month etc. The application provides also a way to setup a limit for the energy consumption of a specific device (TV, etc.) or a group of devices (devices in the living room, etc.). Once the threshold of the energy consumption of the associated device or device group is exceeded, John will receive a notification immediately (beep alarm on his mobile phone, SMS, Mail, etc.). In addition to the monitoring of the energy consumption, the mobile application can be used to monitor other measurement values such as temperature, humidity, etc. Regarding to privacy issues, the application will keep all these sensor values available only for devices in the same personal zone. The energy provider of John offers also a new payment option called "PrePaidEnergy", which allows customers to pay in advance for the energy consumption. This option is possible if only the energy provider can monitor the total energy consumption in real time. John wants to use this option therefore he uses a service running in his personal cloud which collects the energy consumption of all connected devices to calculate the total energy consumption. The Energy provider has only access to the new exposed service and read only the total energy consumption.

In addition to the monitoring of the energy consumption and other measurement values, John is able to control actuators (switch light ON/OFF, etc.) directly from his webinos enabled mobile application. He can also define rules to control actuators based on different context parameters such as sensors, personal context, etc.

#### *4.5.5.3 Possible alignment with webinos*

In both scenarios it is required to monitor the energy consumption and other measurement values of different home devices. This is possible through Webinos: supposing all sensors are connected to a hardware board where a webinos (micro)-PZP is running. On the mobile device is also a PZP running. The webinos application running on the same device can use the discovery API offered by webinos to search for all sensor services available in the personal zone. For each sensor the application can use the Sensor API provided by webinos to subscribe to each specific sensor and receives values in real time. The same applies for controlling actuators by using the actuator API. Summarized, the following webinos features are relevant to scenarios described above:

- Service Discovery: Discover Sensors/Actuators in the personal zone using the webinos discovery API.
- Monitor Sensor Values: Make use of the webinos Generic Sensor API to subscribe to specific sensors and receives new reported values in real time.
- Control Actuators: Make use of the webinos Actuator API to change the state (e.g. turn ON/OFF) of a specific Actuator (e.g. switch).
- Privacy/Security: webinos core components for security and privacy.
- Total Energy Consumption: expose new service for providing the total energy consumption using the webinos service exposer

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Make total energy available for energy provider: webinos privacy module provides a way to make specific services in a personal zone available in another one.

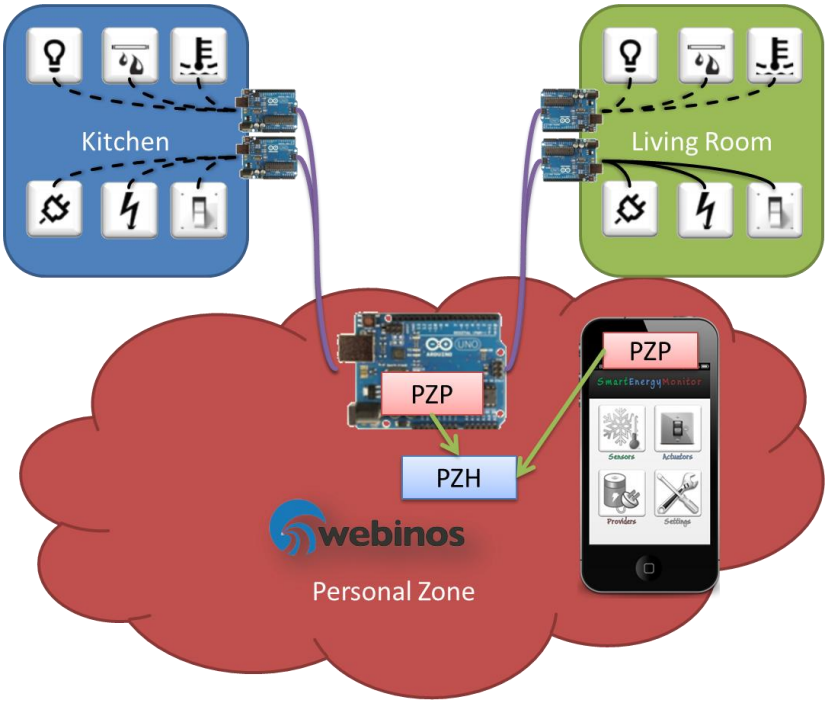



**Figure 2 Smart Energy + webinos**

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 5 Use Cases

In this chapter the updated list of use cases for the webinos platform are listed. The list is based on updates of the first set of use cases as described in D2.1 Scenarios and Use Cases. Because new feature requests from both within and external to the consortia can arrive at any time, the list of use cases as well as the use case descriptions are liable to update. The latest version can be found in the webinos WP2 git repository and the WP 2 Redmine wiki.

Before listing the revised use cases, we describe the approach taken for updating the use cases, and summarise the changes made.

## 5.1 Review Process

The use case review and update process was designed to satisfy three objectives:

- Check whether use cases are still relevant for webinos.
- Revise use cases based on our current understanding of the webinos architecture.
- Explain, in as much detail as feasible, how use cases are implemented in architectural terms.

The process involved carrying out two steps for each use case. In the first step, the use case was inspected to determine its continual relevance for webinos. Assuming the use case was still within the scope then two concurrent activities were carried out in the next step: the use case was revised to better reflect out improved understanding of the webinos architecture, and -- while revising the use case -- review comments were added to justify the use case revisions made.

These activities are described in more detail in the sections below. Use case reviews were carried out by cross-partner working groups. The table below shows the assigned responsibilities according to the original categories defined in D2.1. These were later regrouped to better reflect the dependencies between scenarios, use cases, and requirements. Following each review, a cross-review was carried out was, again, carried out by in cross-partner working groups.

| **Category** | **Responsibility** | **Cross-Review** |
|---|---|---|
| Virtual Device | FOKUS | OXFORD |
| User Management | TUM | IBBT |
| Application Execution Environment | IBBT | BMW |
| Communication | FOKUS | OXFORD |
| Application Life-cycle Management | DTAG | BMW |
| Content Adaptation | NTUA | IBBT |
| Context Awareness | IBBT | POLITO |

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

| | | |
|---|---|---|
| Policies | TUM | POLITO |

### 5.1.1 Activity 1: Check use case relevance

To establish the relevance of each use case, existing webinos deliverables were reviewed in order to answer a number of questions. Once completed, each use case review was uploaded to the webinos redmine repository.

- How does this use case relate to other use cases?
- Does this use case operationalise known requirements?
- Is this use case explicitly described in the webinos architecture?
- Is this use case implicitly partially or completely implemented by one or more backlog items in WP 4 (webinos implementation)?

For a use case to be relevant and, therefore, amenable to revision as part of D2.4, the answers to all four questions did not need to be yes. However, after establishing answers to each question, a decision was proposed about how to proceed with the use case. The available options were:

| **Option** | **Criteria** |
|---|---|
| *Reject* | The use case describes behaviour which is either no longer within webinos' scope, or is infeasible given the project's time and resource constraints. |
| *Defer* | The use case describes behaviour which might be carried out by webinos, but this behaviour not within the scope of the webinos runtime. Where appropriate, this use case should be assigned to the WP 5.2 or WP 6 team to ensure this use case is addressed by webinos applications. If the WP 5.2 and 6 teams do not believe the use case relevant then the use case should be rejected. |
| *Accept* | The use case describes behaviour explicitly or implicitly stated within the D5.1, D.5.2 and/or the backlog items in WP 4. |

If the use case is not graded as Accept, then the review & update process for the use case stops here. In total, 77 from 111 use cases were rejected or deferred. Thus, 34 were kept (all numbers including alternate flows of use cases).

### 5.1.2 Activity 2: Model use case map and revise the use case

One of the most effective ways of validating the effectiveness of the use case is to try realising it in some way. Taking a step back and contextualising how this might be realised by personas in different scenarios is one way of doing this. Another way is looking at the steps in more detail and precisely describing what parts of webinos are responsible for different aspects of the use case. By reflecting on who or what parts of webinos are responsible for different parts of a use case, it is possibly to identify ambiguities, inconsistencies, or omissions that may not have been apparent from a cursory reading of the use case. Analysis at this level may also identify missing steps or even missing use cases.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

To facilitate this analysis, **use case maps** (BUHR1996) were created for each use case. Use case maps consist of a number of boxes that represent actors or components in the webinos architecture. The model shows the causal path taken when carrying out the different use case steps. Each step is signified by a cross, which is placed depending on who or what component is responsible for it. Preparing the use case map, as well as revising the use case in general, involves inspecting the architectural deliverables to see who is responsible for different parts of the use case. Care was taken to ensure that the use case and the use case map remained at an appropriate level of abstraction.

During the preparation of the use case maps, the use case steps were also updated based on insights that were gleaned during the modelling process.

### 5.1.3 Activity 3: Review and update the use case

In the majority of cases, each use case reviewer was not the original D2.1 use case author. However, given time constraints, whoever reviewed accepted use case was also responsible for revising it for D2.4. To ensure that subjectivity did not creep into the update process, revisions to the use case were accompanied with a review comment that justified the revision being made. The following considerations were given to reviewers for carrying out the individual elements of each use case:

| **Component** | **Considerations** |
|---|---|
| Description | Does this summarise the purpose of the use case, rather than describe the general context? The latter is more appropriate for scenarios rather than use cases. |
| Preconditions | Many of the current use cases are described within a larger context. Does this section describe everything that is expected to hold before the use case can be carried out as described? Is everything that is described applicable described at an appropriate level of relevance for this use case? |
| Flow | Are there any terms described in the use case steps ambiguous and open to misinterpretation? Is this use case defined at a sufficient level for someone to go away and implement all or part of this use case as a backlog item? Does this use case describe or reference components or solution constraints that are not relevant given this use case's level of abstraction? |
| Postconditions | Does this accurately describe the state of webinos or the related webinos-enabled application once the use case flow has successfully run its course? |

### 5.1.4 Activity 4: Add review and update use case status

The final stage involved collecting all review steps together including the creation of a review page with comments that is linked with the general list of use cases, creating digital versions of the use case maps, and updating the status of the use case to be ACCEPTED, REJECTED, or DEFERRED.

As previously stated, after the first round of reviews, a cross review was carried out to find agreement between the involved partners. The final decisions on rejected and deferred use cases were made during a WP2 workshop in Ghent in March 7 -8 2012 to ensure that use cases were not removed without a group consensus being reached.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

## 5.2 Summary of Changes

All use cases that were accepted were re-grouped during categories different from those used for D2.1. The original use case families like "Virtual Device" or "Application Life-cycle Management" were replaced with the classifications that were originally used for the requirements functional areas in D2.2. Namely "Device and Service Functional Capability" (CAP), "Discovery and Addressing" (DA), "Identity" (ID), "Lifecycle" (LC), "Negotiation and Compatibility" (NC), "Policy and Security" (PS), "Remote Notifications and Messaging" (NM), and "Transfer and Management of State" (TMS). This change was made because, having reviewed the use cases that were accepted, this seemed a more appropriate categorisation scheme given our better understanding of the webinos platform's scope. This change also helped provide a common categorical scheme across scenarios, use cases, and requirements, which improves readability and traceability across all artifacts.

Of the use cases that were removed, the majority were found to be too application specific. Many of these use cases could be realised by implementing applications using specified webinos functionality, rather than changing the webinos platform to accommodate the applications. Most of them, 22, were already classified as *example* use cases during the first phase of the project. Based on our revision of the scenarios, we concluded that additional example use cases were unnecessary. An additional 22 use cases that originally were not part of the examples sections were also removed for the same reasons. For example "WOS-UC-TA4-004 Resume Uploading of Data" was concerned with resuming uploading of data after a lost connection was re-established. Having discuss this and other similar use cases, this was not considered to be behaviour that was exclusive to the webinos platform. Consequently, if an application could exhibit this behaviour by using several system events (such as network status) and APIs (such as File APIs) provided by webinos then these fulfilled the same role as scenarios rather than platform-specific use cases.

Other use cases were removed because they either duplicated other use cases, or other similar use cases could be trivially adapted to cover the needs described in the use case. For example the use case "Communication between webinos Applications" (original identifier WOS-UC-TA1-004) described the same features needed as described in "NM1: Subscribe and Publish Events" (original identifier WOS-UC-TA4-001). Both described application to application communication across devices, which meant there was little need for maintaining both use cases.

The final group of removed use cases were those which were out of scope. Because our understanding of webinos' scope is clearer than it was at the start of the project, several use cases describe behaviour which is no longer appropriate

This just means that during the project life it was becoming clear what webinos is and how it works which basically was not that clear in the beginning of the project. So several use cases can now be considered as beeing out of scope. For example the use case "WOS-UC-TA1-006 Pairing a new device to clone settings" was talking about setting up a new device with webinos just by turning it on and automatically discover other nearby webnios devices. Now, with the introduction of the webinos personal zone concepts, it is defined how devices can be attached to a user. For this it is seen sufficiently to describe the need for attaching devices to user's as, for example, covered in use case "DA3: Virtual Device Ownership" (original identifier WOS-UC-Ta1-003).

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

## 5.3 Use Case Description Template

Each use case contains a title and a unique identifier. This identifier is based on the functional area and an integer uniquely identifying the use case within its functional area.

Use cases also contain the following attributes:

**Author**: The author of the original use case

**Actors**: The stakeholders involved in the use case

**Description**: A short narrative description the scope or context of the use case. The description also contains a reference to the original use case where appropriate.

**Preconditions**: The preconditions that must be fulfilled before the use case can be applied.

**Flow**: The step by step flow of the use case.

**Postconditions**: The postconditions that are true after the use case was applied.

**Use Case Map**: A use case map diagram illustrating the components or actors responsible for each use case flow step.

**Related Scenarios**: A list of scenarios from where this use case can be derived.

**Related Requirements**: A list of requirements names which the use case is either derived from, or operationalised by.

## 5.4 List of Use Cases

### 5.4.1 CAP1: Sensors and Actuators

Author: Andre Paul

Actors: Application, User

#### *5.4.1.1 Description*

The use case describes how to access data from sensors or use actuators which may or may not webinos enabled. Different sensor types are possible, temperature sensor, light switch, proximity sensor and integration of external web services which providing sensor like data are only some examples. Based on D2.1 use case WOS-UC-TA1-005.

#### *5.4.1.2 Preconditions*

The user and its application is allowed to access the sensors and actuators in scope. Sensor or actuator specific security implications are not in scope of the use case. Non webinos enabled sensors or actuators are bound to a PZP which acts as mediator to access the services.

