Bringing Autonomic Services to Life

D6.5—Application Scenario Implementation (Companion Document)

Status and Version:	Final	
Date of issue:	11.01.2008	
Distribution:	Project Internal	
Author(s):	Name	Partner
	Antonio Di Ferdinando	ICL
Checked by:	Antonio Manzalini	TI

Table of Contents

1 Introduction

1.1 Purpose and Scope

This document is intended as a companion description for the deliverable D6.5, "Application Scenario Implementation". Such deliverable, in fact, of type "Prototype", foresees the delivery of the actual source code subject of the project review.

The present document, thus, has been thought for the purpose of describing the implementation of the CASCADAS application scenario. The scenario incorporates uses of pervasive computing and wide area communications where scalability and real-time constraints are crucial and difficult to establish with current technologies. The document explains the way components of the application scenario interact to the extent of providing a real-time pervasive advertisement scenario.

1.2 Reference Material

1.2.1 Reference Documents

[1] D6.1— Part A: "Description of Application Scenarios and of the Services to be Provided"

[2] D6.2— Part A: "Experimental QoS Evaluation in Autonomic Network Environments"

[3] D6.2— Part B: "Distributed Test-bed Specifications"

[4] D6.3— "Proof of Concept Design of the Application test-bed"

[5] D4.1— "Design of a Distributed Security Architecture"

[6] D3.1— "Aggregation Algorithms, Overlay Dynamics and Implications for Self-Organised Distributed Systems"

[7] D4.2— "Open-source toolkit for security in CASCADAS

1.2.2 Acronyms

PAE : Profile Assessment Environment

AE :Auction Environment

CIS: Crowd Information Service

DBE : Database Environment

KNE : Knowledge Network Environment

Bringing Autonomic Services to Life

HMA : Hashing for Message Authentication

HMAC: Hashing for Message Authentication Code

1.3 Document History

Version	Date	Authors	Comment
0.1	28/09/2007	Antonio Di Ferdinando	Initial ToC
0.2	29/09/2007	Antonio Di Ferdinando	Structure of initial content
0.3	01/12/2007	Antonio Di Ferdinando	Initial Draft
0.4	10/12/2007	Antonio Di Ferdinando	Integration of description for other demos
0.5	13/12/2007	Antonio Di Ferdinando	Modification after feedback
0.6	18/12/2007	Antonio Di Ferdinando	Integration of supervision system
Final	18/12/2007	Antonio Manzalini	Revision and check

2 Structure

Novelty of the research carried out in the context of the CASCADAS project will be demonstrated through implementation of a real-life application scenario through the use of the latest version of the CASCADAS Toolkit.
In order to catch and widen the range of aspects of the technology there developed, the aforementioned application scenario will be divided into three separate demonstrations. More precisely:

- **Pervasive Advertisement**: Will provide a scenario where a set of ACEs governs a system through which it will be possible to select advertisements according to patterns with which tastes and interests of the audience present in the venue evolve in time in a pervasive fashion. The actual advert so be shown at a certain moment will be determined through a distributed auction, conducted remotely, where ACE-governed machines simulating business companies will compete to the extent of acquiring the rights to advertise for an *a priori* determined slot of time the good or service that better suits tasted and interests of the crowd currently present in the venue. Advertisements will have the form of multimedia files to be played at the display in the venue. This demonstration will employ technology developed in WP1, WP2 and WP5.

- **Self-Organized Aggregation for Pervasive Advertisement:** will focus on showing self-organization techniques to match the type of adverts shown on public displays to the changing predominant tastes and interests of a variable audience (i.e. passers-by). This demonstration will focus on showing effectiveness of algorithms developed in WP3.
- **Authenticated ACE-Based Communication:** will focus on showing how authentication and cryptographic algorithms can be integrated into ACEs so as to enhance them with security features. This demonstration will draw contributions from WP1 and WP4.

2.1 Pervasive Advertisement

CASCADAS Application Scenario can be broadly divided into 5 environments, integrated:

