Designing Reference Architecture for Providing Virtual Home Environment

Tomislav Marenić and Krešimir Mlinarić
Department of Telecommunications
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
E-mail: tomislav.marenic@fer.hr, kresimir.mlinaric@fer.hr

Abstract - **This paper presents a proposal for a reference network centric architecture designed to enable mobility of personalized service environments across different networks and terminals. This concept is addressed in the 3GPP (Third Generation Partnership Project) specifications as Virtual Home Environment (VHE). Main objective of VHE is to enable user centric communications and ubiquitous personalization, thus providing users with consistent usage experience. The proposed architecture deals with various issues raised by VHE requirements, such as: handling different front-end technologies, defining and maintaining personal service profiles, convergence of telecomm networks and Internet etc. A partial implementation of the proposed architecture, based on mobile agents and Open Service Access (OSA), was implemented and tested in a laboratory environment.**

***Keywords - mobile agents, Virtual Home Environment (VHE), personal mobility, service mobility, Open Service Access (OSA), service provisioning, terminal centric, network centr*ic**

I. Introduction

Introduction of 3G networks, such as UMTS (Universal Mobile Telecommunications System) in Europe, will lead to some essential changes in the basic network architecture, which have to be considered when planning future services. The most significant factor is the inevitable convergence of traditional telecommunication networks and Internet into a universal All-IP core network. Users will be able to access that core network using different access networks, some of which are fixed, and some of which are mobile.

The aforementioned convergence will enable development of different services that combine features from both types of networks, as well as access to those services from mobile terminals. It is expected that the combination of these two factors will lead to evolution and expansion of Mobile Internet. In such an environment, personalization will become increasingly important as more and more complex services appear, just like it did during the evolution of the fixed Internet. However, it is expected that separate personalization of each service will not be sufficient to achieve wide-spread adoption of new networks and their related services.

The basic idea of UPT (Universal personal Telecommunication) concept, introduced in [1], is that users are no longer associated to the specific network access point – as in present fixed telecommunication networks, or a specific terminal – as in present mobile telecommunication networks. Instead of concentrating on technology and connecting personalization data to networks, services or terminals, UPT introduces the idea of unique personal profiles tied to a specific user. Such profiles contain information about user's personal set of subscribed services and their settings, as well as other personal data not tied to any specific service.

In conformance with the described user-centric communication concepts, users should be able to use their personal profiles irrespective of the network they are currently attached to or the terminal they are currently using. This concept is called personal mobility, and should not be mistaken for terminal mobility enabled by mobile access networks, which is only a subset of personal mobility. 3GPP (Third Generation Partnership Project) introduced this concept in the UMTS specifications under the name Virtual Home Environment (VHE).

This paper is organized as follows. In section II, a brief introduction into basic VHE terms and ideas is given, as well as a summary of different views of VHE. Section III offers a categorization of different personal mobility models, which is later used throughout the paper. Section IV represents the main contribution of the paper, since it describes the proposed reference architecture for providing VHE support to end users. The largest part of this architecture is placed in the network, because one of the design goals was to reduce terminal requirements to the minimum, so that VHE functionality can be accessed from a wide variety of devices. Section V describes an implementation of the subset of the proposed architecture, which uses mobile agents for dynamic service provisioning. In section VI, related work is presented and compared to the solution proposed in this paper. Finally, in section VII, the achieved results are summarized and plans for future work are given.

II. Virtual Home Environment

This section will present three notions of VHE. First one is from an actual research project, second one from a standardization body, while the third one covers user's perspective of the functionality VHE should offer. Those definitions are not different in essence, but they emphasize different aspects of personal mobility. Because of clarity and completeness, we present them one by one.

It obviously makes no sense to talk about the detailed standardization of the VHE concept, since it is defined as a general concept whose actual implementation depends heavily on the exact environment. This paper will rely mainly on 3GPP terms and definitions, since it is the official standardization body for UMTS. However, some definitions have also been taken from VESPER (Virtual Home Environment for Service Personalization and Roaming Users) [2], a European Union project whose purpose is to offer practical implementations and thus validate VHE concept. VESPER was chosen because it currently the most significant VHE project that we know of, with several ongoing research activities. A more detailed overview of VESPER will be given in section VI.

