E2E Mobility Management in Ubiquitous and Ambient Networks Context

Rachad Nassar and Noëmie Simoni

TELECOM ParisTech, LTCI, UMR 5141 CNRS
46, rue Barrault F 75634 Paris Cedex 13, France
{rachad.nassar,noemie.simoni}@telecom-paristech.fr

Abstract. With the rapid evolution of the Next Generation Networks (NGN) concept in the communication industry, mobility requirements have become major challenges that raise the complexity of resource management. In order to maintain continuous end-users sessions, an End-to-End (E2E) mobility management solution must be applied. For this purpose, we introduce the Community of Interest (CoI) concept, also called Virtual Community (VXC) concept, at each NGN layer (service, network, equipment, user). In this paper, we propose to gather elements into VXCs according to two main interests: first, "Ubiquity" which refers to elements that have equivalent QoS and functionality, and second, "Location" which refers to elements that have the same mapping address. These gatherings guarantee the user-centric mobile approach by taking into consideration end-users needs, their demanded QoS and real-time changes in their ambient environments. At he end of this paper, a feasibility study that is based on JXTA (Juxtapose) and EJB (Enterprise Java Bean) technologies is provided.

Keywords: E2E Mobility Management, Community of Interest, Virtual Community, Ubiquity, Location, Context-awareness.

1 Introduction

Nowadays, with the emergence of successive network generations and technologies (2G, 3G, 4G, etc.) and the convergence towards IP-based core networks, the Next Generation Networks (NGN) concept has seen a rapid evolution. Thus, new complex challenges arise. The ubiquity, fix/mobile convergence, and ambient Internet aspects are some of the problems that need to be answered. Such an evolution also induces more complex environments and contexts for end-users, especially by stimulating the growth of their nomadic behavior. Normally, end-users claim the access to any service without any temporal, geographical, technical or economical barriers. They aim to establish a dynamic session that best suits their preferences and their QoS requirements. According to this user-centric context, end-users want to preserve their session while continuously executing their services whatever the access-network, the core-network, equipment and service providers are. All the previously mentioned issues impose the need of an End-to-End (E2E) mobility management that overcomes the mobility challenges on the four NGN visibility levels:

P. Sénac, M. Ott, and A. Seneviratne (Eds.): MobiQuitous 2010, LNICST 73, pp. 312–323, 2012.

- **Terminal Mobility:** represents the ability of a terminal to switch between access networks while preserving the execution of the same subscribed services list.
- **Network Mobility:** represents the ability to move a gateway router without interrupting the opened session.
- **User Mobility:** represents the ability of an end-user to switch between terminals during the access to his/her services.
- **Service Mobility:** represents the ability to execute the subscribed services even if the end-user is connected to foreign service providers.

Our main objective is to guarantee this E2E mobility management while taking into consideration the end-users preferences and QoS requirements, without neglecting the real-time changes in their ambient environments. In order to ensure an efficient E2E management, mobility solutions should simplify the heterogeneous end-users contexts through the regrouping and partitioning of their ambient resources. Therefore, the way of gathering ambient resources becomes the first challenge to overcome. Once the gathering is insured, managing the resources becomes another challenge. Many questions need to be answered: Based on which criteria this gathering should be performed? Can it satisfy the user-centric flexible context? How to manage the gathered resources so that a seamless and dynamic E2E mobility management is guaranteed?

The remainder of this paper is organized as follows. Section 2 discusses existing mobility solutions. In Section 3, we detail our E2E mobility management solution that answers the aforementioned questions. It is based on gathering NGN resources into virtual communities according to a novel concept, namely the Community of Interest (CoI) concept. Section 4 shows the feasibility of the propositions by providing implementations using JXTA (Juxtapose) and EJBs (Enterprise Java Beans) technologies. Finally, conclusion and future perspectives are presented in Section 5.

2 Overview on Existing Mobility Solutions

In the NGN context, mobility management has become one of the main issues of modern communication. In the last decade, several studies evolved to overcome this challenge. In this section, we provide an overview of existing mobility solutions.