#### *5.4.1.3 Flow*

1. The User starts an application that needs to have access to certain sensors or actuators. For example temperature, heart rate, light, heater etc.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

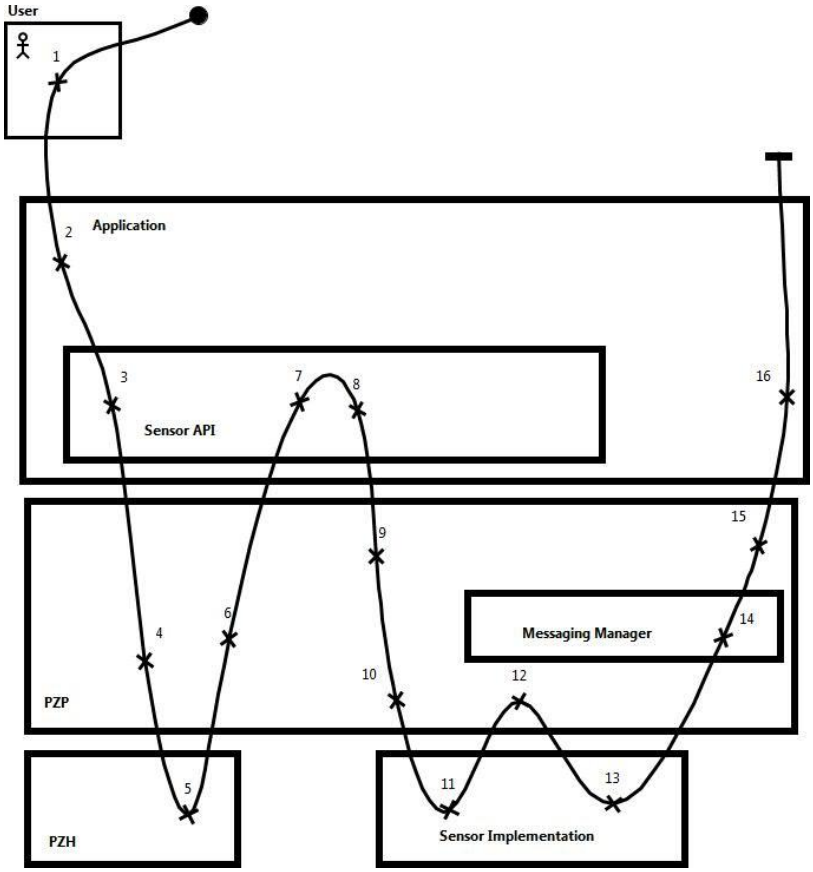
2. Therefore, the Application asks the webinos runtimes discovery mechanism to get access to certain sensor or actuator types.
3. The webinos runtime forwards the request to the PZP.
4. The PZP executes the query and asks the PZH for known sensor and actuators that matching the requested type.
5. The PZH provides information about available sensor and actuator services back to the PZP.
6. The PZP provides the service information to the application.
7. The application selects a sensor or actuator service to use and binds to the service.
8. The application invokes the sensor or actuator using related sensor or actuator API calls.
9. The PZP processes the invocation requests and forwards the request to the device where the service is located (which may involve forwarding through the PZH).
10. The PZP where the service is bound to receives the invocation request.
11. In case the service is directly available on the same device as the PZP the service is executed and step 15 follows.
12. In case the sensor of actuator is mediated through the PZP the PZP translates the invocation request into sensor or actuator specific commands.
13. The service provides results back to the mediating PZP
14. The PZP translates the results into webinos specific messages.
15. The PZP sends the results of the service invocation back to the requesting application.
16. The application makes use of the provided information.

#### *5.4.1.4 Postconditions*

A sensor or actuator that is not webinos enabled can be accessed and the application can interact with sensors and actuators.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.1.5 Use Case Map*




#### *5.4.1.6 Related Scenarios*

- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files
- S-PS1: Usable Privacy Controls for webinos
- S-NC3: Background Monitoring Services

#### *5.4.1.7 Related Requirements*

- Description-based discovery
- Service availability detection
- Un-supported sensor/actuator discovery
- non-webinos device API
- Remote device access
- webinos enabled sensor access

### 5.4.2 CAP2: User Profile

Author: Andre Paul

Actors: Application, User

#### *5.4.2.1 Description*

An application wants to access a user's profile. This comprises information such as personal data such as gender, age, home address and e-Mail address. Based on D2.1 use case WOS-UC-TA2-001.

#### *5.4.2.2 Preconditions*

The user has entered his profile info which is stored in his Personal Zone.

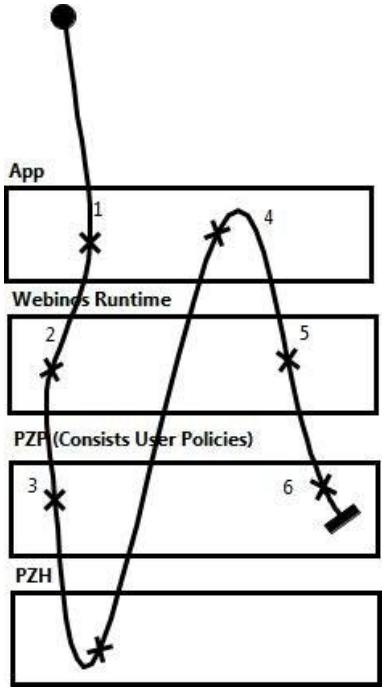
#### *5.4.2.3 Flow*

1. An Application requires information about the User, it requests the webinos runtime to grant access to this information.
2. The Webinos Runtime checks the PZP if access should be granted.
3. The PZP checks the users policies if the application should be given access and returns this information to the application
4. The application requests the needed profile information to the Webinos Runtime.
5. The Webinos Runtime requests this information to the PZP.
6. The PZP returns the info to the application.

#### *5.4.2.4 Postconditions*

Profile data related to the User is available to the requesting application.

#### *5.4.2.5 Use Case Map*




#### *5.4.2.6 Related Scenarios*

- S-PS3: Privacy Controls and Analytics for Corporations and Small Business
- S-LC1: Designing Policy-aware webinos Applications

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- S-NC2: Learning Assistance for Disabled Students

#### *5.4.2.7 Related Requirements*

- User identity
- User profile access

### 5.4.3 CAP3: Background Execution

Author: Andre Paul

Actors: User

#### *5.4.3.1 Description*

Applications get the possibility to run code in the background that keeps running even when the main app is closed Based on D2.1 use case WOS-UC-TA3-005.

#### *5.4.3.2 Preconditions*

The Application that wants to execute something in background is running.

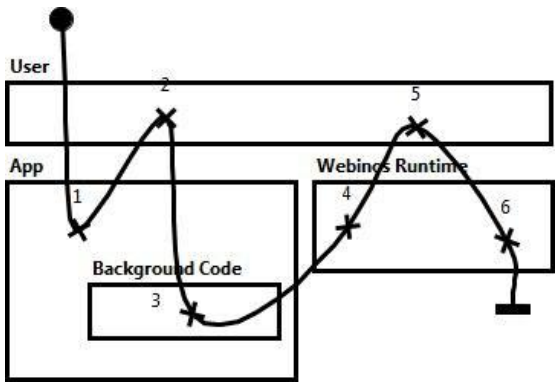
#### *5.4.3.3 Flow*

1. The Application assigns code that should be executed in the background to the System (e.g. listing to temperature sensors).
2. The main Application is closed by the User.
3. The background code notifies the Webinos Runtime that the main Application should be launched (e.g., because it is over 30 degrees now).
4. The Webinos Runtime informs the User about the occurrence of a certain event that triggers launching an Application.
5. The User decides whether he want to launch the Application or not (do nothing).
6. If the User chose to do so, the Application is launched

#### *5.4.3.4 Postconditions*

Code is executed in the background without the need for a running main Application. For the User the background execution is not visible.

#### *5.4.3.5 Use Case Map*



#### *5.4.3.6 Related Scenarios*

- S-NC3: Background Monitoring Services

#### *5.4.3.7 Related Requirements*

- Application event
- Application launch notification
- Event type subscription
- Analytical change correlation
- Background application
- Background event notification
- Event-based application start-up
- Foreground priority
- Foreground request
- Non-GUI background applications

### 5.4.4 CAP4: Background Execution (alternative)

Author: Andre Paul

Actors: User

#### *5.4.4.1 Description*

Applications can register code to run automatically in the background at startup or at install time. Based on D2.1 use case WOS-UC-TA3-005_1.

#### *5.4.4.2 Preconditions*

The Application that wants to execute something in background is running or is being installed.

#### *5.4.4.3 Flow*

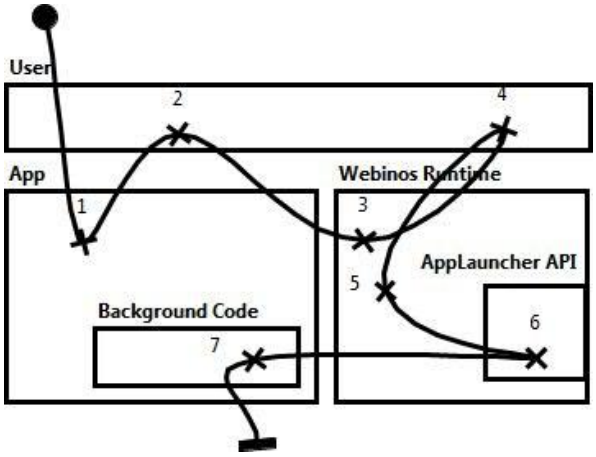
1. The Application registers for background execution on System start-up.
2. The User turns off the Device
3. The Webinos Runtime shuts down
4. The User turns on the Device
5. The Webinos Runtime starts up
6. After the Webinos runtime boot sequence is completed, It starts up the background code from the application
7. The background code starts up

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.4.4 Postconditions*

Code is executed in the background without the need for a running main Application. For the User the background execution is not visible.

#### *5.4.4.5 Use Case Map*




#### *5.4.4.6 Related Scenarios*

- None

#### *5.4.4.7 Related Requirements*

- Subscription
- webinos event
- Automatic background start-up
- Background application
- Non-GUI background applications

### 5.4.5 CAP5: Wake-on-Webinos

Author: Hans Stokking

Actors: Application, User

#### *5.4.5.1 Description*

This usecase is about waking-up other device upon demand, to enable devices to have an energy-saving sleeping mode and still be usable for services. Based on D2.1 use case WOS-UC-TA4-017.

#### *5.4.5.2 Preconditions*

A webinos enabled device is registered as one of an end users devices. Some services that can be used remotely by applications are located on this device.

#### *5.4.5.3 Flow*

1. The user starts the webinos runtime (note, the webinos runtime could be also automatically started, e.g., after the device was successfully started)

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

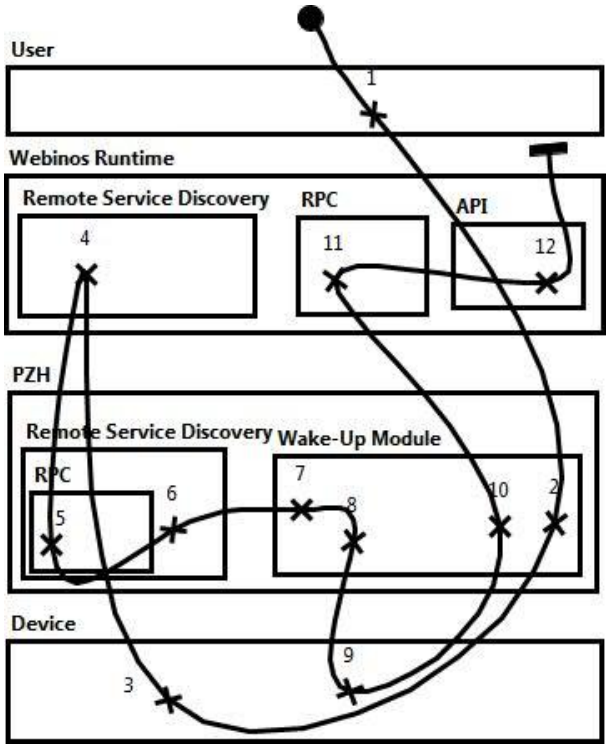
2. The webinos runtime registers itself at a wake-up component as being available and exposes its services to the discovery components.
3. The device goes in idle mode and after a while in sleeping mode, for example a low-energy mode in which it can still be activated over the network (this depends on the device and the native operating system).
4. An application running on another device wants to make use of a service located on the device with the use of service discovery.
5. The RPC mechanism receives this service request.
6. The service discovery determines that this request should be routed to the device.
7. A wake-up component looks up the device status, and finds out that the current status is sleeping / not available.
8. The wake-up component sends a wake-up message via the network to the device (which, for example, can directly be send to the device or through other devices running in the same network as the target device).
9. The device receives the request and activates.
10. The webinos runtime activates, and sends an acknowledgement to state that it is available for service requests.
11. The RPC mechanism forwards the original service request to the target device
12. The remote service request is executed using the desired API.

#### *5.4.5.4 Postconditions*

The device is in active mode and available for service requests. If the device cannot be woken up for some reason, the webinos system should proceed as if the device was unavailable in the first place.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.5.5 Use Case Map*




#### *5.4.5.6 Related Scenarios*

- S-NC2: Learning Assistance for Disabled Students

#### *5.4.5.7 Related Requirements*

- Event type subscription
- webinos event
- Background event notification
- Device Status API
- Event-based application start-up
- Foreground priority
- Remote device access
- Wake-up message subscription

### 5.4.6 DA1: Virtual Device

Author: Andre Paul

Actors: Application, User

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.6.1 Description*

webinos allows the applications to use device features available on the user's currently used device or on other remotely available devices (especially mobile, PC, automotive, and home media devices). Based on D2.1 use case WOS-UC-TA1-001.

#### *5.4.6.2 Preconditions*

The user has at two webinos enabled devices, which are known to each other. Both devices are running and are connected so that communication is possible and using services from each other is allowed and permitted.

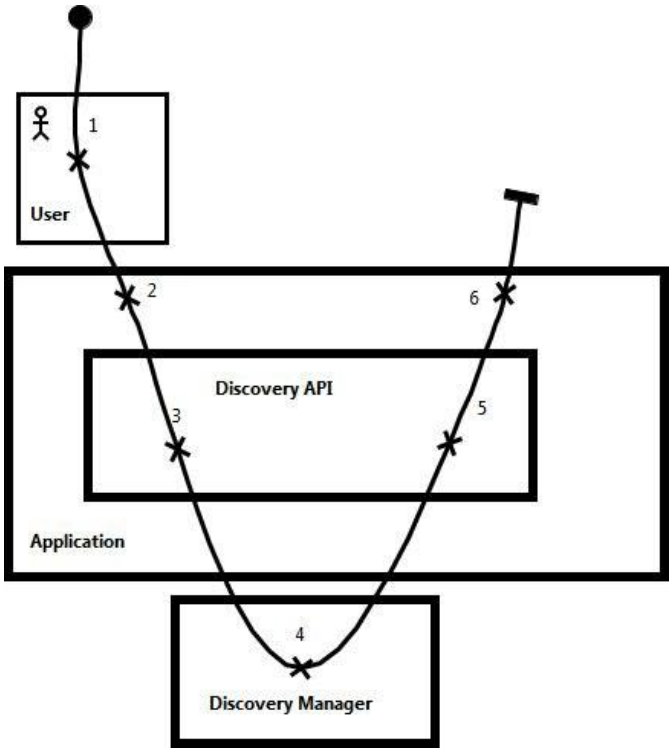
#### *5.4.6.3 Flow*

1. User starts an application on one of his devices.
2. Application needs to use a camera service which is not accessible using local APIs.
3. Application uses a service discovery mechanism to find all camera services available to the user.
4. webinos creates a set of services available to the application and user which meet the discovery criteria.
5. Application receives a number of found services that can be used.
6. Application accesses a discovered service to carry out his desired task.

#### *5.4.6.4 Postconditions*

A camera service available on a remote device can be used by the application.

#### *5.4.6.5 Use Case Map*




This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.6.6 Related Scenarios*

- S-CAP2: Report Disorder
- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files
- S-DA1: Smart Device Integration

#### *5.4.6.7 Related Requirements*

- Feature access
- Service identity
- Capability availability broadcast
- Network availability broadcast
- Service discovery
- Un-supported sensor/actuator discovery

### 5.4.7 DA2: Network Independent Virtual Device

Author: Andre Paul

Actors: Application, User

#### *5.4.7.1 Description*

webinos allows applications to use device features available on a user's currently used device or on other remotely available devices (especially mobile, PC, automotive, and home media devices) no matter where the device is and how it is physically accessible (e.g., network that is used, physical location, GSM vs. WiFi etc.). Based on D2.1 use case WOS-UC-TA1-002.