A. VESPER

In the context of VESPER, the following are defined as the basic functionalities required of VHE:

- Personalization of Service Environment - Personal Service Environment (PSE) is a concept for describing how the user wishes to control and interact with his/her communication services. PSE consists of one or more user profiles, each connected to a certain context. Each of these profiles contains two basic types of information: a list of active services and their service-specific preferences, and general preferences – e.g. related to user interface, personalized data etc. User has the freedom to activate the appropriate profile, which enables management of communications according to different situations or needs, e.g. being at work, in the car or at home.
- Adaptation of Service Environment - the customized environment follows the users while they are roaming within different networks and using different terminals. Since different terminals have different capabilities (e.g. some do offer the audio environment, some do not) and so do different networks (e.g. bandwidth), it is necessary to use the adaptation according to the terminal and network capabilities. Such service adaptation should comply with user preferences.
- Service Portability – users are offered the same service experience in a visited network as in his home network.

B. 3GPP

3GPP specification [3] gives a compact definition of VHE, which comprises most of the VESPER requirements. VHE is defined as a concept for PSE portability across network boundaries and between terminals. The concept of the VHE is such that users are consistently presented with the same personalized features, user interface customization and services in whatever network and whatever terminal, within their inherent technical limitations.

3GPP specifications go further, and define not only VHE requirements and functionalities, but also a flexible environment which should support VHE provisioning in real-life networks. Fundamental elements of such environments are homogenous service execution platforms, which should enable simple portability of services between different networks and terminals.

Mobile Execution Environment (MExE) specification [4] defines a standardized execution environment in the terminal. In MExE specifications, terminals are categorized by giving them different MExE classmarks. Classmark 1 corresponds to WAP devices, classmarks 2 and 3 to Personal Java and J2ME enabled devices, respectively, while classmark 4 represents terminals that support Microsoft .NET technology.

Open Service Access (OSA) specification [5] defines an open, standardized interface towards network capabilities, thus providing standardized and homogenous execution environment for network centric services. Network heterogeneity is hidden behind a unified API (Application Programming Interface), which is accessed through CORBA (Common Object Request Broker Architecture) middleware [6]. CORBA enables additional features, such as programming language independence, location transparency, naming service etc. Fig. 1 shows basic OSA architecture.

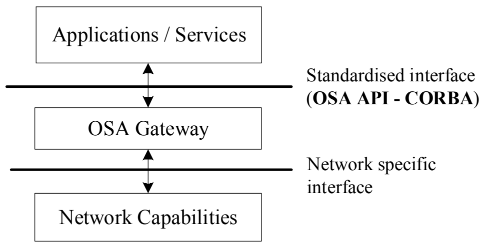

Figure 1. OSA Architecture

Implementation of telecommunication services today is highly dependable on details of hardware and software configuration of the exact environment where they are meant to be deployed. Services designed to work with one vendor's equipment will probably not work on any other equipment. Even when dealing with a single vendor, sometimes services works only on a specific version of that vendor's software. It is impossible to fulfill VHE requirements in such an environment, because services designed to work in one network probably can't function anywhere else. OSA gateways should map platform-specifics into a unified API, thus creating a standardized execution environment and allowing simple portability of service logic from one network to another.

Additionally, programming with an API is a lot easier then programming with a protocol, such as INAP [7], which is used in today's telecomm networks. Therefore, OSA will enable the existing army of Internet developers to start developing telecommunication services quickly. The result will be a possibility for rapid integration of Internet technologies and telecommunication networks' capabilities. OSA also enables access to network resources for 3rd party service providers, which do not necessarily have to be related with the network operator. This is possible because of OSA security features, which include mutual authentication between the application and the OSA framework, based on digital signatures, and thus enable opening of network resources to independent service providers.

C. User's view of VHE

From the user's perspective, VHE is a concept for providing consistency in use of communication services. The main objective of VHE is to enable keeping the same "look and feel" when switching between different networks and terminals. Fig. 2 (taken from [3]) gives a logical VHE role model from the user's viewpoint.

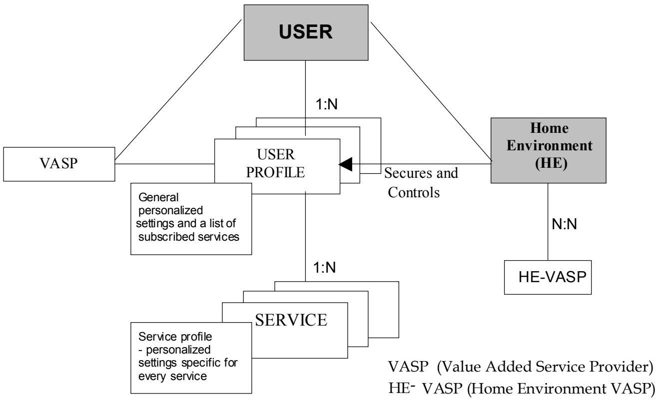

Figure 2. Logical VHE Role Model (user's view)

It should be noted that user's view of VHE contains no notion of home or visited network, or the terminal currently used. This is fully conformant to the fundamental VHE idea, according to which the user does not want to worry about technical details of the resources he is currently using, but is only interested in his personalized environment.