- **Handover:** It is a technology for managing terminal mobility at the transmission layer. It maintains the continuity of communication, either when users change the connectivity type (vertical Handover), or when they switch among attachment points in the cellular system (horizontal Handover). For this purpose, several Handover technologies were proposed [1], such as Soft Handover, Softer Handover, and Hard Handover.
- **Mobile IP:** It is introduced at the network layer in order to allow end-users to move through various access networks with their applications in progress, and while maintaining a permanent IP address [2]. For each visited network, the end-user acquires a temporary IP address called Care of Address (CoA). The latter is different from the permanently obtained address called Home Address (HoA).

- **IMS/SIP:** IP Multimedia Subsystem (IMS) is an architectural platform used for the delivery of IP multimedia in mobile networks. The signaling protocol used by IMS to communicate with the application layer is the Session Initialization Protocol (SIP) [3]. This protocol mainly considers session mobility in the case where end-users change their terminals.
- **HIP:** Host Identity Protocol (HIP) separates the two IP address roles: localization and identification [4]. This separation is guaranteed by introducing a new layer between the network and application layers of the TCP/IP model, as well as a new identifier called Host Identity. Hence, user displacement becomes transparent for applications since they only see the end-user's permanent Host Identity instead of his/her variable IP Address.

Having briefly mentioned existing mobility solutions, we note that each solution is limited to its functional layer and none of them can answer all the aforementioned NGN mobility types. Therefore, a new solution with a global mobility vision is needed. In this paper, we propose an innovative mobility management approach that guarantees a continuous E2E mobile session for each end-user. We should mention that the originality of our solution is in its architectural model that is based on Peer-to-Peer (P2P) auto-managed components. Effectively, on the performance level, our novel gathering model is not comparable to other mobility solutions since it manages a larger scope of the problem. It supports all layers and mobility types in order to reach a continuous E2E session.

3 Proposition

Mobility is a main challenging issue for NGN and a main cause of instability in any NGN environment. To overcome this problem and to guarantee continuous end-users sessions, we propose in this paper a dynamic seamless E2E mobility management solution which is based on regrouping elements into Virtual Communities (VXC, X: Service, Connectivity, Equipment or User). This novel concept is also called the Community of Interest (CoI) concept. Each CoI defines a group of elements that share a common interest. Community members are auto-managed and can exchange information with each others in pursuit of shared goals. The CoI creation and management concepts are applied at four layers (user, equipment, network and service) and can consequently handle the four aforementioned mobility types (user mobility, terminal mobility, network mobility and service mobility). In order to take into consideration the user-centric approach, the sessions E2E QoS and the real-time changes in end-users ambient environments, we propose in the next sub-sections to gather elements into communities according to two main interests: "Ubiquity" (Section 3.1) and "Location" (Section 3.2). In the last sub-section (Section 3.3), we combine these two community types in order to create ubiquity and location based VXCs. Therefore, throughout end-users movements, we anticipate their demands by creating the communities that represent their ambient environments and that respond to their preferences.

3.1 Ubiquity-Based Virtual Community: VXCU

One of the main aspects of our user-centric approach is to guarantee all services anytime, anyhow and anywhere. To reach this goal, ubiquitous elements should be deployed on each NGN layer. For this purpose, we propose to apply "Ubiquity" as the first "Interest" and to gather ubiquitous elements into Ubiquity-based Virtual Communities (VXCUs, U: Ubiquity). In our context, "Ubiquity" represents elements having equivalent QoS and the same functionality. The VXCU concept is applied on the service, network, equipment and user layers in order to guarantee an E2E mobility management. Normally, in a mobile context, some components in end-user's session may not continue to fulfill their SLA or the end-user's QoS requirements. To solve these mobility problems, we dynamically replace the current component by a ubiquitous counterpart that belongs to the same VXCU. Therefore, we seamlessly guarantee a continuous end-user's session while maintaining the end-user's E2E demanded QoS.