- **Profile Assessment Environment (PAE):** is the environment where profiles are tracked. This environment is built of several (possibly heterogeneous) devices physically located in the demonstrator venue. Each of these will build an own profile through the use of an RFID reader. No ACEs are expected to be present.
- **Knowledge Network Environment (KNE):** is the environment where anonymous profiles are received and analyzed in order to build the knowledge network. This environment will be composed of a device running an ACE receiving profiles and providing the **Crowd Information Service (CIS)**. The device hosting the ACE will be physically located in the venue chosen for the demo.
- **Auction Environment (AE):** is the environment where slots of time will be auctioned. This environment will be composed of 12 machines (1 seller, 1 Auction Centre and 10 bidders) each of which will run an ACE. Bidders are expected to contact the KNE to the extent of obtaining a statistical analysis of people present in the environment where the winning advertisement will be displayed. In case this solution shows to be unfeasible (whose motivation would most likely to be security permissions in the testbed at ICL), information obtained by the CIS might be embedded as a description for the time slot to be auctioned. The ACE on the seller will export an **Auction Service (AS)**.
- **Database Environment (DBE**): is the environment where multimedia contents, i.e. advertisements to be shown, will be contained. Databases will be categorized according to 'tags' defining owner company, good or service subject of the movie and relevant interest.
- **Display Environment (DE):** is the environment where advertisements will actually be displayed. This environment will be composed by a (probably fixed) device physically connected to a display screen. An ACE will run inside the device, and will offer a **Display Service (DS)**. The display screen will be divided into a grid, and each of the cells of the grid will be a distinct auction. However, auctions will need to be synchronized in a way that they will need to terminate at the same time, so that cells representing auctions won by the same bidder can be aggregated and resulting in the advertisement to be shown at bigger resolution.

The 5 environments described above will coordinate to the extent of realizing the advertisement scenario characterized by pervasive features. Such coordination is described in detail in [4].

2.1.1 Dynamics

The pervasive advertisement scenario aims to emulate a future scenario where enterprises compete to acquire the rights of advertising in a display. This latter is assumed to be physically located in a venue subject to passage and staying of a nigh number of persons, such as for instance a shopping mall. Enterprises are assumed to be present in a multi-agent platform governed by ACEs, and compete to acquire advertisement rights through a real-time electronic networked auction.

In the context of our demo, ten enterprises are assumed to be represented by ACEs. Also, we defined a set of business areas each enterprise is interested in. In particular for the application scenario we have chosen the following enterprises:

Enterprise	Expedia	Virgin	Blockbuster	Sony	Apple
Interests	Travel, Internet	Travel, Internet, reading, cinema, music	Cinema, gaming, music	Cinema, internet, gaming, music	Cinema, internet, music, gaming
Enterprise	Amazon	Microsoft	AmEx	The Guardian	Kodak
Interests	Reading, internet, cinema, music, sport	Internet, gaming, music	Travel, internet, cinema, gaming, music, reading,	Cinema, internet, travel, sport, reading, music	Sport, music, travel

Table 1: Enterprises chosen as auction competitors.

The choice of what enterprises to include has been made in such a way that the corresponding business interests guarantee a minimum overlap. This will ensure a minimum competition level in the auction.

Enterprises business interests match interests contained in the personal profiles. These will be collected in the Profile Assessment Environment (PAE) through the use of devices equipped with RFID readers, governed by one ACE, where each RFID will be associated a profile containing a randomly generated set of interests. As RFID antennas will read profiles, they will be sent to the Knowledge Network and there processed in such a way to obtain a snapshot of interests, associated with the current crowd, of statistical nature.

At a certain point, an on a regular time basis thereafter, the ACE governing the Display Environment (DE) will contact the auctioneer in the Auction Environment (AE) and ask to auction a fixed-length slot of time, where an advertisement can be shown in the display, among the set of competitors. According to the auctioning model, then, competitors will come acquainted of the auction and query the Knowledge Network (KN) to obtain the snapshot of the current composition of the audience in the venue. The actual composition will contain a hierarchy of interests, which will give competitors an eventual incentive to submit a bid on the auction. The incentive will obviously be given according to a matching between the hierarchy and each competitor business interest, and a bid will be submitted if and only if a relevant matching is detected.

Bidders will thus start to compete on the auction, querying the KN for context information each time they consider submitting a bid. Auctions are timed according to the slot of time under consideration, whose length will be 3 minutes. Upon expiration of the auction, the auctioneer will communicate the DE the winner, along with the interest the winner has been competing for. Then, the DE will query the DBE in order to obtain the right spot for advertisement purposes and, once replied, the right advertisement will be shown in the display.

2.1.2 Pervasive Supervision

The Pervasive Advertisement scenario also benefits of a Supervision System (SS) natively integrated in the toolkit. For the same of the application scenario, the pervasive advertisement the SS will show its effectiveness by supervising ACEs in the AE. The way this feature will be integrated in the application scenario is detailed below.

The AE comprises two components that are crucial for the correct functioning of the scenario, namely the Seller and the Auction Centre (AC). We thus demonstrate supervision functions by configuring a supervision pervasion around the interaction of these components, by implementing a mutual heartbeat function. If one of these components fails to send or to receive the heartbeat of the other one, we conclude that an error has been occurred and react with a reset of the malfunctioning component.

A particular problem is to save the current state of the component in question, and to restore it after a reset. Since all state information of an ACE is stored in the global and the execution session objects, and the state of the currently executed plan, state restoration can be performed in a generic way.