Home Environment (HE) is the entity responsible for overall VHE provision and control of the PSE of its subscribers. It should not be confused with the Home Network (HN), which is a term describing the network operator with which the user has a contract about provisioning of basic telecommunication services. HE can, but does not necessarily have to be under the direct control of HN operator.

Value Added Service Provider (VASP) is a new role, enabled by OSA concept. Basically, it can be viewed as a third party service provider, since OSA enables network operators to offer their network resources to independent contractors.

III. SPLITTING UP PERSONAL MOBILITY

This section will discuss different cases of personal mobility. First subsection will try to offer a basic classification based on differences between implementation concerns raised by each of these cases. Second subsection will introduce and discuss different mobility models based on that classification.

A. Categorization

In conformance with VHE definition introduced in the previous section, two basic types of personal roaming can be identified. The first one is roaming between different networks, which comes down to changing the location, i.e. the operator whose network is being used. This type of roaming was introduced in GSM (Global System for Mobile Communications) networks, but in context of terminal mobility. In the rest of this paper, this type will be called network roaming (NR). The second one is switching between different terminals, which usually means a change in working conditions, because of differences in terminal capabilities. This type will be referred to as terminal roaming (TR). Fig. 3 shows basic types of personal mobility.

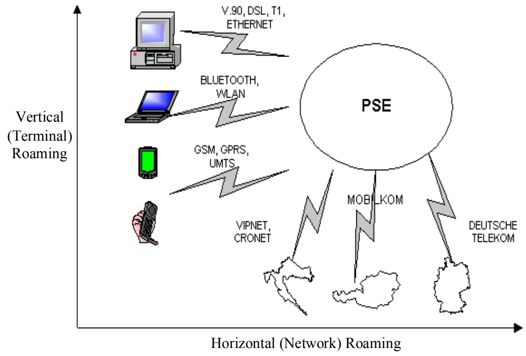

Figure 3. Types of Personal Mobility

Categorization of services will be based on the place of service execution, which turned out to be the most significant factor for design of personal mobility models.

Network Centric (NC) services are executed on some server in the network, while the user is presented only with results of processing his/her input (if any). Typically, user interface is implemented using widely accepted technologies, such as HTML (Hypertext Markup Language) or WML (Wireless Markup Language). That way, the user terminal does not need to support any advanced technologies or proprietary solutions possibly used in service logic implementation. Further on, NC services must be divided in two additional subclasses, regarding their dependence on the network through which the user is connected. The first subclass refers to services strongly dependent of the underlying network, which will be called NC1 services. Typical examples are services that use telecommunication network capabilities, such as different call control services, location based services etc. On the other hand, Internet based services (NC2), typically different kinds of web services and active server-side components (ASP, servlets etc.), are usually independent of the infrastructure, and thus can be used regardless of the user's current network.

The main reason for this additional categorization is the fact that NC1 services sometimes must be migrated to the Serving Network (SN - the network through which the user is currently connected) in order to make any sense. Location based services make a good example, as they can't operate from a HN when user is roaming, because they are inherently related to user's location. Another good example are services related to outgoing calls. Case study related to such a service will be described in section V. Additionally, NC1 services are sometimes tied to a specific type of terminal, and it makes no sense to use them from different terminals. Again, location based services make a good example, since they are inherently tied to mobile terminals, and can not be used from a PC. On the other hand, NC2 services are mostly independent of the user's current location or the used terminal, except for some adjustments of user interface to terminal capabilities.

Terminal Centric (TC) services are executed within the user's current terminal. In order to be suitable for dynamic provisioning on diverse terminals, TC services should be implemented as independent software components, which don't require any special installation procedures. Java applications make a good example of such TC services. Their logic is typically contained inside .jar files, and simple downloading of those is needed in order to use the service.

B. Mobility models

NC2-NR - The only thing required is that the currently visited network supports basic capabilities (e.g. bandwidth) for interaction with the user. As previously stated, service mobility is possible only within inherent limitations of network or terminal, so if the visited network does not support minimum requirements user should be informed about it in a straightforward way.

TC-NR - If the TC service interacts with the network, it is possible that the visited network does not support everything the service expects from it, e.g. available bandwidth or QoS (Quality of Service) negotiation mechanisms. In that case, service could try to adapt to different network conditions, or simply inform the user that it can not operate in the current network. This is another case of inherent limitations, which sometimes can not be solved by interventions in service logic.