Actually, the VXCU model is based on a Peer-to-Peer (P2P) self community management process. Each component is auto-managed and acts as a peer. In the following, we divide the communities creation and management into two phases, namely the deployment and exploitation phases.

In the deployment phase, each component notifies other peers about his functionality and his current QoS. In the first case, the peers that support the same functionality and guarantee an equivalent QoS reply to the notification. These peers are members of a specific VXCU. Consequently, the component joins this VXCU and puts the Community ID into its real-time profile. The latter is the subject of another working group. It instantly describes each NGN component. It is a real-time representation of this component. In the second case, the component did not receive any reply to his notification. Thus, it considers that there is no existing community that answers its QoS and its functionality. Hence, it creates a new community that adopts the component's own characteristics (QoS and functionality). In addition, the newly created VXCU contains a new Community ID that is put into the component's real-time profile.

After explaining what happens in the deployment phase and how we can create and update these ubiquity based VXCs (VXCUs), we need to manage these communities in the exploitation phase. In this phase, all NGN elements confront instant QoS variation. In order to dynamically manage the communities that contain these elements, we develop a QoS-Agent in each NGN component. This QoS-Agent permits the auto-management of its corresponding component. In this auto-management process, each element compares its current QoS with the QoS range of its corresponding community. If the current QoS is into the community's QoS range, then the QoS-Agent sends a notification called "IN Contract" in order to inform the other community members that the community contract is still respected. If not, the QoS-Agent sends a notification called "OUT Contact". In this case, the element exits from the community and tries to find another VXCU with a new Community ID.

It should be mentioned that all these communities (VXCUs) and real-time profiles are stored in a novel knowledge base called Infoware [5]. It is the subject of another working group in our UBIS ("User-centric": uBiquity and Integration of Services) project. This Infoware manages efficiently and dynamically the decisional and reactive information. It is well structured and acts as a real-time informational inference. In this sub-section, we explained how the E2E end-user's session uses ubiquitous VXCU members in order to be easily adapted to the changes caused by mobility. In the next sub-section, we apply the CoI concept to overcome the spatial mobility challenge by using the end-user's location as an "Interest".

3.2 Location-Based Virtual Community: VXCL

Another main aspect of our user-centric approach is the end-user's context awareness [6]. The context is any relevant information that can be provided to characterize the end-user's ambient environment. End-users are context aware when they are capable to adapt their ambient sessions according to the received ambient information. As opposed to fixed distributed systems, mobile systems applications are executed in an extremely dynamic context. Specifically, "Location" is one of the main aspects that cause great modifications in end-users context. For this purpose, context aware end-users should receive information according to their locations. Many concepts are treating location based information. Ambient Networks (ANs) [7] is one of these concepts. It tries to incorporate location based information into service provisioning, network composition, and service adaptation. Exploiting this information provides end-users with added-value services and an enhanced communication experience. Actually, ANs intervene by adapting network interconnections in order to provide a wide range of services. Consequently, we notice that ANs main contribution is on the network and service layers. In this paper, we enhance the ANs concept, enrich end-users experiences, and anticipate end-users demands by providing information about all end-users E2E location based ambient resources (services, networks, equipment). To reach this goal, we apply the CoI concept while using the end-user's "Location" as "Interest" for creating and managing Location-based Virtual Communities (VXCLs, L: Location). The latter are location aware communities that gather E2E resources that have the same location as the end-user. For each end-user's location, three VXCLs are created (VSCL, VCCL and VECL).

In the first step, we introduce the four basic services that are used in the creation and management of these VXCLs:

– **Location Basic Service (LBS):** This component determines an element position (Mapping Address: e.g. 11/300 George St, Sydney NSW 2000, Australia) while having its Element ID as an input.
– **Discovery Basic Service (DBS):** This component launches a search in order to discover new elements that meet some demanded criteria (indicated at the input).
– **Presence Basic Service (PBS):** This component filters an obtained list according to the resources states. It selects among a list of Element IDs the ones corresponding to "Availabe" (Accessible by the end-user), "Activable" (activated by the provider but not yet used) and "Activated" (Activable and used) elements.
– **Sorting By Type Basic Service:** This component sorts by type (service, network, equipment) a list of Element IDs received at the input.