#### *5.4.7.2 Preconditions*

The user has at least two webinos enabled devices, which are known to each other. Both devices are running and are connected so that communication is possible and using services from each other is allowed and permitted. The two devices used are running in completely separated network areas (e.g., home network vs. GSM network or home network A vs. home network B).

#### *5.4.7.3 Flow*

1. User starts an application on one of his devices.
2. The application needs to use a camera service which is not accessible using local APIs.
3. The application uses a service discovery mechanism provided by the webinos runtime to find all camera services available to the user.
4. webinos discovers devices that are located in the same network
5. webinos discovers devices that are located in separated networks

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

6. webinos creates a set of services available to the application and user which meat the discovery criteria while the set includes cameras available within different networks.

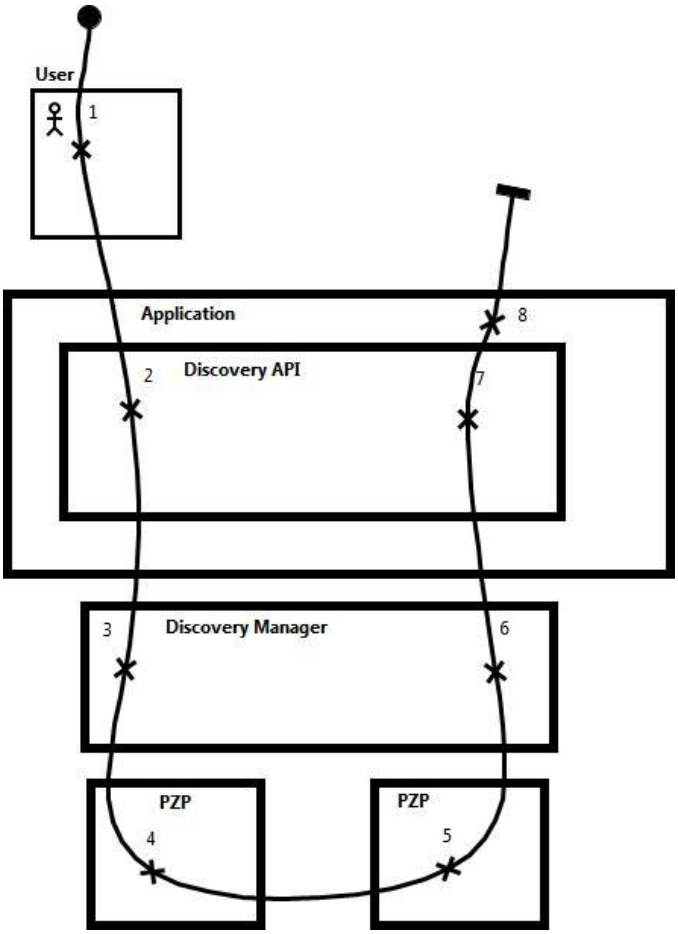
7. Application receives a reference to a discovered camera services that can be used.

8. Application accesses a discovered camera service to carry out his desired task.

#### *5.4.7.4 Postconditions*

A service available on a device in a different personal zone can be used by the application.

#### *5.4.7.5 Use Case Map*




#### *5.4.7.6 Related Scenarios*

- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files

- S-NC3: Background Monitoring Services

#### *5.4.7.7 Related Requirements*

- Feature access

- Service identity

- Capability availability broadcast

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Network availability broadcast
- Personal device management
- Service discovery
- Un-supported sensor/actuator discovery

### 5.4.8 DA3: Virtual Device Ownership

Author: Andre Paul

Actors: Application, User

#### *5.4.8.1 Description*

webinos not only allow to use services from remote devices if the involved devices are belonging to the same user, it also allows access services from devices belonging to a different user. Based on D2.1 use case WOS-UC-Ta1-003.

#### *5.4.8.2 Preconditions*

Both user, user A and user B, representing two different webinos accounts, thus, belonging to two different personal zones. Both users have webinos enabled devices which are provide access to certain services. Accessing and using remote resources of a different user is not possible. User B knows user A's identity. User A is running an application requiring access to User B's media.

#### *5.4.8.3 Flow*

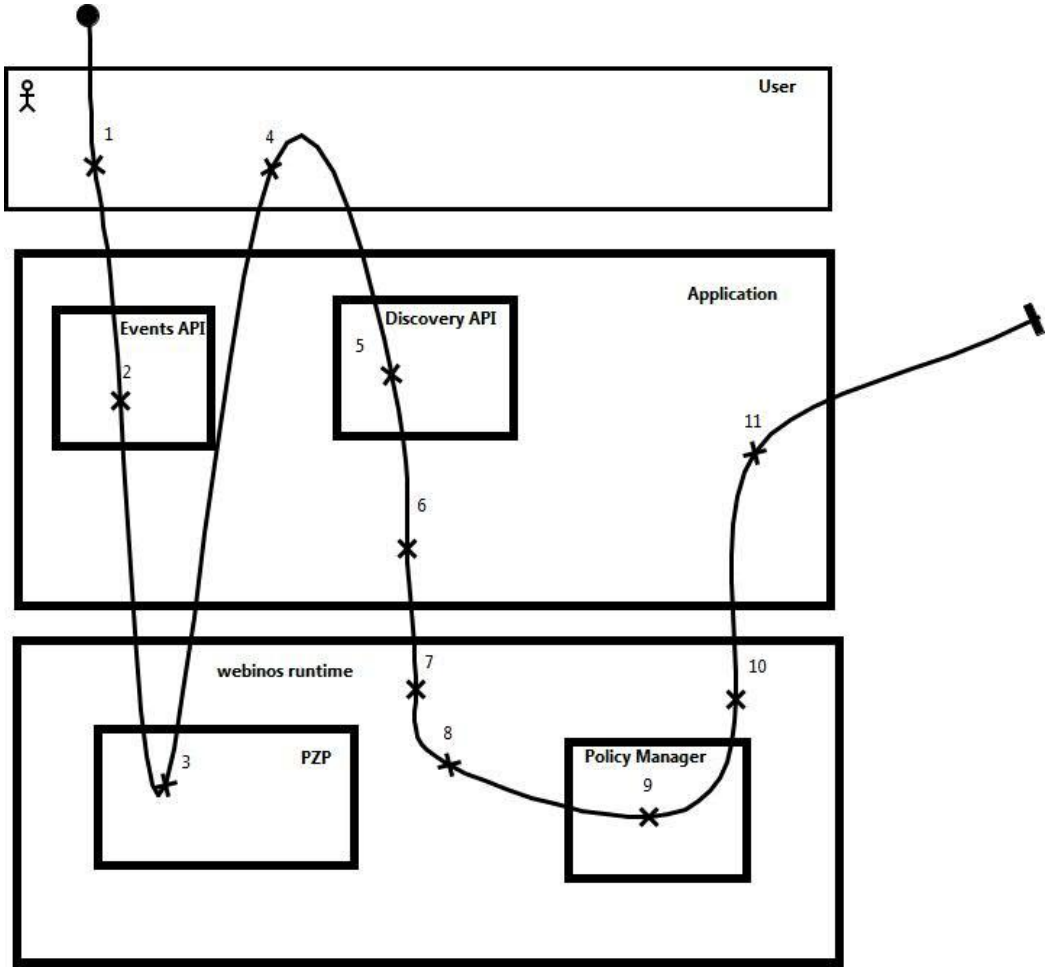
1. User A starts an application that needs to use a certain service provided by a device from user B (e.g., access to the media folder or using the camera to take a picture).
2. The webinos runtime for user A sends a request to user B's webinos runtime.
3. The webinos runtime of user B notifies user B.
4. User B accepts the invitation of user A.
5. Application uses webinos service discovery to find user B's media.
6. Application invokes the related service API.
7. The webinos runtime for User A sends an invocation requests to the device of user B that hosts the required media.
8. The webinos runtime of the target devices determines whether it is allowed to execute the service.
9. User B's webinos runtime authorizes access to the application.
10. User B's webinos runtime satisfies the media access request,
11. Application uses the media resource.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.8.4 Postconditions*

User A's application has access to authorised media files in User B's personal zone.

#### *5.4.8.5 Use Case Map*




#### *5.4.8.6 Related Scenarios*

- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files

#### *5.4.8.7 Related Requirements*

- Shareable services
- User association alert
- WAC/XACML policy
- Capability availability broadcast
- Personal device management
- Service discovery

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

### 5.4.9 DA4: Discovering the application

Author: Katrin Jordan

Actors: Application, User

#### *5.4.9.1 Description*

The user becomes aware of a new application. Based on D2.1 use case WOS-UC-TA5-001.

#### *5.4.9.2 Preconditions*

The application has been developed and is accessible on device(s) and/or the network. Devices are registered, i.e. are identified, authenticated and authorised

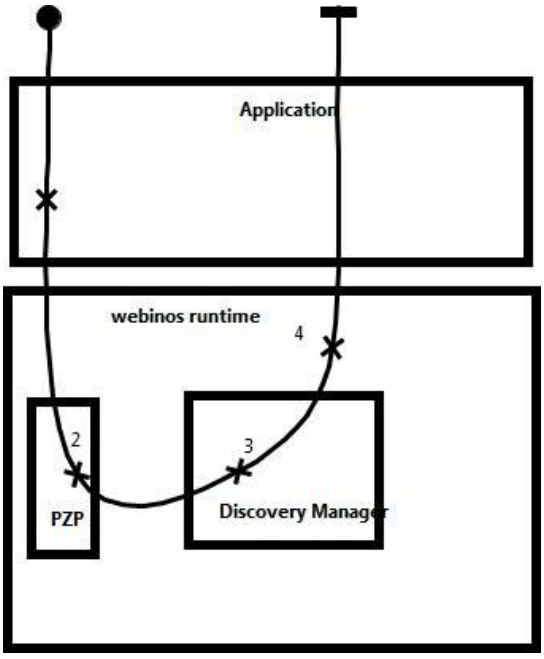
#### *5.4.9.3 Flow*

1. Application scans the available networks for applications based on certain API criteria.
2. webinos discovers a personal zone with a running application matching this criteria.
3. webinos carries out a device check is done to ensure availability and compatibility of the required features.
4. webinos notifies the user of successful detection, with an identifier of the application and its destination.

#### *5.4.9.4 Postconditions*

The application is available for further use, subject to authorisation.

#### *5.4.9.5 Use Case Map*




#### *5.4.9.6 Related Scenarios*

- S-DA2: Input Method Dictionary Sharing across Devices
- S-PS1: Usable Privacy Controls for webinos

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- S-DA3: Converging Applications within and across Devices

#### 5.4.9.7 Related Requirements

- Application maintenance
- Application-based discovery
- Personal device management
- Service discovery

### 5.4.10 DA5: Finding devices in close physical and social proximity

Author: Heiko Desruelle

Actors: Application, User

#### 5.4.10.1 Description

Applications should be able to obtain a filtered list of entities based on their physical and social proximity. Physical proximity is based on the entities location and expressed in terms of a certain physical range. Social proximity on the other hand, focuses on peoples social connections rather than the location-based aspect of physical proximity. Based on D2.1 use case WOS-UC-TA7-004.

#### 5.4.10.2 Preconditions

The user's personal zone has a list of acceptable devices, according to his preferences.

#### 5.4.10.3 Flow

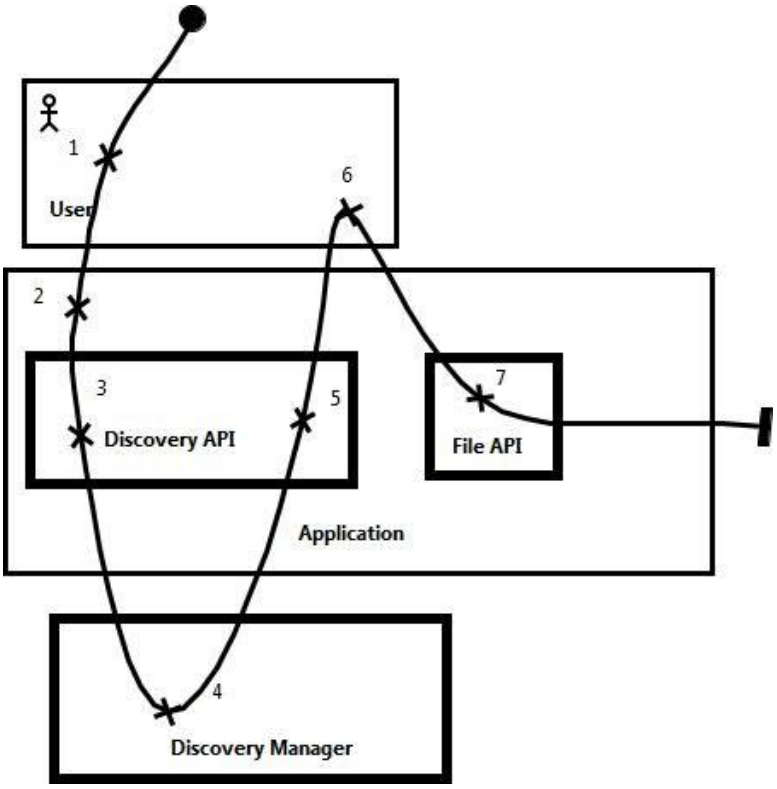
1. User uses an application and selects an option to access media files from another device.
2. Application queries the webinos runtime for an overview of the user's devices with the required capabilities within a specified, preferential physical range.
3. The webinos runtime requests the list of devices.
4. The personal zone hub executes the query and returns the resulting list of devices.
5. Application presents the results ordered by proximity, availability and capabilities.
6. The User selects a device.
7. Application instructs the webinos runtime to access media files from the discovered device.

#### 5.4.10.4 Postconditions

The identified media is available on the requesting user's device.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.10.5 Use Case Map*




#### *5.4.10.6 Related Scenarios*

- S-NC2: Learning Assistance for Disabled Students

#### *5.4.10.7 Related Requirements*

- Analytical information query
- Analytical information schema
- Capability availability broadcast
- Location-based discovery
- Personal device management
- Service availability detection
- Service discovery
- User social proximity

### 5.4.11 DA6: Finding devices in close physical and social proximity (alternative)

Author: Heiko Desruelle

Actors: Application, User

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.11.1 Description*

Applications should be able to get a filtered list of entities based on their physical and social proximity. Physical proximity is based on the entities location and expressed in terms of a certain physical range. Social proximity on the other hand, focuses on peoples social connections rather than the location-based aspect of physical proximity. Based on D2.1 use case WOS-UC-TA7-004_1.

#### *5.4.11.2 Preconditions*

The User is visiting another user. Both users have webinos enabled devices. The user has a list of acceptable devices, according to the personal physical and social proximity preferences.