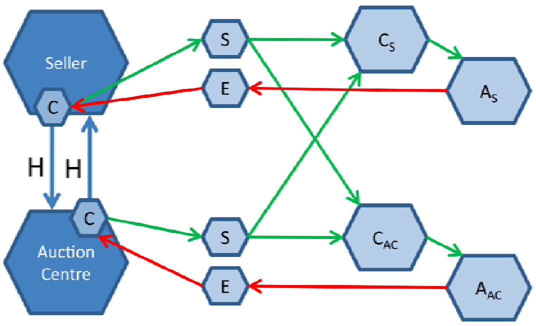

Figure 1: Supervision pervasion for Seller/Auction Centre

The detailed interaction between the system under supervision and the supervisor is the following (compare Figure 1).

Bringing Autonomic Services to Life

1. After contracting the AC, the Seller discovers a supervisor using the GN/GA protocol. Once contracted, it sends its own ACE address and the ACE address of the Auction Center to the supervisor.

2. The supervisor then configures a complete supervision pervasion comprising two control loops, one for validating the AC utilizing heartbeats from the Seller, and the other one for detecting heartbeat interaction initiated by the Auction Centre towards the Seller. Both the Seller and the AC are instrumented by bus[1] and gateway supervision checker objects. The correspondent bus supervision checkers are responsible for emitting a heartbeat message (by deploying it on the internal communication bus) in regular time frames.

3. The heartbeat messages will be received by the gateway supervision checkers of the opposite components. Sensor ACEs on both sides are responsible to request the detected time of the last incoming heartbeat message at their respective components, as well as the sending time of heartbeats in the opposite direction.

4. By computing the difference between both times (which is the job of the correlators) we can infer at which component a fault has been occurred (assessors).

5. If a fault has been detected, the effectors are triggered to issue a reset.

Note that neither the Seller nor the Auction Centre is aware (i.e. have special instrumentation in their self-models or functional repositories) of sending/receiving heartbeat messages.

A number of means will be compiled in a supervision demonstration GUI to make the supervision activities visible during a demonstration.

- A button to configure the supervision pervasion. Pressing this button will cause the Seller to employ a supervisor.
- A button to release the supervisor that will result in the cancellation of the supervision contracts.
- Buttons to force a failure at the Seller and AC, respectively.

2.2 Self-Organized Aggregation For Pervasive Advertisement

In this context, we propose using self-organized aggregation to form dynamic clusters of adverts, so as to be able to generate suitable sequences "on the fly", while guaranteeing some level of randomness (avoiding repetitive loops).

We assume that each advert will be categorized by a set of properties (field values),instead of the abstract "types" or "colors" envisaged in the theoretical work carried out by WP3 [6].

[1] With the term *Bus* here we refer to the bus part of the Manager organ as defined in Deliverable D1.3 "First Prototype Integration".

For instance, one advert could have "sportswear" as its "product category" value and "20 to 40" as its "age group" value. A non-dependent self-aggregation process would be continuously running for every field/property, resulting in each advert surrounding itself with other adverts of identical (if discrete) or nearby (if continuous) values.

Selection of the next advert would then be made by the one currently on display, simply by passing the "token" to one of its neighbours, based on the current characteristics of the audience. For instance, if the dominant criterion is "20 to 40", a "sportswear" advert could pass the token to one of its "age group" neighbours, even if it relates to a "food" item. Symmetrically, if the identified commonality in the audience had been a keen interest in "sportswear", then the next advert could have been picked among "sportswear" neighbours, even if the target customer is in the "under 20" age group.

Obviously, if/when the advert holding the token detects a mismatch between all of its characteristics and those of the audience; it should release the token instead of passing it to one of its neighbours. Note that if this "mismatch transfer" is done at random, it is equivalent to a random walk-based exploration of the whole space of available adverts, which could constitute an easy way of identifying suitable topics without the need for much computation (potentially very useful in a pervasive computing scenario).

The proof-of-concept demonstrator will be self-contained and include 3 separate simulated elements:

- A dynamic data repository, in which adverts self-aggregate according to their various properties
- A dynamic virtual audience comprised of user profiles categorized according to the same properties
- A virtual screen, on which the currently selected advert is displayed

Although the details have not been completely finalized, visualization is expected to make use of symbols to represent abstract categories. For instance, three categories, each with its own set of possible values, could be combined using graphically compatible attributes:

1. Attribute = *shape*, possible values (3) = *circle*, *square*, *triangle*
2. Attribute = *color*, possible values (4) = *red*, *green*, *blue*, *yellow*
3. Attribute = *design*, possible values (2) = *full*, *empty*

So for example, an advert for "books", "20-40", "female" would appear as a *full* (gender) *yellow* (product) *square* (age group), and an advert for "music", "20-40", "male" as *empty* (different gender), *red* (different product), *square* (same age group). To illustrate the match-making process, user profiles will obviously use identical symbols, though some may have more than one value simultaneously (e.g. to reflect an interest in several types of products).