In our context, "Location" is represented by a Mapping Address that appears in the end-user's real-time profile. Therefore, Mapping Address might change when end-users

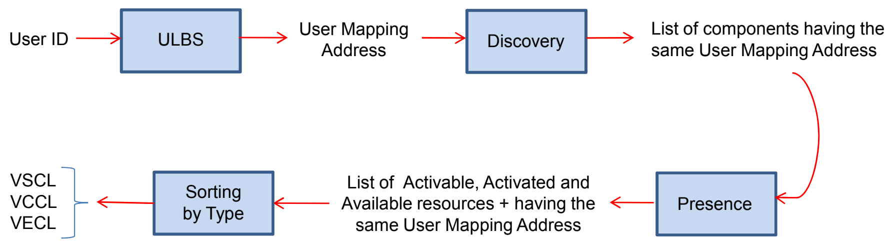

Fig. 1. VXCLs creation workflow

change their terminals (user mobility) or change their location (terminal mobility). In order to easily adapt the E2E session to this spatial mobility, VXCLs are dynamically and seamlessly created and managed according to the hereafter workflow (Fig. 1) that is composed out of six steps:

- **Step 1:** When an end-user changes his/her Mapping Address during an activated session, he/she directly notifies the Infoware that contains the real-time profile in order to update its Mapping Address field.
- **Step 2:** The Location Basic Service localizes the end-user by using the end-user's ID as input. ULBS gives the new Mapping Address at its output.
- **Step 3:** The Discovery Basic Service finds all ambient resources (services, networks, equipment) that have the same Mapping Address as the end-user. For this purpose, DBS uses the end-user's Mapping Address given at its input.
- **Step 4:** The Presence Basic Service is applied on the discovered resources list. It chooses among these resources the "Available", "Activable", and "Activated" ones.
- **Step 5:** The Sorting By Type (Service, Network, Equipment) basic service classifies the obtained result into three lists (VSCL, VCCL and VECL).
- **Step 6:** If the end-user prefers to receive the VXCL lists that correspond to his/her location, he/she should activate a service called the "VXCL Service". The latter permits to launch the workflow and then to send the obtained results to the end-user.

In this sub-section, we introduced the location based virtual community approach that permits to support the ANs concept and to enhance the context awareness for end-users. It should be mentioned that these VXCLs are not stored for a long time because they expire once the end-user changes his/her location. After explaining the VXCU and VXCL approaches, we study in the next sub-section the possibility of their combination.

3.3 CoIs Combination: VXCL&U

In order to manage mobility challenges, we introduce the CoI concept with two main "Interests". Each "Interest" permits the creation and management of a type of VXCs. However, the combination of these interests is considered as an obvious inference that enhances the E2E mobility management process and the end-user's context awareness. In fact, it is possible to have several elements that belong to the same VXCU and VXCL. Consequently, they form what we call a Location and Ubiquity-based Virtual Community (VXCL&U).

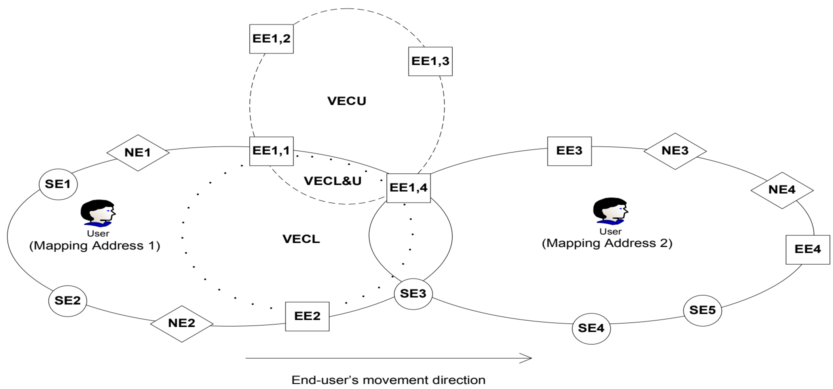

Fig. 2. VXCL&U: VXCU and VXCL combination

An example of VECL&U creation is given in figure 2. In fact, for each Mapping Address, the end-user gets a VECL. Among the VECL members, *EE1,1* and *EE1,4* are QoS and functionally equivalent (belong to the same VECU). Thus, they form a new VECL&U. If the end-user faces a QoS degradation in one of these equipment, he/she can switch to the second equipment in order to maintain his/her continuous session.