#### *5.4.11.3 Flow*

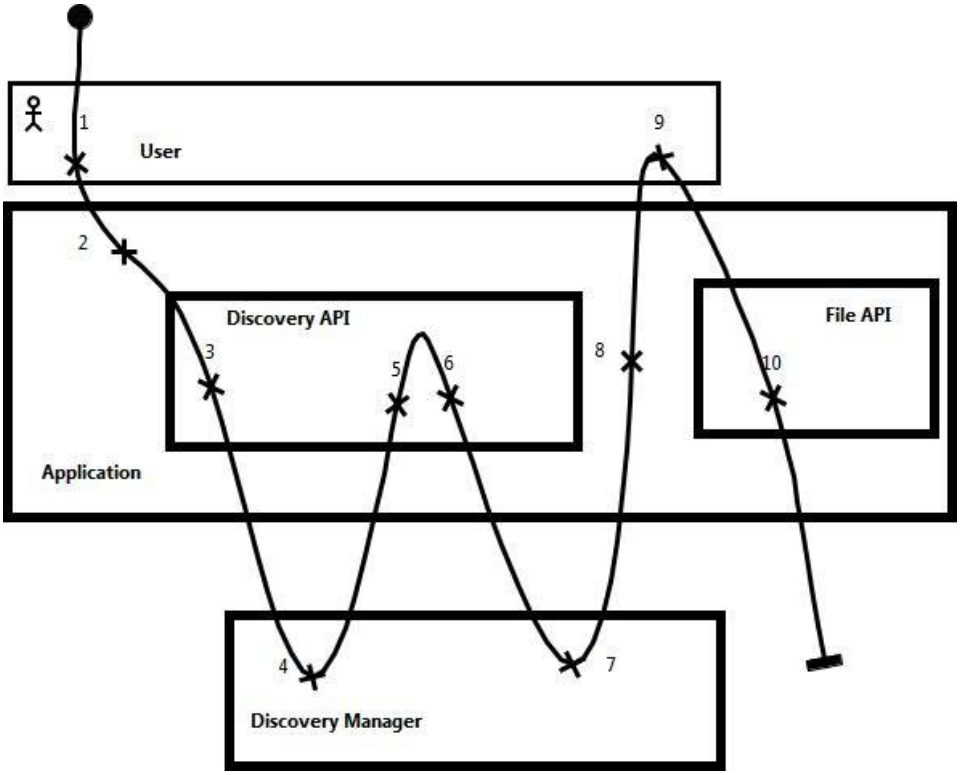
1. User uses an application and selects an option to obtain a media file from another device
2. Application queries the webinos runtime for an overview of authorised devices with the required capabilities within a certain physical range.
3. The webinos runtime requests the list of devices from the personal zone hub.
4. webinos executes the query and returns the resulting list of devices.
5. Application queries webinos for devices within both a preferential physical and social proximity.
6. The webinos runtime requests the list of devices from the personal zone hub.
7. webinos executes the query and returns the resulting list of devices.
8. Application presents the results ordered by social/physical proximity, availability, capabilities and associated users.
9. The User selects a device.
10. The Application instructs the webinos to obtain the media file.

#### *5.4.11.4 Postconditions*

The media file is obtained from the device selected within the preferred social proximity.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.11.5 Use Case Map*




#### *5.4.11.6 Related Scenarios*

- S-DA3: Converging Applications within and across Devices

#### *5.4.11.7 Related Requirements*

- Analytical information query
- Analytical information schema
- Capability availability broadcast
- Location-based discovery
- Service availability detection
- Service discovery
- User social proximity

### 5.4.12 LC1: Installation and update of webinos applications

Author: Katrin Jordan

Actors: User

#### *5.4.12.1 Description*

The user downloads, installs and renders applications and settings from the network (an application store) or another device. The application functionality will be available in offline mode, Data will be

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

synchronised once online connection is restored and based on user/app settings. Based on D2.1 use case LC1.

#### *5.4.12.2 Preconditions*

The application has been developed and is available online (e.g. in an app store). The webinos device has not yet installed an application, either already downloaded or still to be downloaded. The user will give authorisation to install the application by the webinos policy enforcement point. The application certificate is valid (the keys used are the same as the keys stated, the expiry date has not passed, and none of the keys have been revoked, and the digest matches the application binary)

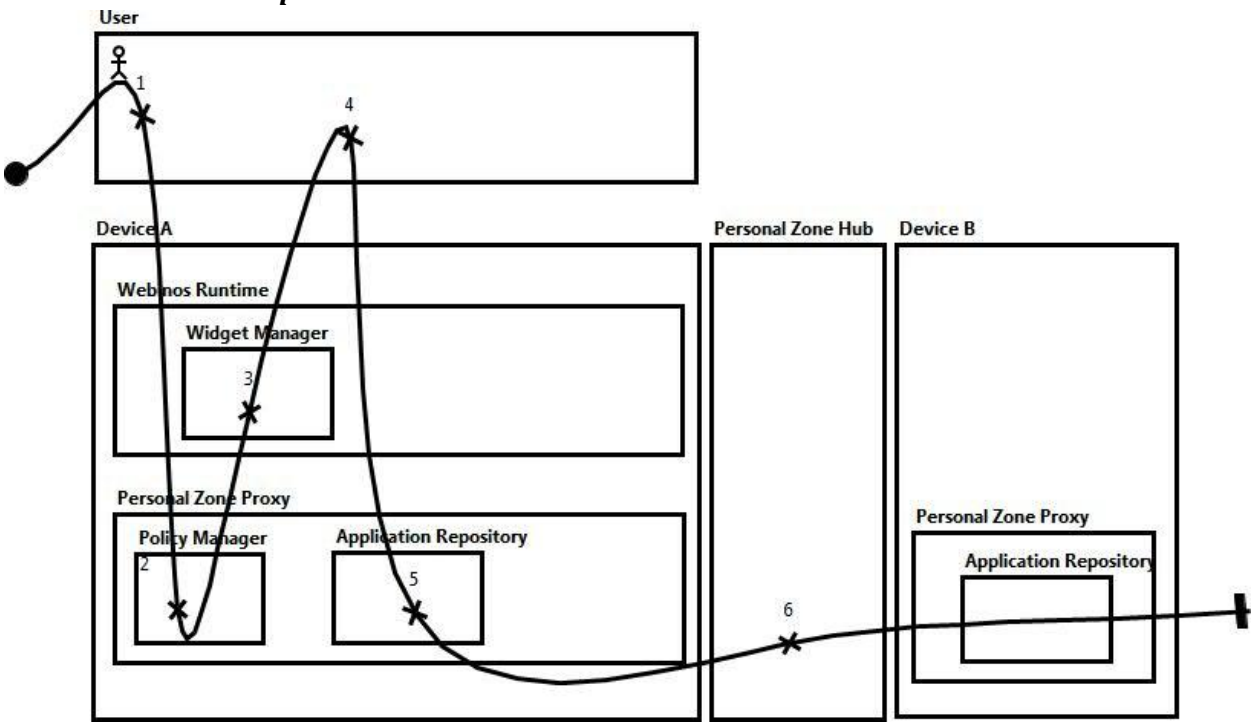
#### *5.4.12.3 Flow*

1. The end user attempts to install an application
2. Webinos checks the local policy to see whether any restrictions are in place.
3. The user is presented with information about the application, including the company (or person) who created it, what it claims to do, which features of the device it will use, which content it will have access to, and any additional certificates that are available. If there are privacy-sensitive implications to the application, these are presented as part of this process.
4. The user is asked whether the application should be made available to all their devices.
5. The user selects yes and installs the application to his devices.

#### *5.4.12.4 Postconditions*

The application is installed to the device and available for use on all instances of webinos that the user owns.

#### *5.4.12.5 Use Case Map*



#### *5.4.12.6 Related Scenarios*

- S-LC1: Designing Policy-aware webinos Applications
- S-CAP2: Report Disorder

#### *5.4.12.7 Related Requirements*

- Application data policy
- Application intent
- Application maintenance
- Contact details
- Human-readable version
- Multi-device install

### 5.4.13 LC2: Update of applications

Author: Katrin Jordan

Actors: User

#### *5.4.13.1 Description*

A webinos application is updated via remote trigger. Based on D2.1 use case LC2.

#### *5.4.13.2 Preconditions*

A newer version for an application is available (new features, security updates etc.).

#### *5.4.13.3 Flow*

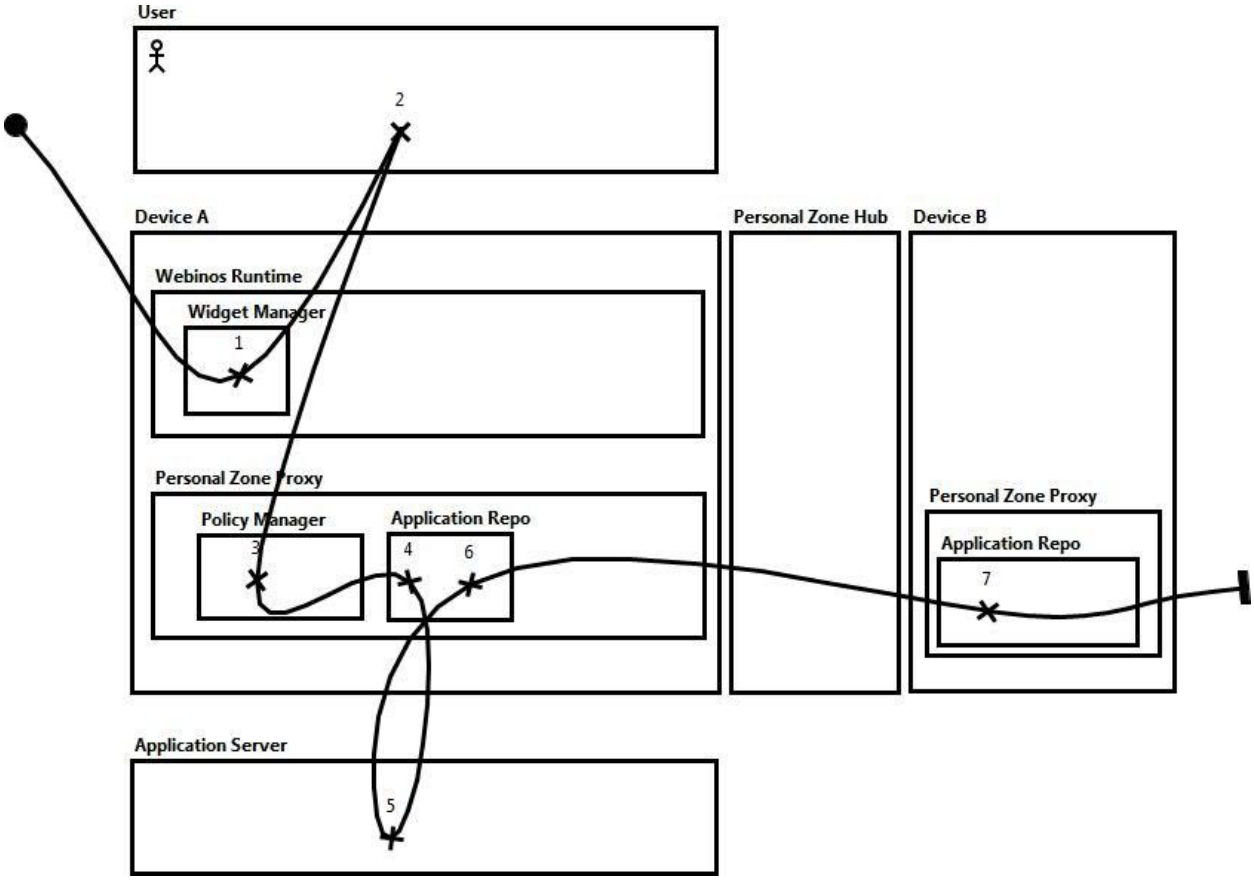
1. The webinos runtime becomes aware of the availability of an updated application and notifies the user.
2. User confirms the application update.
3. Prior to the application download, the policy manager performs a check if an update is allowed.
4. The download is triggered
5. The application packaged is downloaded from an application server to the local application repository
6. After the download is completed an integrity and security check is performed.
7. The application package is distributed to the other devices inside the personal zone

#### *5.4.13.4 Postconditions*

Application is updated;The previous, older, version will be deleted.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

### *5.4.13.5 Use Case Map*




### *5.4.13.6 Related Scenarios*

- None

### *5.4.13.7 Related Requirements*

- Application intent
- Event policy
- Policy management
- WAC/XACML policy
- Application event
- Application maintenance
- Multi-device install

## 5.4.14 LC3: Removal of applications

Author: Katrin Jordan

Actors: User

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.14.1 Description*

A webinos application is removed via remote action. Based on D2.1 use case LC3.

#### *5.4.14.2 Preconditions*

An application is already installed.

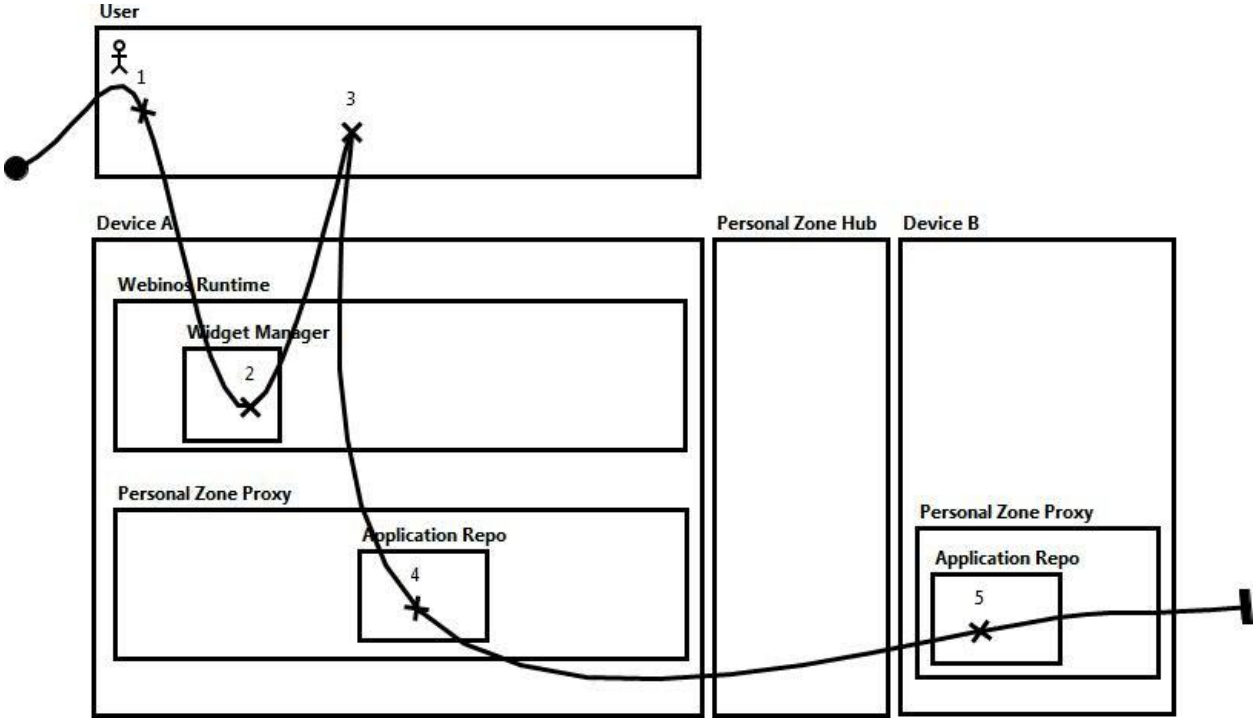
#### *5.4.14.3 Flow*

1. User opens the management view of the webinos runtime.
2. Widget manager presents the list of installed applications
3. User selects the application which shall be removed from the devices and confirms the selection.
4. The Widget manager deletes the application locally and triggers the removal on the remote devices.
5. Application is removed on remote devices.

#### *5.4.14.4 Postconditions*

Application is no longer available on the users device.

#### *5.4.14.5 Use Case Map*




#### *5.4.14.6 Related Scenarios*

- None

#### *5.4.14.7 Related Requirements*

- Application event

- Application maintenance

## 5.4.15 NC1: Content Adaptation

Author: Christian Fuhrhop

Actors: User

### *5.4.15.1 Description*

The system needs to be able to adapt the presentation of content to the capabilities of the system the content is presented on. Based on D2.1 use case WOS-UC-TA6-003.