NC1-TR - As described earlier, some NC1 services make sense only when related to certain terminals. If this is not the case, NC1 service should support different user interfaces, adjusted to the currently used terminal and its presentation capabilities. Such adjustment must be based on a terminal profile, describing its hardware and software capabilities. More information on representation of terminal profiles will be given in section IV.

NC2-TR - The same rules apply as for the NC1-TR combination, but without additional restrictions on specific terminal types.

NC1-NR - There are two basic principles that can be used for this combination.

a) SN is used only as a proxy, while the service logic is executed in the user's HN – shown on Fig. 4

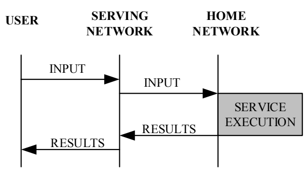

Figure 4. Service execution in the HN

b) Service logic is transferred to SN and executed there – shown on Fig. 5

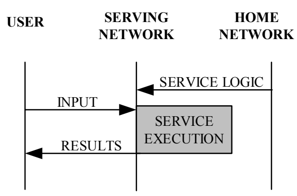

Figure 5. Service execution in the SN

Both of these approaches will be enabled by OSA, once it is introduced in real–life networks. OSA is based on CORBA, which supports location transparency, so it is possible for the service to be executed in the HN and have the same effect as if it was in the SN. Further on, since OSA offers a homogenous execution environment, it is possible to simply transfer and dynamically deploy services in the SN, regardless of differences in hardware and/or software configuration between HN and SN

Which of these approaches is used depends on specific service requirements, but must remain transparent to the end user. Primary issues that have to be considered are related to QoS, because long delays introduced by physical distance are simply not acceptable for some services. Case study described in section V will deal with such an implementation of NC1-NR service mobility model, in which service logic is transferred to the SN.

TC-TR - This combination can also be divided into two subclasses.

a) If the visited terminal is capable of running the service, but does not have service logic stored locally, it can be downloaded (manually or automatically) from the Internet. Whether some service can be executed on some terminal or not can be decided by comparing the terminal profile, which describes terminal capabilities, and service profile, which describes service requirements. More information on terminal profiles will be given in section IV. Paper [8] describes an implementation of this approach. There can also be several service implementations, and which one of those is downloaded depends on the capabilities of the current terminal.
b) When the visited terminal does not support any of the available TC versions, service execution can be moved to the network. Essentially, it comes down to creating a NC2 version of original TC application. As previously mentioned, this version will typically communicate with the user using widely accepted technologies, such as HTML or WML, while more complex technologies stay in the network. Paper [9] describes an implementation based on this approach.

IV. ARCHITECTURE DESCRIPTION

This section will define a reference architecture which covers all of the described functionality VHE should offer to end users. It should be viewed as a framework that enables concrete implementations of different VHE aspects. Section V will describe one such case study, which presents a subset of the proposed architecture.

According to the requirements defined in the previous sections, three key principles have been established and followed during design phase of this architecture:

- The proposed architecture shall enable personal mobility across a large number of different terminals (PCs, mobile and fixed phones, hand-held devices, laptops…) and access technologies (UMTS, GSM/GPRS, Bluetooth, WLAN…)
- Terminal requirements, regarding its capabilities and technologies it must support, should be minimal

- The solution shall be in line with current 3GPP standards
- Security issues will be addressed in an appropriate manner

Fig. 6 shows the proposed architecture. Having in mind 3-layer network architecture, the proposed solution mostly affects the application layer, while its smaller part, responsible for possible QoS adaptations in cooperation with the telecommunication network, is part of the control layer.

As depicted in Fig. 6, the network part of the framework for providing VHE logically consists of a home network (HN) and a serving network (SN). HN is the network operator with which user has a subscription, while SN is the network that the user is currently connected through, i.e. a network on whose coverage area the user is currently located (often called visited or foreign network). Since it can be supposed that every SN is in the same time someone's HN, HN and SN are almost identical.

The network part must be capable of dealing with HTTP and WAP requests, since these are the two most widely accepted technologies. Because of their inherent characteristics, such as autonomy, reactivity, pro-activity, social ability and mobility [10], it was evaluated that mobile agents present a very promising solution for dynamic service provisioning. This is why the proposed architecture provides support for agent technology as well.

Application servers (APS) must be able to support dynamic deployment of new services, since dynamic installation is a prerequisite for enabling service mobility for the roaming users.