4 Feasibility

To validate the feasibility of our CoI proposition, we implement the VXCU approach by using the P2P network proposed by JXTA (Section 4.1), and we implement the VXCL approach by using EJBs (Section 4.2).

```
<?xml version="1.0" ?>
<!DOCTYPE jxta:PD (View Source for full doctype...)>
<jxta:PD type="jxta:PD" xmlns:jxta="http://jxta.org">
  <PID>urn:jxta:uuid-59616261646162614A78746150325033962AABD49FFD4CDC9F54E187EABE4CE303</PID>
  <Name>PeerES</Name>
  <ElementID>ElementID1</ElementID>
  <Type>Service</Type>
  <Funct>Funct1</Funct>
  <Desc>A specification</Desc>
  <SURI />
  <Vers>1.0</Vers>
  <QOS>QOS1</QOS>
  <SipAddr>127.0.0.1:5061</SipAddr>
</jxta:PD>
```

(a)

```
<?xml version="1.0" ?>
<!DOCTYPE jxta:PGD (View Source for full doctype...)>
<jxta:PGD type="jxta:PGD" xmlns:jxta="http://jxta.org">
  <GID>urn:jxta:uuid-8AAB12F4997C46EBBB6D9F36FAE17AE202</GID>
  <CommunityID>CommunityID2</CommunityID>
  <Funct>Funct2</Funct>
  <type>VSCU</type>
  <SURI />
  <Vers>1.0</Vers>
  <QOS>QOS2</QOS>
</jxta:PGD>
```

(b)

Fig. 3. Advertisements: (a) PeerAdvertisement, (b) PeerGroupAdvertisement

```
myDiscoveryService.addDiscoveryListener((DiscoveryListener) this);
myDiscoveryService.getRemoteAdvertisements(null,DiscoveryService.GROUP,"Funct",Funct,10);
```

Fig. 4. JXTA remote discovery methods

4.1 JXTA for Feasible VXCU

The JXTA platform [8] is based on a P2P network and a set of open-source protocols developed by SUN. JXTA abstracts the complexity of the underlying layers (network, transport) and permits the conception of auto-managed and stateless components.

In a P2P manner, these components (service, network, equipment) called Edge Peers, communicate, collaborate and share their resources. For this purpose, they publish advertisements (XML models) that contain their own descriptions. In this paper, we describe two types of advertisement: the first one is the PeerAdvertisement (Fig. 3(a)) which describes each peer. It contains the peer's name, ID, type, functionality, QoS, etc. The second one is the PeerGroupAdvertisement (Fig. 3(b)) which describes a peer group. The latter represents our VXCU. It contains the VXCU's communityID, type, functionality, QoS, etc. In the following, we limit the feasibility part to the service layer. Thus, each peer refers to a Service Element (SE).

In the deployment phase, each peer discovers its ubiquitous peers (with equivalent QoS and functionality) and creates or subscribes to a community called VSCU. For this purpose, we conceive and integrate a JXTA-Agent into each peer. Its role is to instantiate an object (peer) in the JXTA platform by using the *newNetPeerGroup()* method. The latter permits the peer creation and its access to all basic services that are proposed by the JXTA default group (NetPeerGroup). When we are deploying a peer, the latter publishes a PeerAdvertisement. Our JXTA-Agent launches the *searchCommunity()* method in order to discover the VSCU that contains its ubiquitous peers. First, the JXTA-Agent starts the search in its NetPeerGroup. If the SE does not locally find its desired VSCU, it spreads the search into the whole JXTA network by using a Rendez-vous Peer. The SE uses a DiscoveryListener to keep listening to the replies during a fixed timeout. According to figure 4, we use the *getRemoteAdvertisements()* method while having the "SE functionality" as the discovery criteria. Thus, a list of PeerGroupAdvertisements is obtained. On this list, we apply a newly developed method in order to filter the results according to the QoS criteria. Therefore, we can consider that the discovery is launched according to the QoS and functionality criteria. Hence, two cases appear:

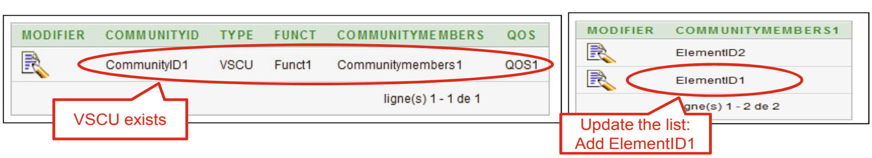

Fig. 5. First case: VSCU update

Fig. 6. Second case: VSCU creation

In the first case, the SE receives a local or a remote PeerGroupAdvertisement describing the VSCU that suits its QoS and functionality. Consequently, the SE joins the VSCU by using the *joinToGroup()* method. In order to conserve all the communities and elements information, we create three tables (CommunityProfile, CommunityMembers and RealtimeProfile) in the database. They are managed by the JXTA-Agent by using a newly developed class called *CommunityManagement*. In this first case (Fig. 5), the JXTA-Agent of the SE launches an *update()* method to add the SE's Element ID (*ElementID1*) to the found VSCU CommunityMembers table (*CommunityMembers1*). It also launches another *update()* to add the found VSCU Community ID (*CommunityID1*) to the SE RealtimeProfile table.

In the second case, the SE did not receive a PeerGroupAdvertisement before the timeout, thus it did not find any VSCU that suits its QoS and functionality. Consequently, the SE creates a new VSCU with a new Community ID. The JXTA-Agent publishes the PeerGroupAdvertisement that describes this new VSCU. Then, it uses the *CommunityManagement* methods to update the database tables (Fig. 6). First, it adds the new community into the CommunityProfile. Second, it creates a new table (*CommunityMembers1*) that only contains one SE (*ElementID1*). Third, it updates the SE RealtimeProfile table by adding the new VSCU Community ID (*CommunityID1*).

After explaining the deployment's feasibility part, we pass to the exploitation phase. In this part, the QoS-Agent launches "IN/OUT Contract" messages. In the case of the "OUT Contract", the JXTA-Agent uses the *leaveFromGroup()* method of the *CommunityManagement* class in order to delete the SE's Element ID from the CommunityMembers table, and to delete the SE's Community ID from the RealtimeProfile table.

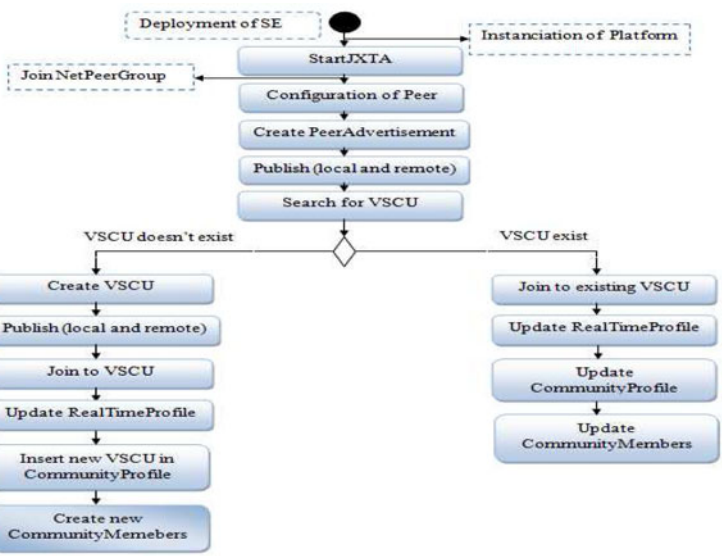

Fig. 7. Logic process of JXTA-Agents

The figure 7 represents the aforementioned logic process of our JXTA-Agent. The same process can be done on the network and equipment layers.