### *5.4.15.2 Preconditions*

An application is showing a user interface that benefits from adapting to the capabilities of the device.

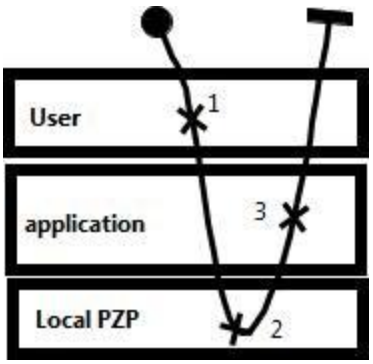
### *5.4.15.3 Flow*

1. A number of users participate in a video conference,
2. The video conference application asks for local device characteristics to the local PZP
3. The video conference application adapts what each user can see on the base of the specific device characteristics

### *5.4.15.4 Postconditions*

All participants of the video conference receive a user experience that is most appropriate for their device and environment.* Users with large screen devices (such as TVs) will see the video streams of all participating users in equal size.* Users with medium size screens (desktop computers) will see the current speaker in a large window and the streams of all participants in smaller thumbnail windows.* Users with mobile phones will only see the stream of the current speaker.* Users that are driving cars will not see a video stream, to avoid distractions, but will only receive an audio stream with additional acoustic cues to identify the current speaker.

### *5.4.15.5 Use Case Map*




### *5.4.15.6 Related Scenarios*

- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files
- S-NC2: Learning Assistance for Disabled Students

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- S-DA3: Converging Applications within and across Devices
- S-DA1: Smart Device Integration
- S-NC3: Background Monitoring Services

#### *5.4.15.7 Related Requirements*

- GUI components
- Interface layout services
- Media format information
- Audio and video access

### 5.4.16 NC2: Device Based Analytics Retrieval

Author: George Gionis

Actors: Application

#### *5.4.16.1 Description*

Webinos retrieves contextual information based on the device or device set under scope. Based on D2.1 use case WOS-UC-TA7-001.

#### *5.4.16.2 Preconditions*

The user has a device or set of devices with webinos installed.

#### *5.4.16.3 Flow*

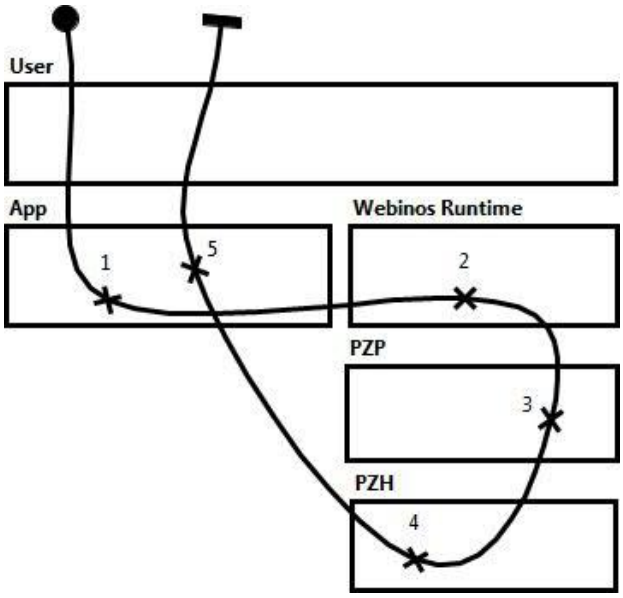
1. The Application queries the Webinos runtime for analytics information about the user's devices
2. The Webinos runtime queries the PZP for this information.
3. The PZP determines it doesn't possess this information, or this information is not up to date, and queries the PZH for this information.
4. The PZH checks the user's policies and returns the information allowed to be given. Examples can be: - device identity: device type, device model, hardware configuration. - device status: device is active (or not), operating system version, installed applications, currently running applications, current application active (on display). - device sensors readings: GPS, IP, accelerometer, photo camera, flash, speed, consumption, temperature, time. - connected devices: feature(s) currently shared to other device(s), feature(s) currently shared by other device(s), metadata of connected devices (could be a subset of all the above). The above information are an indicative set of the analytics information the system can retrieve from the registered users devices.
5. The Application accesses the data retrieved by webinos

#### *5.4.16.4 Postconditions*

The application can modify its behavior according to the context retrieved

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.16.5 Use Case Map*




#### *5.4.16.6 Related Scenarios*

- None

#### *5.4.16.7 Related Requirements*

- Analytics control
- Policy spec
- Feature access
- OS information
- Analytical information
- Personal device management

### 5.4.17 NM1: Subscribe and Publish Events

Author: Anders Isberg

Actors: Application

#### *5.4.17.1 Description*

Applications can publish application specific information. When the information is published the data is distributed to all authorised applications that are subscribed to the information. The application can decide whether the information is distributed to a specific set of receivers or to all applications that are interested in the specific information type. Thus, information can be also distributed without having prior knowledge about the destinations of the distributed information. Based on D2.1 use case WOS-UC-TA4-001.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.17.2 Preconditions*

Application A wants to publish information to others is running one at least one device. Application B has subscribed to the information matching the subscription of Application A.

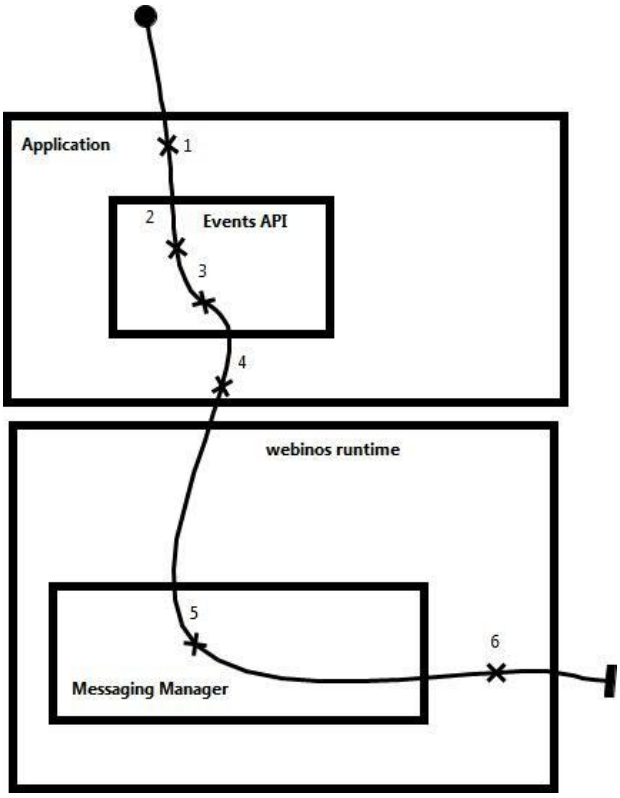
#### *5.4.17.3 Flow*

1. Application A creates an identifiable event that describes the information that should be published.
2. Application A attaches information about the intended recipients, which everybody that has subscribed to the identifiable event.
3. Application A attaches caching data indicating how long the event should be stored for if recipients cannot be contacted.
4. Application A triggers the publication of the information.
5. webinos checks who has been subscribed to the event in question and attempts to deliver the event irrespective of subscribing application location.
6. webinos detects that Application B is not reachable and caches the event for the specified time frame.

#### *5.4.17.4 Postconditions*

All authorised applications that are subscribed to the information and are reachable considering the given caching restrictions have received the published information.

#### *5.4.17.5 Use Case Map*




This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.17.6 Related Scenarios*

- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files
- S-CAP1: Creating Applications for webinos

#### *5.4.17.7 Related Requirements*

- Application event
- Event distribution
- Event type subscription
- Event type unsubscription
- Multiple entity eventing
- Subscription
- Wake-up message subscription

### 5.4.18 NM2: Subscribe and Publish Events (alternative)

Author: Anders Isberg

Actors: Application

#### *5.4.18.1 Description*

In addition to application based publishing of events webinos itself is able to publish a set of system events like, for example, system was booted, WiFi connection becomes available, low battery level, after a certain interval, at a given timer or others. A full list of events should be defined in the related requirement and specification documents. Based on D2.1 use case WOS-UC-TA4-001_1.

#### *5.4.18.2 Preconditions*

Webinos is running on at least one device and an application that wants to be informed about webinos events is currently running.

#### *5.4.18.3 Flow*

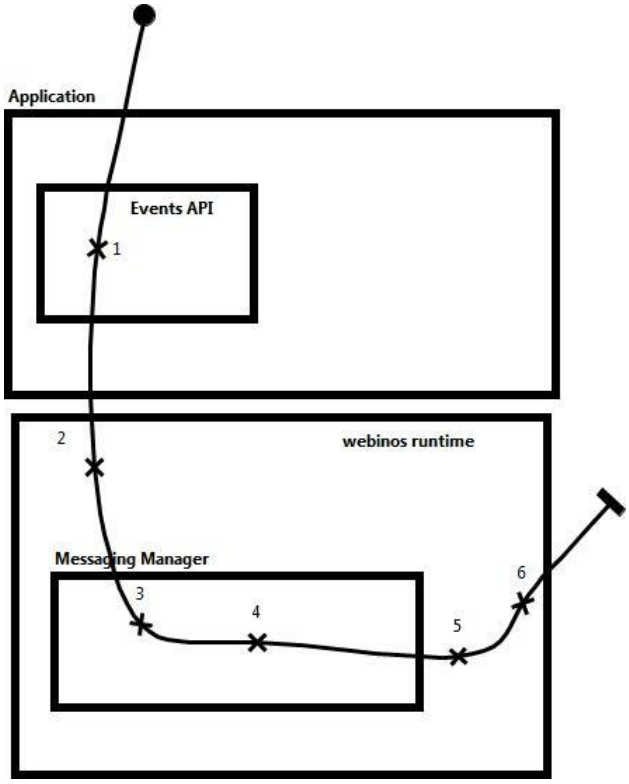
1. Application subscribes to specific events that are published by the webinos runtime.
2. webinos detects that conditions for a specific webinos event are met (e.g., WiFi connectivity, low battery level or other pre defined and specified webinos system events).
3. webinos creates an event that describes the occurred conditions.
4. webinos publishes the event.
5. webinos checks which applications have been subscribed to the event in question.
6. The event is distributed to all subscribing applications on the same or other devices, including the Application.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.18.4 Postconditions*

Application has received the published information.

#### *5.4.18.5 Use Case Map*




#### *5.4.18.6 Related Scenarios*

- None

#### *5.4.18.7 Related Requirements*

- Entity event subscription
- Event distribution
- Event name conflict
- Event type subscription
- Multiple entity eventing
- Subscription
- Unique event instance
- webinos event creator
- webinos event

### 5.4.19 NM3: Start an application triggered by a Published Event

Author: André Paul, based on WOS-UC-TA4-002

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

Actors: Application, User

#### *5.4.19.1 Description*

webinos allows applications to publish and subscribe to application based events. In addition to this, webinos itself publishes certain events where applications can be registered to. webinos allows a permanent subscription to events so that applications that are not running are automatically started based on the event and afterwards the application can receive the event. Based on D2.1 use case WOS-UC-TA4-002_3.

#### *5.4.19.2 Preconditions*

An application subscribes to a wifi connectivity event with the request to be automatically started if it is not already running.

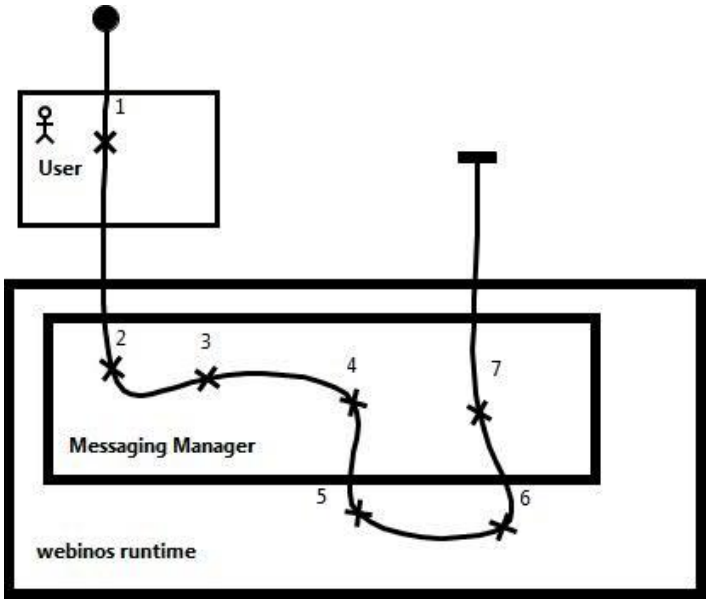
#### *5.4.19.3 Flow*

1. User terminates the application.
2. webinos detects that conditions for a specific webinos event for wifi connectivity is met.
3. webinos creates an event that describes the occurred condition.
4. webinos publishes the event.
5. webinos checks which applications have been subscribed to the event in question.
6. webinos starts applications that are registered to the event but that are currently not running.
7. The event is distributed to subscribing applications running on the same or on other, remote, devices, including the application.

#### *5.4.19.4 Postconditions*

The application has been started, is reachable, and has received the published information.

#### *5.4.19.5 Use Case Map*




This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.19.6 Related Scenarios*

- None

#### *5.4.19.7 Related Requirements*

- webinos event
- Distributed application installation
- Event-based application start-up

### 5.4.20 NM4: Proximity detection of Services

Author: Anders Isberg

Actors: Application, User

#### *5.4.20.1 Description*

A user prefers devices and services which are in physical proximity because they are more likely to relevant than other. Based on D2.1 use case WOS-UC-TA4-003.

#### *5.4.20.2 Preconditions*

Discovered service include meta data that describing its physical location.

#### *5.4.20.3 Flow*

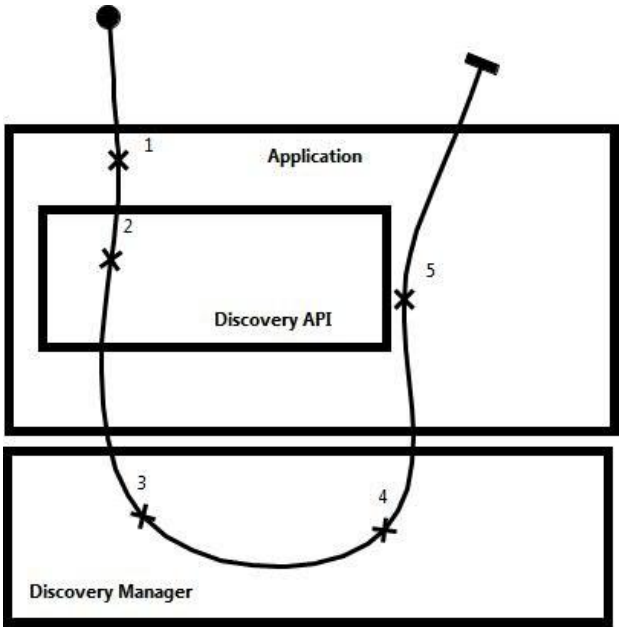
1. User executes an application that can make use of remote services.
2. Application requests service discovery to use a certain service type in close proximity.
3. webinos tries to discover the service types in question across all available and accessible devices.
4. webinos collects discovered service information and reports this back to the application.
5. Application makes use of the searched services in close proximity.