VHE HN and VHE SN are the entities that perform the tasks of exchanging the information, contents and services related to VHE provisioning for the roaming users. It should be noted that those two are not standardized by 3GPP, since they cover the functionalities declared to be out of scope of 3GPP specifications. Each HN has to maintain information that specifies details about VHE SN settings in all the networks that HN has a signed roaming contract with. This is a rather reasonable request, since network operators have to sign an offline roaming contract anyway, so they can exchange both technical and legal information. That way VHE HN and VHE SN can use predefined settings to manipulate services and user profiles across heterogeneous networks and terminals

Depending on technologies used in each of the networks, VHE HN can be significantly different from VHE SN, but they must support some common communication and security mechanisms in order to establish VHE between their belonging networks. Therefore, they will typically have to support a broad range of technologies, such as mobile agents, RMI (Remote Method Invocation), CORBA, SOAP (Simple Object Access Protocol) etc.

Although VHE HN and VHE SN are not standardized by 3GPP, this does not present a problem regarding service portability, because services don't directly use those entities are don't have to be aware of them.

For the purpose of keeping the user profiles, which are necessary for service personalization, 3GPP specifies a special network entity that is referred to as Home Subscriber Server (HSS). HSS is the equivalent of Home Location Register (HLR), which has been used in GSM, with some added functionalities, such as keeping and managing user profiles. Regarding user profiles, VHE must enable different network entities to access those profiles independent of their location or technology they are using. It is also very important to define profiles control access mechanisms, and to make it possible to recover user profiles at any time. Since the service delivery directly depends on the active profile, VHE shall enable the user to manage his/her profiles (e.g. modify, activate, deactivate, etc.). The user can select his/her profile either statically (the user manually chooses the profile from the list of available profiles) or dynamically (the active profile is selected automatically based upon some predefined criteria - e.g. time of day, location, terminal used, etc.).

To make the profiles globally accessible, they have to be online at all times – which means they have to be stored in the network. On the other hand, it makes no sense to reach parts of profiles related to some terminal settings from HE each time they are used. The solution of that problem is to store parts of profiles in the terminal, while the whole profile is always stored in HE, which makes it accessible at any time, even in the case of terminal loss or damage. Whenever it is supposed that profile has been changed, HE must be able to perform its synchronization. Logical model of the user profiles distribution is described in [3].

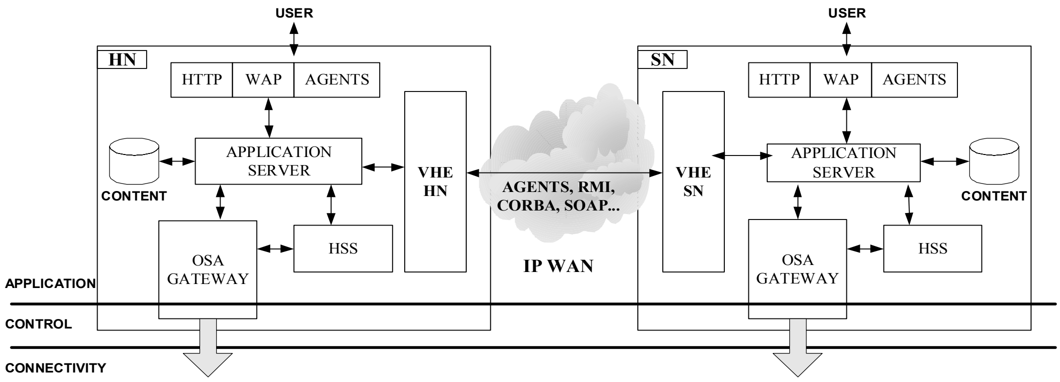

Figure 6. Network part of the architecture

As already mentioned, one of the design goals was to reduce terminal requirements to the minimum, and thus enable usage of the proposed architecture across a wide variety of devices. The terminal part of the architecture is shown on Fig. 7.

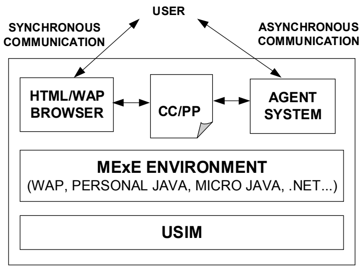

Figure 7. Terminal part of the architecture

It was assumed that every terminal must have a HTML or WAP browser. Although the emphasis of this work is on use of agent technologies, an installed agent platform is not a mandatory requirement for terminals. However, terminals with agent support will be able to deal with more advanced solutions. Opposite to WAP and HTTP, that enable strictly synchronous communication with constant user interaction, agents introduce possibility of asynchronous communication as well, which allows increased autonomy.

The architecture also supposes that every terminal supports MExE specification, which is one of the generic toolkits defined by 3GPP. MExE provides a standardized execution environment in a terminal, where a term terminal refers to a mobile device together with the USIM (Universal Subscriber Identity Module).