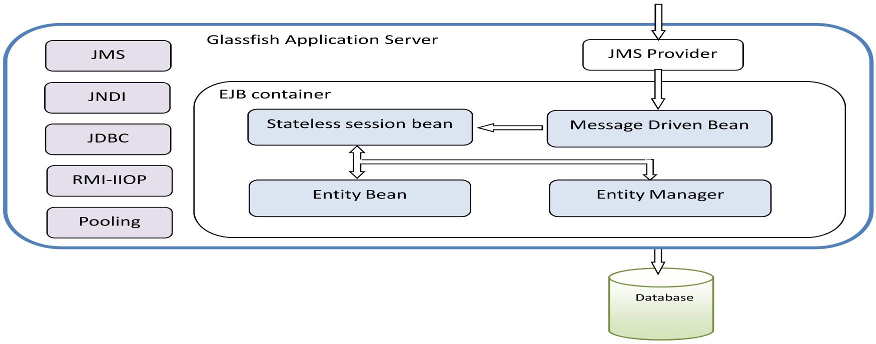

Fig. 8. EJB architecture in the Glassfish application server

4.2 EJBs for Feasible VXCL

The VXCLs creation process is based on a workflow that relates four basic services (Location, Discovery, Presence and Sorting By Type). In order to validate our proposition, we have developed the aforementioned basic services as independent EJBs [9]. The latter permit the creation of different autonomous and loosely coupled components. Besides, EJB provides transparency for distributed transactions and helps the creation of portable and scalable solutions. EJBs are deployed on Glassfish v3 application server [10]. Glassfish v3 is JEE6 certified and consequently supports different APIs such as JMS, JNDI, JDBC and RMI-IIOP. Actually, each EJB service component has the following structure (Fig. 8):

MODIFIER	ELEMENT_ID	TYPE	STATUS	ADDRESS	COMMUNITYID
	ElementID1	service	activated	address1	CommunityID1
	ElementID4	user	accessible	address1	-
	ElementID6	netw	available	address1	CommunityID4
	ElementID7	equip	available	address4	CommunityID3
	ElementID3	service	activatable	address2	CommunityID2
	ElementID5	equip	activated	address1	CommunityID5
	elementID9	service	non activated	address1	-
	ElementID2	service	activated	address1	CommunityID1
	ElementID8	user	accessible	address3	-
	ElementID10	service	activatable	address1	communityID6
				ligne(s) 1 - 10 de 10	

Fig. 9. The “Database” table

– **Entity Bean:** it is a lightweight persistence domain object that represents one or more tables in a relational database. In our project, the information is stocked in an Infoware. The latter is not used in our feasibility study; instead we assembled all the needed information in one Oracle table that we called “Database”. In figure 9, we present a database sample containing different elements with different Types, Statuses, Mapping Addresses and Community IDs.

– **Entity Manager:** it is used to create and remove persistent entity instances, to find entities by their primary key, and to query over entities.
– **Stateless Session Bean:** All beans hold *Conversations* with clients at some level. A *Conversation* is a number of method calls between the client and the bean. A *Stateless Session Bean* is a bean that holds *Conversations* that span a single method call. It is stateless because it does not require state to be maintained across multiple clients requests. Hence, each *Stateless Session Bean* serves many clients and permits to easily pool and reuse components. We must note that these *Stateless Session Beans* are used to implement our stateless components but they interfere in statefull sessions.
– **Message Driven Bean:** it allows the asynchronous messages processing.

```
INFO: MESSAGE BEAN: Message received:
INFO: ElementID1, ElementID6, ElementID5, ElementID2, ElementID10
INFO:
 The VSCL list is :[ElementID1, ElementID2, ElementID10]
 The VECL list is[ElementID5]
 The VCCL list is :[ElementID6]
```

Fig. 10. The VXCL created lists

In this feasibility part, we consider a moving user. His/her address changes from "address 0" to "address 1". Thus, DBS finds all the resources that have an identical Mapping Address. In consequence, PBS filters the DBS result and provides a list that only contains "Available", "Activable" and "Activated" components. Finally, the "Sorting By Type" component sorts the obtained list according to elements types (service, network, equipment). Three lists that represent the different VXCLs are obtained at the output (Fig. 10).