#### *5.4.20.4 Postconditions*

Application has discovered and make use of services in close physical proximity.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.20.5 Use Case Map*




#### *5.4.20.6 Related Scenarios*

- S-DA3: Converging Applications within and across Devices

#### *5.4.20.7 Related Requirements*

- Identity communication
- Remote device identity
- Application-based discovery
- Description-based discovery
- Device event discovery
- Device status discovery
- Location-based discovery
- Service availability detection
- Service discovery
- User social proximity
- User-offered service discovery
- Wireless connectivity discovery

### 5.4.21 PS1: Automatic login for External Services

Author: Christian Nord, Andre Paul

Actors: Application, User

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.21.1 Description*

A user automatically logs in to external services. Based on D2.1 use case WOS-UC-Ta2-003.

#### *5.4.21.2 Preconditions*

The user has already registered with the service and the service has been added to his webinos account.

#### *5.4.21.3 Flow*

1. The user points his Internet application, e.g., the browser, to a service end-point or runs a webinos application/runtime that provides access to the service.

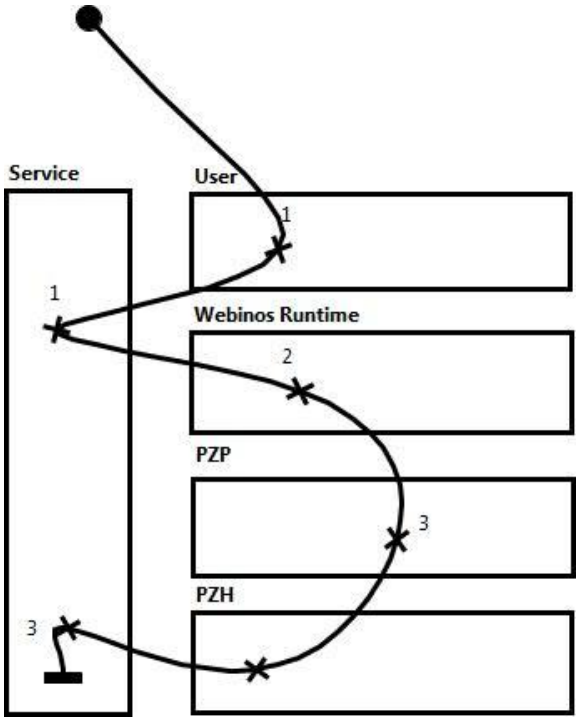
2. In the background the webinos runtime provides the service a unique identity identifier (Certificates) when the user connects to the service

3. By using the offered identity the service can, by accessing the webinos runtime for the user, determine if the user is already a registered user and grant access

#### *5.4.21.4 Postconditions*

The user is granted access to the service without login prompting.

#### *5.4.21.5 Use Case Map*




#### *5.4.21.6 Related Scenarios*

- S-NC3: Background Monitoring Services

#### *5.4.21.7 Related Requirements*

- Policy spec
- Service identity

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

## 5.4.22 PS10: Local storage of credentials

Author: John Lyle, Shamal Faily

Actors: Application, User

### *5.4.22.1 Description*

Credentials are stored by webinos as part of an application install. Based on D2.1 use case WOS-UC-TA9-012.

### *5.4.22.2 Preconditions*

A webinos-enabled application is being installed. The application specification indicates that webinos should store social network credentials. An app user signs up to a new social network.

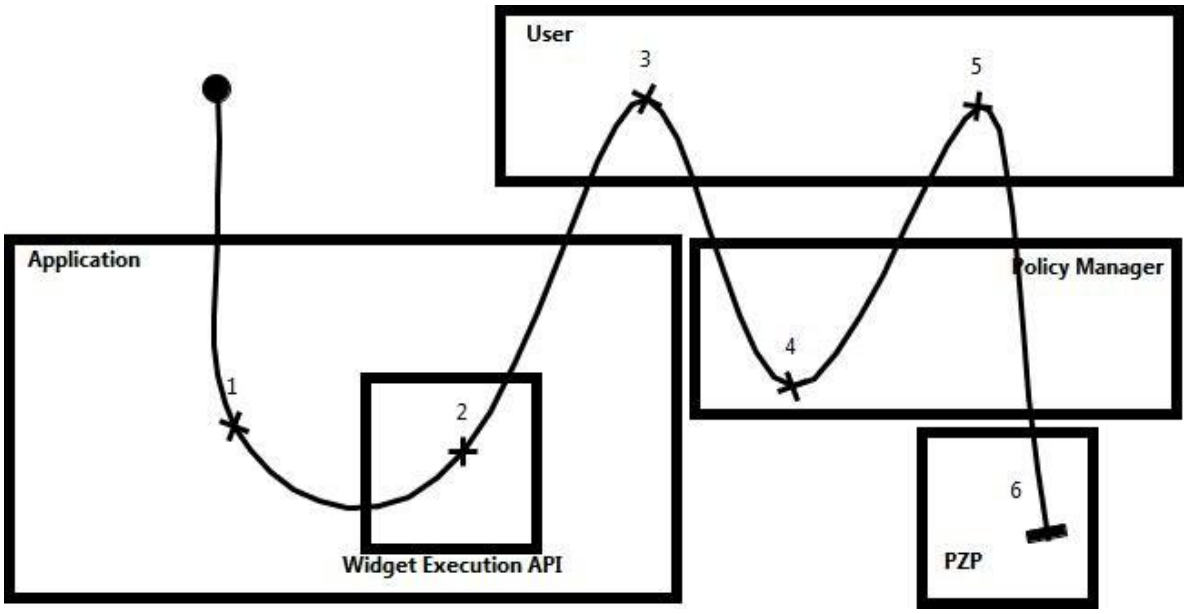
### *5.4.22.3 Flow*

1. User installs webinos-enabled social networking application.
2. Application requests social network credentials from the User.
3. User enters social network credentials.
4. Application requests permission to the PZP's policy manager to store social network credentials.
5. User grants blanket permission for webinos to store social network credentials.
6. Application stores social networking credentials within the device's PZP.

### *5.4.22.4 Postconditions*

User's credentials are stored by webinos and do not need to be re-entered whenever he is authenticated to his devices.

### *5.4.22.5 Use Case Map*




This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### 5.4.22.6 Related Scenarios

- None

#### 5.4.22.7 Related Requirements

- Confidential credentials storage
- Credentials access restriction
- Runtime alert
- WAC/XACML policy
- Application maintenance

### 5.4.23 PS2: Account Management

Author: Christian Nord, Andre Paul

Actors: User

#### 5.4.23.1 Description

The user wants to check his service subscriptions and authorizations with the webinos runtime. Based on D2.1 use case WOS-UC-Ta2-004.

#### 5.4.23.2 Preconditions

- A valid webinos runtime account related to the User that handles the account management must be available.* The user must be authenticated towards the webinos runtime.
- **Flow**

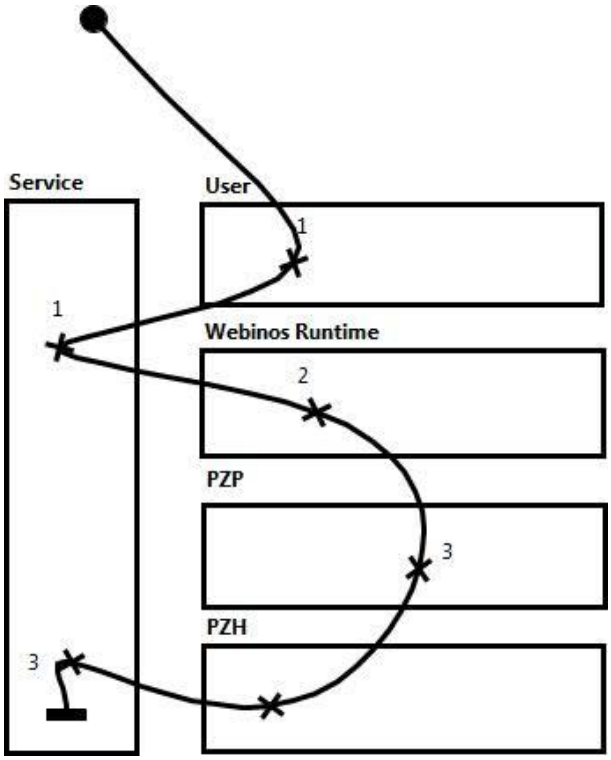
1. The user goes to his account management page.
2. The user can view all current accounts that he has signed up to, their validity and services that the user allows them to be used towards.
3. The user is now able to cancel any accounts and de-link accounts from service authorizations.

#### 5.4.23.3 Postconditions

The user's changes to his managed accounts are saved.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.23.4 Use Case Map*




#### *5.4.23.5 Related Scenarios*

- None

#### *5.4.23.6 Related Requirements*

- Policy management
- User identity
- Personal device management

### 5.4.24 PS3: Authentication

Author: Andre Paul

Actors: Application, User

#### *5.4.24.1 Description*

The user logs in to so he's authorized towards the webinos runtime. Based on D2.1 use case WOS-UC-TA2-005.

#### *5.4.24.2 Preconditions*

No trust between the webinos runtime and the User. The webinos runtime has knowledge about different trustworthy User identities.

- **Flow**

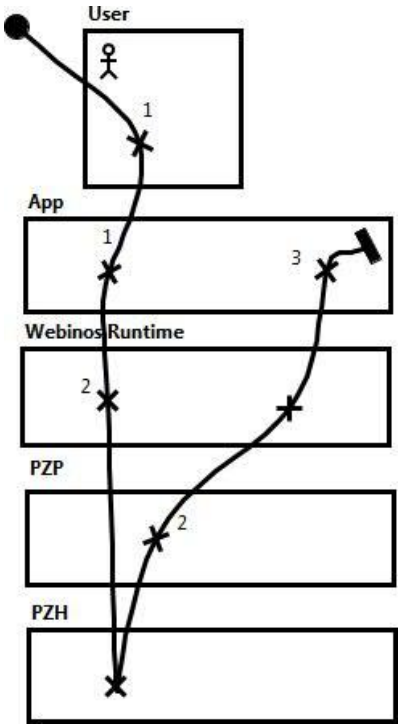
1. At the beginning of a session (e.g., turning on a device), the the webinos runtime presents the user with the option to log in or to use the system anonymously

2. The user logs in to the webinos runtime.
3. The webinos runtime checks the user's credentials/authorization key/...
4. After the user is logged in, he is allowed to access all of his services.

#### *5.4.24.3 Postconditions*

Trust between the webinos runtime and the user is accomplished.

#### *5.4.24.4 Use Case Map*




#### *5.4.24.5 Related Scenarios*

- S-TMS2: Single Sign-On, Multiple Devices
- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files

#### *5.4.24.6 Related Requirements*

- Hierarchical policy enforcement
- Policy management
- User identity
- Personal device management

### 5.4.25 PS4: Store Content in a Secure File Storage

Author: Anders Isberg

Actors: Application, User

#### *5.4.25.1 Description*

Applications may need to store application specific data or other content in a protected storage space. This storage space should be only accessible by the related application so that no other application can access or manipulate the stored data and even the user has no access. Based on D2.1 use case WOS-UC-TA4-005.

#### *5.4.25.2 Preconditions*

An application that needs to store private data is available to the user.

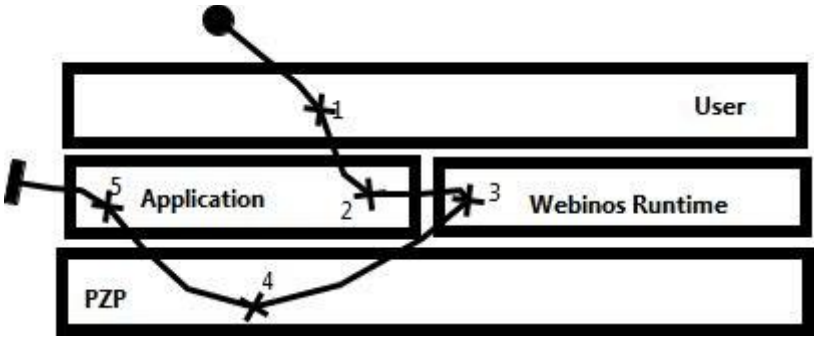
#### *5.4.25.3 Flow*

1. The end user starts the application
2. The application retrieves private data that must be stored in a protected environment (e.g., private user data provided by the user, application specific data generated by the application itself, or media files or other content)
3. The application requests API access to protected storage space
4. The policy manager inside the PZP checks if the application has write access to the protected storage, and grant the access on the base of configured policies
5. The application starts to write content to the protected storage space.

#### *5.4.25.4 Postconditions*

The content stored in the protected storage can be accessed for read and write operations by the original application but not by the user or by other webinos or native applications.

#### *5.4.25.5 Use Case Map*




#### *5.4.25.6 Related Scenarios*

- None

#### *5.4.25.7 Related Requirements*

- Hierarchical policy enforcement
- Policy management
- Runtime alert
- Secure storage

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

## 5.4.26 PS5: The publicity and privacy of analytics information

Author: John Lyle, Hans Myrhaug

Actors: Application, User

### *5.4.26.1 Description*

This use case describes two mobile phones interacting to share potentially private analytics information. It illustrates aspects of sharing analytics information across devices and applications. Based on D2.1 use case WOS-UC-TA7-006.

### *5.4.26.2 Preconditions*

A user and his friend both have a webinos-enabled mobile device that he uses daily. His friend has promiscuous policy settings for a social application that he is an active user of. The user has a least one friend on the same social network.Trust relationships between PZHs of the friends' zones are established.

### *5.4.26.3 Flow*

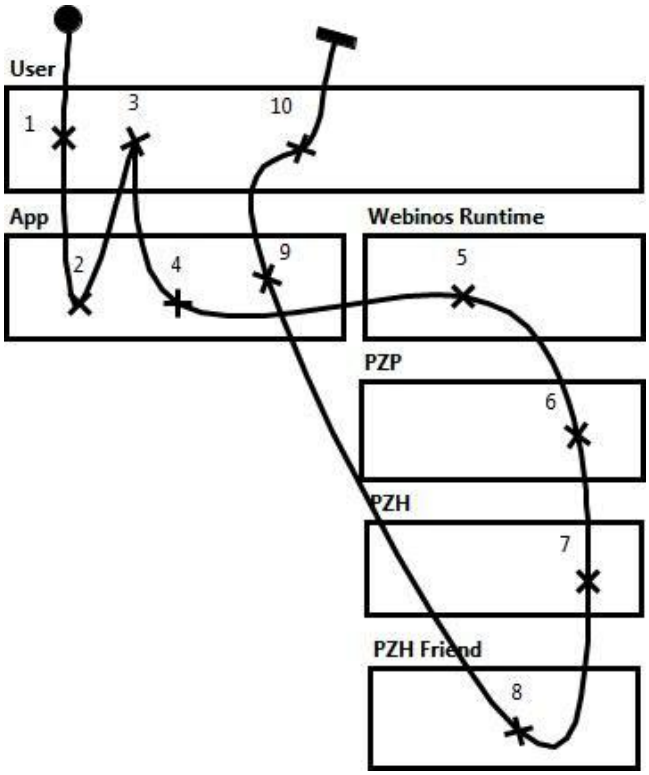
1. The User starts the social network application.
2. The Application shows that a friend has just updated his status to say that he is nearby.
3. The User has not seen his friend for a long time, and would like to see him. However, he does not want to interrupt, so instead requests some information from users mobile device instead of calling him.
4. The Application requests some of his friends data to the webinos runtime.
5. The webinos runtime requests the information to the PZP
6. The PZP requests the information to the PZH
7. The PZH requests the information to the friends PZH
8. The friend's PZH receives the request, determines if the policy allows sharing which data and returns the data allowed to be given. The policy is relatively promiscuous, and states that any of his friends may find out whether he is free later in the day, and whether any of his friends are nearby. However, it denies access to his location data.
9. The Application shows this data to the user

### *5.4.26.4 Postconditions*

The user receives analytics information about his friends activities in line with his friends personal profile settings.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.26.5 Use Case Map*




#### *5.4.26.6 Related Scenarios*

- None

#### *5.4.26.7 Related Requirements*

- Analytics control
- Hierarchical policy enforcement
- Analytical information
- User-offered service discovery

### 5.4.27 PS6: The publicity and privacy of context information (alternative)

Author: John Lyle, Hans Myrhaug

Actors: User

#### *5.4.27.1 Description*

This use case describes two mobile phones interacting to share potentially private analytics information. It illustrates aspects of sharing analytics information across devices and applications. Based on D2.1 use case WOS-UC-TA7-006_1.