It is anticipated that the visualization will allow for real-time modification of the virtual audience's characteristics, so as to demonstrate that the simulation is not scripted and that the system is truly adaptable.

Time allowing, additional features like the ability to add new adverts to the database to illustrate their seamless integration into the existing overlays may be implemented.

2.3 Authenticated ACE-Based Communication

The demonstration subject of this section focuses on the integration of security aspects in the CASCADAS autonomic toolkit. The contribution of WP4 to the CASCADAS autonomic toolkit is not intended to provide new cryptographic suites, or define new concepts to implement security in Autonomic Communication Elements (ACEs), but rather integrate security functionalities smoothly.

Hence, our aim is to show an ACE capable of exploiting security functions to protect data or to avoid unauthorized entities to access data. This will highlight the use of basic cryptographic principles on top of the ACE conceptual definition. WP4 will consider the ACE toolkit released by WP1 and existing standard cryptographic libraries to create a secure ACE of type B, as defined in [5].

Preliminary work in this direction has been already performed during the definition of the Open-source Security toolkit for CASCADAS. We have analyzed the impact of cryptographic algorithms and hash functions with respect to the heterogeneous nature of the devices. This helps to understand what the performances of these algorithms and functions are with respect to key size, device capabilities and cryptosystem type.

As shown in Deliverable D4.2 [7], security functionalities can be defined in line with the definition of the ACEs provided by WP1. The work targets the Specific Part (Repository Functionality and Self-Model) of the ACE structure. Security functionalities will be part of the "Specific Interface" which contains security functionalities, implemented in the "Specific Feature" by means of cryptographic libraries, which characterize the ACE behaviour. For the sake of clarity we have presented a semantic description of the job the ACE is able to do (GA) and the indispensable and essential actions and conditions to accomplish it (GN), as defined in Deliverable D4.2.

The security principles that will be demonstrated for the demo are authentication, confidentiality and integrity. To comply with the requirement of light ACEs, i.e., simple functionalities for each ACE, we envision the formation of virtual security domains constituted by ACEs under the same administrative control. For the demo, it is sufficient to assume that nodes have a pre-shared key which is used for authentication and confidentiality. Nodes that know the key can join the domain and they can have access to the information. The first part of the demo consists in demonstrating the integration of authentication functionalities in the ACEs. Authentication could be done by using a challenge-response protocol that requires the unauthenticated party to calculate the HMAC of the challenge by using the pre-shared key. The challenge could be a timestamp to avoid possible reply attacks. If the party is authenticated, the node will be admitted in the community. The second part of the demo consists in showing how confidentiality is integrated in the ACE toolkit. A simple visual application is the transfer of an image between two ACEs. The image is encrypted and displayed at the destination. A third ACE that eavesdrops the traffic cannot display the image because it does not know the shared key.

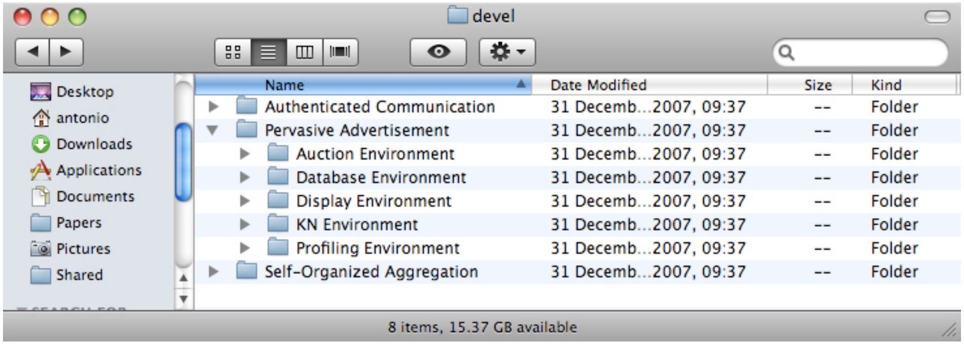

Figure 2: Structure of the source code available on the Subversion repository.

3 Source Code

Deliverable D6.5 "Application Scenario Implementation foresees the delivery the source code realizing all the demonstrations described in previous sections.

Such material can be obtained through the Subversion repository setup for the CASCADAS project as the URL https://cascadas.mik.bme.hu/repositories/cascadas. Specifically, under the trunk/wp6/deliverables/D6.5/devel folder a hierarchy of sub-folders has been created. This is shown in Figure 2 above.

The structure features a sub-folder for each of the demonstrations previously defined and described. In addition, the *Pervasive Advertisement* sub-folder is further divided into as many sub-folders as the environments composing it. Each of the folders will contain the corresponding code, and the structure will be further extended with other sub-folders (for instance with another containing the integrated version of the whole environment) in future.