As described in section III, terminals must also contain information about their hardware and software capabilities. CC/PP (Composite Capabilities/Preference Profiles) [11] is an RDF (Resource Description Framework) based framework, introduced by W3C as a standard for describing device capabilities. RDF is a language designed by W3C (World Wide Web Consortium) for describing the metadata (data about the data). Although RDF is a general concept, not tied to any specific format, it is practically always encoded in XML (eXtensible Markup Language). CC/PP is designed to work with a wide variety of web-enabled devices, from personal computers, laptops, WAP phones, handheld computers, to specialized browsers for users with disabilities. CC/PP profile can contain various information, such as screen size, color depth, processor speed, amount of memory, available audio and video codecs etc. Paper [8] describes a case study of using CC/PP profiles to filter supported services on an unknown terminal, and thus adapt user profile according to terminal capabilities.

V. Case study – Dynamic provisioning of OSA services based on mobile agents

This section describes a case study of the NC1-NR service mobility model. Service which will be moved around is Hot Line – service that enables automatic connection establishing between two phone numbers as soon as the originating phone is off the hook. The first subsection will give a brief overview of Hot Line service, while the second one will describe the proposed solution for dynamic provisioning of Hot Line to roaming users, and its laboratory implementation.

A. Description of Hot Line service

As previously mentioned, Hot Line is a service that automatically connects two predefined numbers as soon as one of them goes off hook, i.e. when the 'Yes' button is pressed on the cell phone. Implementation described here is unidirectional (if there is a hotline A→B it does not imply B→A).

The choice of service to be used was based on two main factors. Firstly, it had to be related to call control, because laboratory implementation uses Ericsson's OSA/Parlay simulator [12], which supports only the basic features of OSA, such as call control and generic user interaction. Additionally, it had to be related to outgoing calls, which makes it suitable for dynamic migration to the visited network. Outgoing calls always go through the visited network, and if a service wants to control them it must be present there. Although it would be possible to transfer call control to services residing in the HN even when users are in roaming, this approach could introduce long delays due to large physical distance, network overload etc. On the other hand, incoming calls always go through the HN anyway, so it makes no sense to transfer services controlling them to the visited network.

Hot Line service is implemented in Java, and connects to the telecommunication network through an OSA gateway. This service represents a good example of how easy it is to develop telecommunication services with OSA, because it was implemented in a week, in only 4 classes (around 650 lines of code), by an Internet developer with no experience in the area of telecommunication services. Another important feature is that it was developed and tested using a simulator running on an ordinary PC, without access to expensive network equipment. These features of OSA, combined with the possibility of open access for independent service providers, offer the prospect for a rapid growth of service market.

B. Dynamic provisioning of Hot Line

The basic scenario this example tries to deal with is the following: user which normally has Hot Line service activated roams to a different network (NR). According to VHE requirements, users should have access to their services regardless of the currently used terminal or network. Therefore, Hot Line must also be activated in the SN. Because of the reasons described in the previous subsection, the best way to achieve this is to dynamically deploy Hot Line in the SN. However, dynamic service provisioning raises some additional issues.

Entity that carries out installation and starting of the service in the SN must know file system details (installation directory, which disk has available space etc.) and have appropriate security permissions. Because of that, it would be fairly complex to implement an architecture in which components from the HN perform remote installations in the SN. Although there are available solutions for performing remote installations, all of them are designed for human administrators, which take over control of the remote computer and perform the installations. Those solutions are therefore not appropriate for dynamic service provisioning, since it requires autonomous installations without human interaction. For those reasons, it is easiest for those operations to be carried out by an entity residing on the VHE server in the SN.

On the other hand, the entity that installs and starts the software has to know many details about the specific software, e.g. which configuration files are to be changed according to local specifics, which initial parameters must be passed to the service etc. It is unfeasible for a single component in the SN to know such details about all possible services from all networks it has a signed roaming contract with.

The described problems are the main reason why mobile agents – autonomous software entities with the ability to migrate between network nodes, were chosen for implementation of dynamic service provisioning. The basic idea is to have a mobile agent originating from the HN, which knows all the service-specific information, but migrates to the SN to perform installation and starting of the service. That way it is possible to solve both of the abovementioned problem groups. Implementation described here will be based on such a mobile agent, which will be used as a carrier of the service logic to the SN. Grasshopper agent system [13] was used, primarily because of its security features.

Fig. 8 shows an AUML (Agent UML) deployment diagram of the laboratory implementation. AUML is an extended version of UML (Unified Modeling Language) used for modeling agent software. The only AUML-specific used in Fig. 3 is the `<<mobile>>` stereotype, which represents agent's migration from one node to another. Additional information about AUML can be found on [14].