5 Conclusion and Perspectives

Mobility is seen as a main challenge for NGN. We should meet all mobility needs in order to maintain a continuous end-user's session. For this purpose, an E2E mobility management solution must be performed. In this paper, we have introduced the Community of Interest (CoI) concept, also called the Virtual Community (VXC) concept, and we apply it on all NGN layers (service, network, equipment, user). This novel concept consists of decreasing the NGN complexity by gathering elements according to specific interests. The latter have been particularly chosen to suit E2E mobility management needs. Therefore, we applied "Ubiquity" as the first interest. Henceforth, for each element degradation, the latter is dynamically and seamlessly replaced by an ubiquitous element (having the same functionality and an equivalent QoS) that belongs to the same Ubiquity-based Virtual Community (VXCU). Furthermore, in order to overcome the spatial mobility challenge, we have increased end-users context-awareness, by applying "Location" as the second interest. In fact, all elements that have the same end-user's Mapping Address are gathered into Location-based Virtual Communities

(VXCL). Consequently, these created VXCLs guarantee a continuous end-user session by anticipating the end-user's demands. Four basic services (Location, Discovery, Presence, Sorting By Type) are conceived and interconnected in order to create and manage these VXCLs. By combining the VXCU and VXCL communities, the end-user could easily choose an ubiquitous element among the VXCL members in order to replace any damaged element in its session. In the feasibility part, we guaranteed the VXCU logic process by using the P2P JXTA platform, and the VXCL logic process by implementing the basic services into EJBs.

In our future work, we will treat the temporal mobility that is based on end-user's activities. For this purpose, an activity-based virtual community approach seems to be a good proposition. Furthermore, other user interests might be used to create and manage communities and to decrease the complexity of mobile contexts.

Acknowledgement. The authors would like to thank Amen LABIADH and Salah BEN SALAH for their precious contribution and valuable work that benefited greatly this paper. The authors would also like to thank all the participants in the UBIS project, financed by the French ANR VERSO 2008 in which is situated our work.

References

1. 3GPP: Manifestations of Handover and SRNS Relocation. 3GPP TS 25.832, V4.0.0 (2001)
2. Xie, J., Howitt, I., Shibeika, I.: IEEE 802.11-based mobile ip fast handoff latency analysis. In: The IEEE International Conference on Communications, pp. 6055–6060. IEEE Press, Glasgow (2007)
3. Schulzrinne, H., Widlund, E.: Application-layer mobility using sip. ACM SIGMOBILE Mobile Computing and Communications Review 4(3), 47–57 (2000)
4. Al-Shraideh, F.: Host identity protocol. In: The Fifth International Conference on Networking and the International Conference on Systems, pp. 203–203. IEEE Press, Port Louis (2006)
5. Simoni, N., Yin, C., Du Chene, G.: An intelligent user centric middleware for ngn: Infosphere and ambientgrid. In: The Third International Conference on Communication System Software and Middleware, Bangalore, India, pp. 599–606 (2008)
6. Dey, A.K.: Understanding and Using Context. Personal and Ubiquitous Computing 5(1), 4–7 (2001)
7. Giaffreda, R., et al.: Context-aware communication in ambient networks. In: The 11th Wireless World Research Forum, Oslo, Norway (2004)
8. SUN Microsystems, `https://jxta.dev.java.net/`
9. Sringanesh, R.P., Brose, G., Silverman, M.: Mastering Entreprise JavaBeans 3.0. Wiley Publishing Inc., Chichester (2006)
10. SUN Microsystems, `https://glassfish.dev.java.net/`