#### *5.4.27.2 Preconditions*

A user and his friend both have a webinos-enabled mobile device that he uses daily. His friend has promiscuous policy settings for a social application that he is an active user of. The user has a least one

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

friend on the same social network.Trust relationships between PZHs of the friends' zones are established.

#### 5.4.27.3 Flow

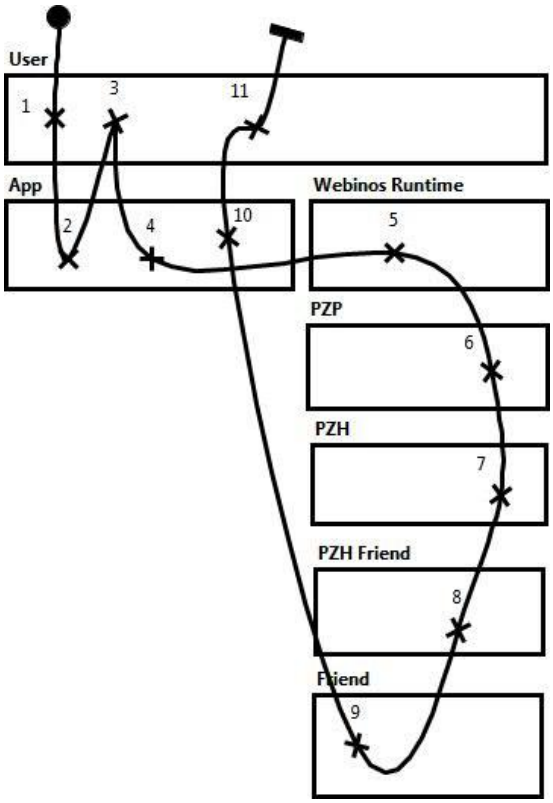
1. The User starts the social network application.
2. The Application shows that a friend has just updated his status to say that he is nearby.
3. The User has not seen his friend for a long time, and would like to see him. However, he does not want to interrupt, so instead requests some information from users mobile device instead of calling him.
4. The Application requests some of his friends data to the webinos runtime.
5. The webinos runtime requests the information to the PZP
6. The PZP requests the information to the PZH
7. The PZH requests the information to the friends PZH
8. The User's friend sets the policy to allow some of his analytics information to be shared, and that in the future all friends on his social network can do the same when he does not have plans in his diary.
9. The Application shows this data to the user.

#### 5.4.27.4 Postconditions

A User gains access to some of his friend's analytics information, and his friend's phone is configured to take the same decision when he is in a similar context.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.27.5 Use Case Map*




#### *5.4.27.6 Related Scenarios*

- None

#### *5.4.27.7 Related Requirements*

- Analytics control
- Analytical information
- User-offered service discovery

### 5.4.28 PS7: Interpreting policies and making access control decisions

Author: John Lyle

Actors: Application, User

#### *5.4.28.1 Description*

A freshly installed application wants to share his photos through a shared repository. This task is controlled by webinos policies. Based on D2.1 use case WOS-UC-TA9-002.

#### *5.4.28.2 Preconditions*

User has a tablet PC running webinosUser has a photo collection on his tablet PC which he would like to share with other people (Users) who are connected.The webinos runtime has a set of policies installed. These are not contradictory or ambiguous, so a decision can always be made based on their content.

#### *5.4.28.3 Flow*

1. The User has installed a new photo-sharing application

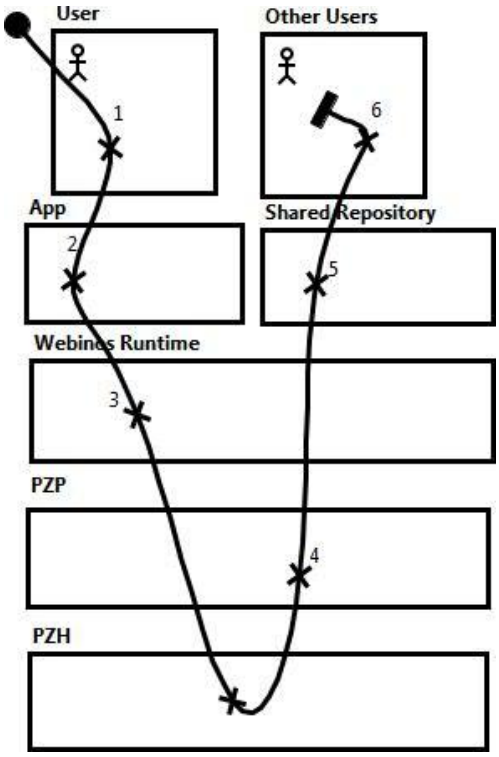
This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

2. The application automatically tries to upload his photos to a shared repository
3. Accessing photos stored on the webinos device his tablet pc results in an access-control request to the webinos runtime.
4. webinos runtime takes the event (application X accessing File F) and checks against the set of policies (PS) installed that are present in the PZP. In this case, the set of policies in the PZP is fresh and valid, so the PZP does not need to update it to the PZH to made the correct access control decision
5. The policies allow this action, and the application resumes uploading photos
6. Other users who are connected can access these photos

#### *5.4.28.4 Postconditions*

The photos are available for sharing with other online friends, on the base of the webinos policies

#### *5.4.28.5 Use Case Map*




#### *5.4.28.6 Related Scenarios*

- S-LC1: Designing Policy-aware webinos Applications
- S-CAP2: Report Disorder
- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files
- S-PS1: Usable Privacy Controls for webinos

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- S-NC3: Background Monitoring Services

#### 5.4.28.7 Related Requirements

- Application intent
- Event policy
- Policy management
- User policy authoring
- New installation notification

### 5.4.29 PS8: Interpreting policies and making access control decisions (alternative)

Author: John Lyle

Actors: Application, User

#### 5.4.29.1 Description

A freshly installed application wants to share his photos through a shared repository. This task is controlled by webinos policies. Based on D2.1 use case WOS-UC-TA9-002_2.

#### 5.4.29.2 Preconditions

User has a tablet PC running webinos. User has a photo collection on his tablet PC which he would like to share with other people (Users) who are connected.The webinos runtime has a set of policies installed. These are not contradictory or ambiguous, so a decision can always be made based on their content.

#### 5.4.29.3 Flow

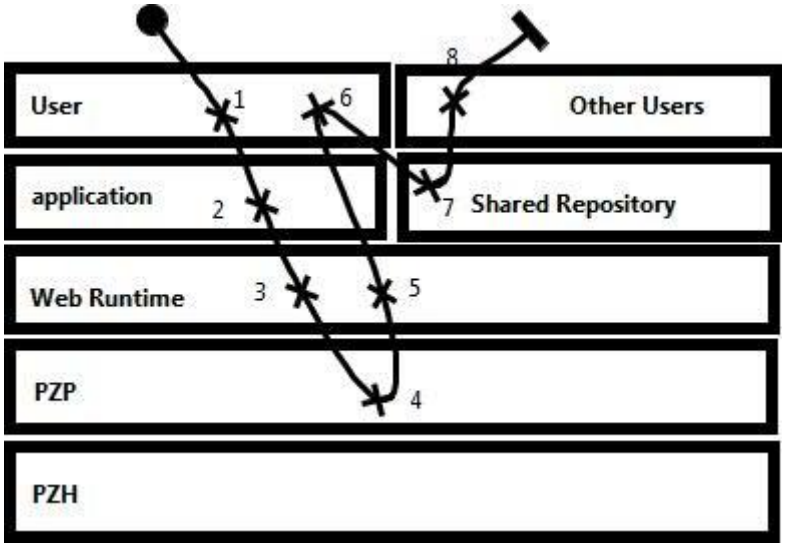
1. The User has installed a new photo-sharing application
2. The application automatically tries to upload his photos to a shared repository
3. Accessing photos stored on the webinos device his tablet pc results in an access-control request to the webinos runtime.
4. Webinos runtime takes the event (application X accessing File F) and checks against the set of policies (PS) installed that are present in the PZH/PZP. In this case, the set of policies in the PZP is fresh and valid, so the PZP does not need to update it to the PZH to made the correct access control decision.
5. Webinos prompts the User to confirm that access to photos is allowed
6. The User agrees
7. The photo-sharing app resumes uploading photos
8. Other users who are connected can access these photos.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.29.4 Postconditions*

The photos are available for sharing with other online friends, on the base of the webinos policies

#### *5.4.29.5 Use Case Map*




#### *5.4.29.6 Related Scenarios*

- S-LC1: Designing Policy-aware webinos Applications

#### *5.4.29.7 Related Requirements*

- Event policy
- Runtime alert
- New installation notification
- Local file storage access
- Remote device access
- Standard API

### 5.4.30 PS9: Enforcing multiple policies on multiple devices

Author: John Lyle

Actors: User

#### *5.4.30.1 Description*

Any policy enforcement done on one of the devices would change on the other device too. Based on D2.1 use case WOS-UC-TA9-003.

#### *5.4.30.2 Preconditions*

The User has webinos runtime running on his mobile and on an in-car entertainment system which is connected to a PZH.Policies are stored in PZH and mobile and in-car PZPs

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

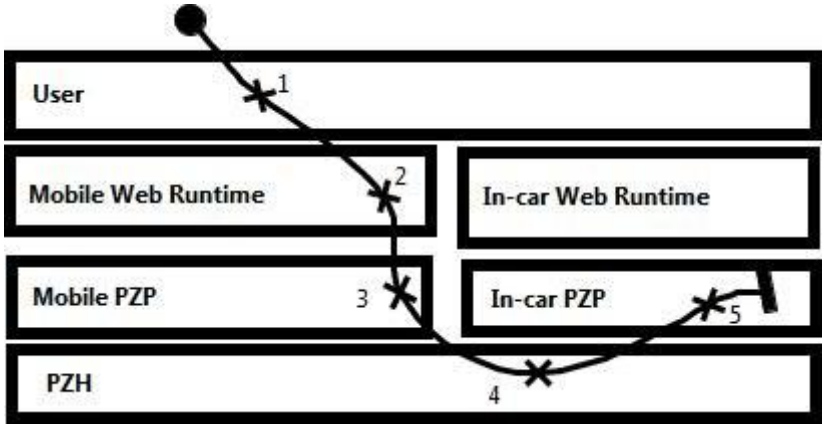
#### *5.4.30.3 Flow*

1. User switches on his webinos enabled mobile
2. User changes his zone policy on the mobile, expressing that he does not want to alert local networks about his presence.
3. the policies are stored locally on the PZP mobile.
4. the PZP changes the zone policies contacting the PZH
5. PZH update policies on the in-car PZP specifying that should remain hidden and not discoverable.

#### *5.4.30.4 Postconditions*

Any webinos device used by the User subsequently will not be discoverable by a remote network. This includes his car whenever he is using it.

#### *5.4.30.5 Use Case Map*




#### *5.4.30.6 Related Scenarios*

- S-PS3: Privacy Controls and Analytics for Corporations and Small Business
- S-LC1: Designing Policy-aware webinos Applications
- S-PS1: Usable Privacy Controls for webinos
- S-NC3: Background Monitoring Services

#### *5.4.30.7 Related Requirements*

- Confidential credentials storage
- Configurable device discovery
- Credentials access restriction
- Policy management
- Policy spec

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- Privacy preferences

### 5.4.31 TMS1: Linking device to a user account

Author: Simon Isenberg, Andre Paul

Actors: User

#### *5.4.31.1 Description*

The user of a device links a device to his personal zone in order to facilitate remote management of the device (data + policy synchronization, revocation, ...), make its services available in the personal zone and enable installing applications remotely. Based on D2.1 use case WOS-UC-TA1-007.

#### *5.4.31.2 Preconditions*

User has already running a personal zone. The device is uniquely identifiable within the personal zone. The user's device and webinos runtime is already running.

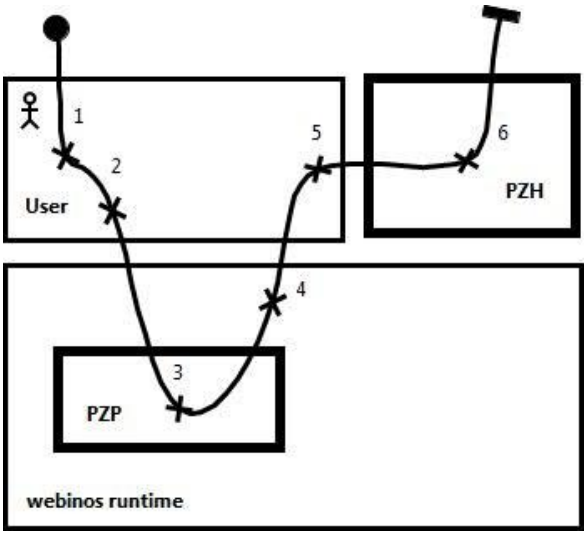
#### *5.4.31.3 Flow*

1. The user selects the function to add his device to his personal zone.
2. The user enters addressing information for his device.
3. webinos' device runtime requests the addition of the device to the user's personal zone.
4. webinos displays the details of the new device and verification information for secure pairing.
5. The user verifies the pairing.
6. webinos adds the new device to the list of user devices available within the user's personal zone.

#### *5.4.31.4 Postconditions*

Device is now a member of the user's personal zone, and services provided by the device can be used by locally or remotely from authorised webinos applications.

#### *5.4.31.5 Use Case Map*



#### *5.4.31.6 Related Scenarios*

- S-DA4: Remote Monitoring of Chronic Disease through Access to Medical Sensor Data and Sharing of Medical Files

#### *5.4.31.7 Related Requirements*

- Device-identity binding
- Personal device authentication
- Personal device management

### 5.4.32 TMS2: Data Synchronization

Author: Christian Fuhrhop

Actors: Application, User

#### *5.4.32.1 Description*

webinos needs to be capable of synchronising data between multiple devices so that applications as well as system components can share data across devices. Based on D2.1 use case WOS-UC-TA4-010.

#### *5.4.32.2 Preconditions*

Devices taking part in the synchronization are connected to each other so that data exchange (synchronization) is possible. The user runs an application that makes use of synchronised data, and the application uses webinos APIs to store application data that is subject to synchronisation.

#### *5.4.32.3 Flow*

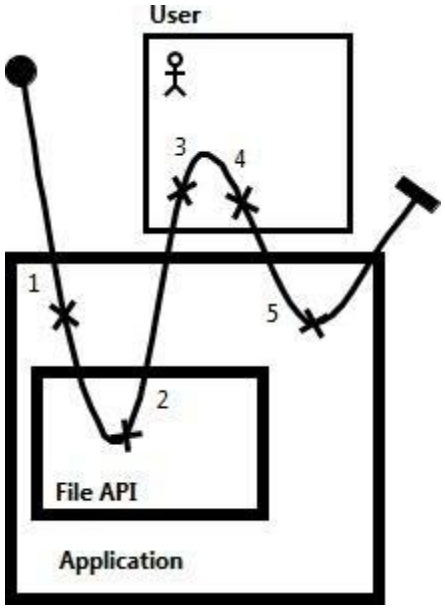
1. The Application updates synchronisable data.
2. webinos synchronises thechanged data with other devices where the same application is installed on.
3. User quit the application.
4. User starts the same application on another device.
5. The Application reads the synchronised data.