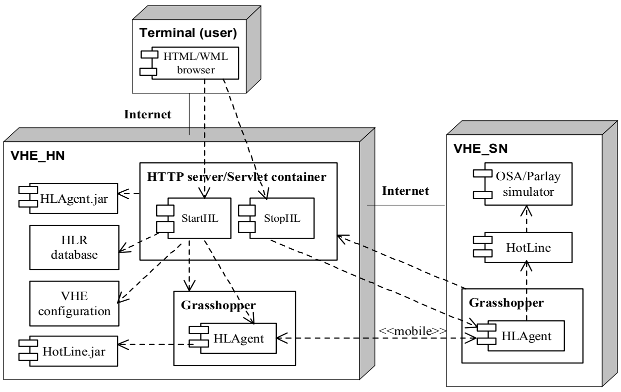

Figure 8. Implementation Deployment Diagram

HLAgent – a mobile agent that installs and starts Hot Line in the SN.

HTTP server/Servlet container – typically Apache Tomcat [15], contains servlets through which user communicates with the system, and serves as a code server for HLAgent.

HLAgent.jar – contains classes for the HLAgent, which Grasshopper agency from the SN loads through the HTTP server.

StartHL, StopHL – servlets through which user starts/stops HotLine while in roaming

HLR database – when user initiates the request, he/she has to specify between which numbers Hot Line will be established. This database simulates HLR (Home Location Register), and contains current location (i.e. the currently visited network) of the phone with the specified number.

VHE configuration – contains information about all the networks that this HN has a roaming contract with. In particular, it specifies parameters of the OSA gateway and Grasshopper agent system in the SN. As previously explained, it is reasonable to expect that network operators exchange such information when signing offline roaming contracts.

HotLine.jar – contains Hot Line service logic. HLAgent loads this file and takes it to the SN.

In real-life implementations there would be no need for user requests to trigger VHE provisioning. OSA Mobility API [16] provides `IpTriggeredUserLocation` interface, which application programmers can use to request user location reports that are triggered by location change. That way an application residing in the HN could register to receive location updates for all users, and automatically initiate VHE provisioning or VHE termination (based on user profiles) as soon as users switch networks. However, since the OSA simulator used for laboratory implementation doesn't support OSA Mobility API, users have to explicitly request VHE provisioning and VHE termination. Java servlets were used for user interaction, because they make it rather simple to communicate with Grasshopper (because it is also Java-based), and provide both HTML and WML user interfaces.

Fig. 9 shows the AUML sequence diagram describing what happens when some roaming user initiates the procedure for dynamic deployment of Hot Line.

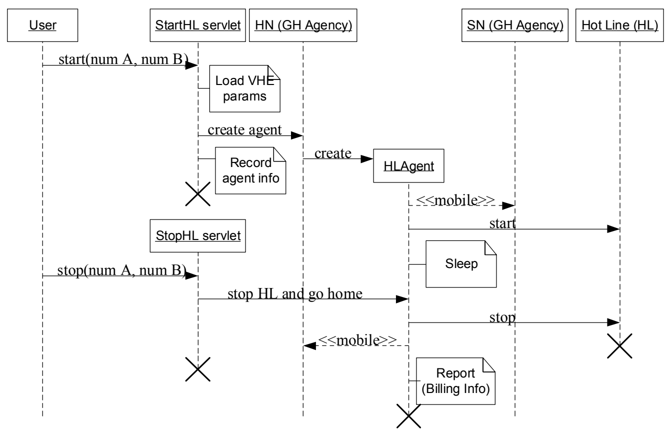

Figure 9. Hot Line provisioning sequence diagram

After the user initiates the request for VHE provisioning, StartHL servlet loads the necessary parameters (the exact address of Grasshopper agency and CORBA parameters of OSA gateway in the SN) from VHE configuration database, creates an instance of HLAgent and gives it the parameters. After creating the HLAgent, StartHL writes its agent identifier and the exact address of the Grasshopper agency at the SN, to be used for later termination of Hot Line. HLAgent loads HotLine.jar into an array of bytes and migrates to the SN. Once in the SN, it uses a customized class loader to create a running instance of Hot Line directly from the memory. That way, service logic is at no time stored at the local disk in the SN, which is especially important if that service is not meant to be publicly available.

After Hot Line is started, HLAgent goes to sleep and waits for a termination message from the HN. When user wants to terminate the service, he/she does it through StopHL servlet, which reads the information written by StartHL, and based on it sends a message to HLAgent. This message wakes HLAgent from sleep, after which it terminates Hot Line and returns home. HLAgent keeps track of time during which Hot Line was active, since it can be used as billing information.