#### *5.4.32.4 Postconditions*

The application on the second device continues to work based on the synchronised data.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

#### *5.4.32.5 Use Case Map*




#### *5.4.32.6 Related Scenarios*

- S-DA2: Input Method Dictionary Sharing across Devices
- S-TMS2: Single Sign-On, Multiple Devices
- S-CAP1: Creating Applications for webinos
- S-NC3: Background Monitoring Services

#### *5.4.32.7 Related Requirements*

- Application data synchronisation
- Application running state
- Data subset application synchronisation
- Session re-establishment
- State and configuration transfer

### 5.4.33 TMS3: Removal of applications (alternative)

Author: Katrin Jordan

Actors: Application, User

#### *5.4.33.1 Description*

A webinos application is removed via remote action. Based on D2.1 use case WOS-UC-TA5-006_1.

#### *5.4.33.2 Preconditions*

An application on a user device is marked to be removed by an authorised entity (e.g. Service provider, User, administrator). A user is authorised to de-install the application on the target device.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

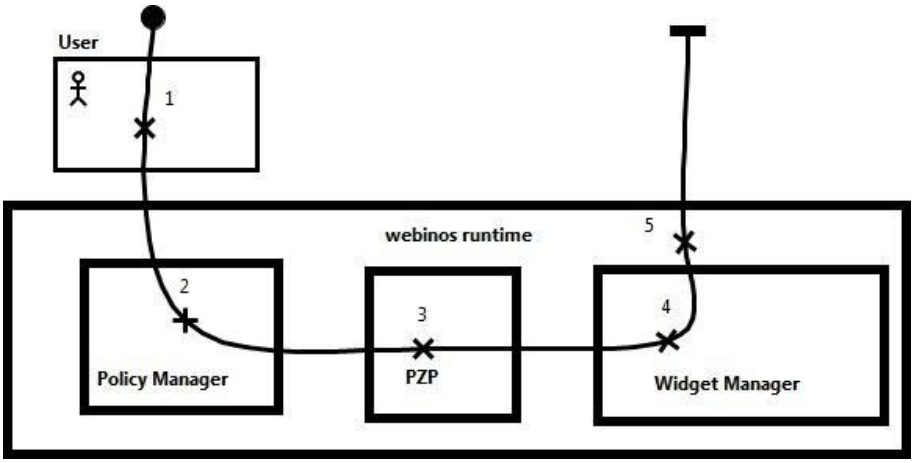
- **Flow**

1. User selects an application for removal using a removal application, and provides a reason for the removal.
2. webinos verifies that the user is authorised to remove the application from the target device.
3. webinos sends a deinstallation request to the user's device.
4. The webinos runtime on the device deinstalls the application.
5. webinos acknowledges removal of the application from the target device.

#### *5.4.33.3 Postconditions*

Application is no longer available on the users device, but personal data created by the device still remains.

#### *5.4.33.4 Use Case Map*




#### *5.4.33.5 Related Scenarios*

- None

#### *5.4.33.6 Related Requirements*

- Event policy
- Runtime alert
- Application maintenance
- Device personal data uninstall
- Maintenance authorisation

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 6 Conclusions

## 6.1 Summary of findings

Deliverable D2.4 provides the definition of use cases and scenarios for phase 2 of the webinos project which are mainly based on updating D2.1 as a result of the specifications, implementation, and requirements work done during phase I of the project. The listed scenarios build the foundation of the webinos project by describing what type of applications could be realized using webinos (scenarios), and also by describing which features are provided by the webinos platform itself (use cases).

A considerable number of use cases were removed because they were considered as being out of scope. Most of them were originally already grouped as being "Example Use Cases" where it was not clear at the early time if the use case will be part of the webinos platform or if the use case is completely separated from webinos, for example as being a application specific use case. The document is still a living document because new scenarios and use cases may be added over time based on feature requests, both from within and outside the project.

## 6.2 Exploitation of deliverable

- **D2.5: Updates on Requirements and available solutions**

Use cases are exploited by D2.5 because they provide additional context to requirements. Each requirement contains a list of use cases that are associated with it. Additionally, the metrics used to determine pre-requirements traceability are informed by whether or not use cases operationalise how requirements are satisfied. Collectively, this information provides a useful means for motivating requirements.

- **D3.3 webinos phase 2: Architectural and Components**

Much of the work being carried out as part of D3.3 will involve aligning the requirements elicited in D2.5 with the system specifications and architectural views of the webinos platform. The use case maps produced as part of this deliverable will play an important role in facilitating this exercise. Not only do these describe the expectations about what architectural components are responsible of different behavioural aspects of webinos, the use cases are also closely aligned with the requirements. This allows architectural designers to determine which requirements need to be refined in order to align with the system specifications.

- **D3.6: webinos phase 2: Security Framework**

The format which has been adopted for scenarios means that it will now be possible to use these scenarios to supplement the architectural risk analysis activities planned as part of D3.6. In particular, when architectural views and situated in threat environments, it is necessary to indicate what stakeholder roles are affected by specific architectural decisions. Because the personas created in D2.7 and used in this deliverable fulfil one or more of these roles, it is possible -- with the aid of tool-support -- to align scenarios with architectural design decisions and define whether specific design considerations positively or negative affect persona activities described in particular scenarios. This work is described in more detail in (FAFL2010).

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

- **D2.9: Cross-screen applications competition**

The deliverable will outline the final processes followed based on the initial plan that was introduced in chapter 2 of this deliverable and the final results of the activity including documentation that will be produced during the competition.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 7 Annex: Terms & Conditions

(based on http://10k.aneventapart.com/legal)

**THE WEBINOS CROSS-SCREEN PRIZE CHALLENGE OFFICIAL RULES**

**PLEASE NOTE:** It is your sole responsibility to review and understand your employer's policies regarding your eligibility to participate in trade promotions. If you are participating in violation of your employer's policies, you may be disqualified from entering or receiving prizes. webinos disclaims any and all liability or responsibility for disputes arising between an employee and their employer related to this matter, and prizes will only be awarded in compliance with the employer's policies.

**COMMON TERMS USED IN THESE RULES:**

These are the official rules that govern how the WEBINOS CROSS-SCREEN PRIZE promotion ("Contest") will operate.

In these rules, "we," "our," and "us" refer to **webinos consortium**, the sponsor of the Contest. "You," "yourself," "I," "me," and "my" refers to an eligible Contest entrant.

**1. CONTEST DESCRIPTION**

This is a skill-based Contest. Entrants create and submit a webinos-based web application to the webinos appstore. Refer to Section 3 ("WHO CAN ENTER") for eligibility determination and Section 4 ("WHAT CONSTITUTES AN ELIGIBLE ENTRY") for additional entry requirements and details. For purposes of this Contest, the web application that you create and submit in the Contest will be called an "entry." All eligible entries received will be judged using judging criteria described in Section 7 ("HOW ENTRIES WILL BE JUDGED").

**2. WHAT ARE THE START AND END DATES?**

This Contest starts at **(TBA)** and ends at **(TBA)** ("Entry Period"). Entries must be received within the Entry Period to be eligible.

**3. WHO CAN ENTER?**

You are eligible to enter this Contest if you meet the following requirements at time of entry:

- You are of 18 years or older or provide written permission by your parent or legal guardian.
- You are NOT a member of webinos consortium or an employee of a webinos partner; and
- You are NOT involved in any part of the administration and execution of this Contest; and
- You are NOT an immediate family or household member of a webinos consortium partner company employee or an employee of a webinos consortium partner company subsidiary.

**4. WHAT CONSTITUTES AN ELIGIBLE ENTRY**

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

To be eligible for judging, an entry must meet the following requirements:

- Use of at least one webinos API & at least two screens.
- Applications must be submitted and accepted at the webinos appstore (check competition website for URL) in order to be eligible.
- Apps must be accompanied by a clear description, screen-shots, and optionally (but we recommend to) with a video and/or screen-capture demoing their usage. The developer at his discretion may submit additional documentation and/or instructions.
- If an application requires special hardware (e.g. fixed appliances/machinery or very expensive equipment) or proprietary/confidential/sensitive data sets in order for its full functionality to be tested, the Judges may request additional proof of the functionality claims (e.g. arrange a live demo) and agree on how this will be publicly demonstrated / show-cased during the awards event before they approve the submission. It remains to the Judges' discretion to disqualify an entry if they believe that it does not meet the criteria.
- Complete the online survey at the end of the application form.
- Give one interview for quality/technical feedback and/or marketing purposes.
- The entry may utilize any combination of approved libraries as specified on the FAQ's page without modification;
- Data from third-party web services, third-part source code, libraries, or assemblies are permitted, provided that you have obtained all consents, approvals, or licenses required and you submit it into this Contest along with your entry.

In addition:

- your entry must be your own original work; and
- your entry cannot have been selected as a winner in any other contest; and
- you must have obtained any and all consents, approvals or licenses required for you to submit your entry; and
- your entry must not otherwise violate the rights of any other person or company without their express written consent.

Entries may NOT contain, as determined by us, in our sole and absolute discretion, any content that:

- is sexually explicit, unnecessarily violent or derogatory of any ethnic, racial, gender, religious, professional or age group; profane or pornographic;
- is obscene or offensive;
- defames, misrepresents or contains disparaging remarks about other people or companies;
- communicates messages or images inconsistent with the positive images and/or good will to which we wish to associate; and/or violates any law;

We reserve the right to reject any entry, in our sole and absolute discretion, that we determine does not meet the above criteria.

**5. HOW MY ENTRY MAY BE USED**

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

Other than what is set forth below, we are not claiming any ownership rights to your entry. However, by submitting your entry, you:

- Agree that, webinos may: (i) use, review, assess, test and otherwise analyze your entry and all its content in connection with this Contest; (ii) feature your entry in connection with the promotion of this Contest and webinos events in all media (now known or later developed).
- By entering this Contest, you agree that use of information in our representatives' unaided memories in the development or deployment of our products or services does not create liability for us under this agreement or copyright or trade secret law; and
- Understand that you are not entitled to any compensation or credit for use of your entry, other than what is described in these Official Rules

Please note that following the end of this Contest your entry may be posted on a website selected by us for viewing by visitors to that website. We are not responsible for any unauthorized use of your entry by visitors to this website. While we reserve these rights, we are not obligated to use your entry for any purpose, even if it has been selected as a winning entry.

If you (or your parent or legal guardian) do not want to grant us these rights to your entry, please do not enter this Contest.

**6. HOW TO SUBMIT YOUR ENTRY**

To submit your web application, navigate to **(TBA)** and enter the Contest by submitting your web application, as instructed. **We will accept (3) entries per person**. We are not responsible for entries that we do not receive, or that we are unable to access, for any reason, or for entries that we receive but are not decipherable for any reason.

We will automatically disqualify any incomplete or illegible entry and any entries that we receive that do not meet the requirements described above and in Section 4 ("WHAT CONSTITUTES AN ELIGIBLE ENTRY").

**7. HOW ENTRIES WILL BE JUDGED**

Winners will be determined by the judges' panel listed on the home page of the contest.

**Judge's Choice**

Following the close of the Entry Period, a panel of judges will review all eligible entries received and, based on the judging criteria, select a number of finalists or prize winners depending on the format of the competition. The decisions of the judges are final and binding. If we do not receive a sufficient number of entries meeting the entry requirements, we reserve the right to not select any winners.

**8. THE PRIZES**

If you are selected as a potential winner of this Contest:

- The prize will be awarded to you and you shall ensure it is used and/or distributed in accordance with your company's policies (Note: We are not responsible for the re-distribution of prizes within your company, if required); and
- You may not designate someone else as the winner. If you are unable or unwilling to accept your prize, we will award it to an alternate potential winner; and
- If you accept a prize, you will be solely responsible for all applicable taxes related to accepting the prize.

**9. HOW WINNERS WILL BE NOTIFIED**

If you are a finalist, we will notify you by sending a message to the e-mail address listed on your entry within seven (7) days following finalists determination. If the notification that we send is returned as undeliverable, or you are otherwise unreachable for any reason, we may consider an alternate finalist.

If there is a dispute as to who is the potential winner, we will consider the potential winner to be the authorized account holder of the e-mail address used to enter the Contest.

Winners will be determined either by a Judges panel or a public voting procedure at the awards event. (specific process to be described as per the specific event)

**10. OTHER CONDITIONS YOU ARE AGREEING TO BY ENTERING THIS CONTEST**

By entering this Contest you (your parent or legal guardian if you are a minor) agree and understand that:

- To abide by these Official Rules;
- To the extent allowed by law, to release and hold harmless webinos consortium, their respective parents companies subsidiaries, affiliates, employees and agents from any and all liability or any injury, loss or damage of any kind arising from or in connection with this Contest or any prize won;
- That webinos' decisions will be final and binding on all matters related to this Contest; and
- That webinos may use of your proper name and country of residence online and in print, or in any other media, in connection with this Contest, without payment or compensation to you, except where prohibited by law.

**11. IF SOMETHING UNEXPECTED HAPPENS AND THE CONTEST CAN'T RUN AS PLANNED**

If any unforeseen or unexpected event that cannot be reasonably anticipated or controlled, (also referred to as force majeure) affects the fairness and / or integrity of this Contest, we reserve the right to cancel, change or suspend this Contest.

**12. FIND OUT WHO WON**

We will post the winning entries online after **(TBA)** at **(TBA)**. This posting will remain active for at least 30 days.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

**13. CONTEST SPONSOR**

The European Commission via webinos project is the main sponsor of this Contest.

**14. PRIVACY**

webinos project Privacy Policy will apply to this Contest, and to all information that we receive from your entry. Please read the Privacy Policy on the webinos project (http://webinos.org) site before accepting the Official Rules and submitting your entry. Please note that by accepting the Official Rules you are also accepting the terms of the Privacy Policy.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.

# 8 References

**BUHR1996**

- Buhr, R. J. A., and Casselman, R. S. Use Case Maps for Object-Oriented Systems. Prentice Hall, 1996.

**COCK2001**

- Cockburn, A. Writing Effective Use Cases. Addison-Wesley, 2001.

**CONTENT**

- FI-Content project, retrieved 21 March 2012, <http://www.fi-content.eu>.

**COOP1999**

- Cooper, A. The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity (2nd Edition). Pearson Higher Education, 1999.

**CORE**

- FI-Ware project, retrieved 21 March 2012, <http://www.fi-ware.eu/>.

**FAFL2010**

- Faily, S., and Flechais, I. Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. International Journal of Secure Software Engineering 1, 3 (July-September 2010), 56–70.

**WEBGL**

- WebGL - OpenGL ES 2.0 for the Web, Khronos Group, retrieved 29 March 2012<http://www.khronos.org/webgl/>.

**KHRONOS**

- The Khronos Group, retrieved 29 March 2012 <http://www.khronos.org/>

**PAYPAL**

- PayPal Developer Central, retrieved 29 March 2012 <https://www.paypalobjects.com/IntegrationCenter/ic_sdk.html>.

This work is partially funded by webinos, an EU-funded project under the EU FP7 ICT Programme, No 257103.