The laboratory implementation uses Grasshopper security features, so all network communication is protected by SSL (Secure Socket Layer), and all agents are digitally signed. Based on their signatures, only those agents that have appropriate security permissions (i.e. come from known networks) can enter the Grasshopper agency at the SN and install the services there.

VI. Related Work

Two most significant VHE research project that we are aware of are CAMELEON (Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks) [17] and previously mentioned VESPER [2]. Both of these are complex projects, that have elaborated general VHE architecture and offered several trial implementations of the proposed concepts. However, neither of them is pragmatic enough to be considered for real-life implementations.

CAMELEON focuses on use of mobile agents, so the developed applications rely on the assumption that all terminals can execute agents, which is far from truth. Further on, CAMELEON doesn't consider security issues at all, and their introduction would make many of the proposed solutions difficult, if not impossible to implement.

VESPER proposes a solution in which VHE-specific components in both terminal and network are introduced. Although these components allow advanced functionalities that wouldn't be possible otherwise, applicability of VESPER architecture is reduced to a very limited number of terminals and networks supporting those special requirements. This is acceptable for a research project, in which testing is conducted in a controlled environment, but it raises a question whether those concepts can be mapped to an open environment without those additional components.

In our opinion, "ubiquitous" is a keyword describing VHE. We argue that it is better to offer basic functionalities available everywhere, than advanced services available only in a small amount of non-standardized environments. In line with that attitude, we aim to provide support for today's customary technologies, although the emphasis of our work is on mobile agents. Additionally, to ensure service portability in open environments, we use only toolkits standardized by 3GPP (MExE and OSA). Further on, security requirements have been considered from the earliest design phases, and the solutions based on de-facto Internet standards (SSL and digital signatures) have been proposed. All of these factors combined make the proposed architecture suitable for implementation in real-life conditions, which is a quality that none of the other VHE projects offers.

VII. Conclusion And Future Work

This paper proposed a reference architecture for providing VHE support to roaming users. A subset of the proposed architecture, related to dynamic service provisioning, was implemented and tested in a laboratory environment. It was reasoned that agent technologies are particularly suitable for service provisioning problems, so the implemented case study relies on mobile agents. Security issues have been addressed by using Grasshopper agent platform security features. All information exchanged over the network is protected by SSL, whereas authentication and authorization of mobile agents is based on digital signatures.

Besides the central problem of personal mobility, this paper also addressed the development of services using standardized telecommunication network APIs, such as OSA. One example of an OSA service was developed, which served as the main part of the case study. Based on experiences during development of that service, the benefits of OSA and the new market opportunities it offers were evaluated.

Future work will concentrate on further implementations, which should cover the whole scope of the proposed architecture, and finally offer a complete solution for VHE provisioning. Because of the qualities shown in this case study, the emphasis of future work will be on use of mobile agents. However, today's customary technologies, such as HTTP or WAP, won't be neglected, since it is anticipated that use of wide spread technologies for user interfaces is a prerequisite for real-life use of the proposed solution. The main line of work will deal with moving the focus from single agents to multiple cooperating agents. The final goal is a multi-agent system for fully autonomous VHE provisioning, in which personal agents represent their owners.

References

[1] ITU-T Recommendation F.850, Principles of UPT

[2] VESPER web site: http://vesper.intranet.gr

[3] 3GPP TS 22.121: Service aspects; The Virtual Home Environment

[4] 3GPP TS 23.057: Mobile Execution Environment (MExE)

[5] 3GPP TS 23.127: Virtual Home Environment/Open service Access

[6] CORBA web site: http://www.corba.org

[7] ITU-T Recommendation Q.1218, Interface Recommendation for Intelligent Network CS-1

[8] Jasmina Šašak, Tomislav Marenić, "Establishing Virtual Home Environment Across Terminals with Diverse Capabilities", accepted for publication at ConTEL 2003, 11-13 June, Zagreb, Croatia

[9] Tomislav Marenić, "Service Mobility Models for Virtual Home Environment: Case Study of a Mobile Agent Based Service", MIPRO 2002, Opatija, Croatia

[10] Brenner W., Zarnekow R. & Wittig H., 1998, Intelligent Software Agents, Springer-Verlag

[11] CC/PP working group: http://www.w3.org/Mobile/CCPP

[12] Ericsson OSA/Parlay Simulator v0.61b, User's Guide

[13] Grasshopper web site: http://www.grasshopper.de

[14] Agent UML web site: http://www.auml.org

[15] Apache software foundation: http://www.apache.org

[16] 3GPP TS 29.198: OSA API; Part 6 – Mobility

[17] CAMELEON web site: http://www.comnets.rwth-aachen.de/~cameleon/