An End-to-End Service Provisioning Scenario for the Residential Environment

Juan C. Dueñas, José L. Ruiz, and Manuel Santillán, Universidad Politécnica de Madrid

ABSTRACT

The home of the future is here, or at least the hardware to support it. Home environments are currently characterized by the existence of a wide variety of connected devices. This infrastructure is useful in itself. However, we could get much more from it. We could use it to provide advanced services to the end user. This article presents the experience acquired during the design and implementation of an end-to-end service provisioning platform. We have followed a standards-driven approach to leverage existing middleware with open source assets. The selected middleware platforms are the OSGi Service Platform at the residential end, J2EE technologies at the service provider end, and Web services for coarse-grained coordination and interaction between the different actors. In this way we create a suitable environment for all the stakeholders.

INTRODUCTION

Residential services are already commercially available and widely used. Some of them are mature, such as home surveillance services, which have been available for years; others are taking off right now, such as video streaming or voice over IP. However, there are other services that, while technically possible, have not been successfully adopted yet, such as domotics (sometimes known as home automation). The lack of inexpensive broadband access networks and powerful devices can explain this delay.

This situation has changed dramatically in recent years. The large increase in broadband connections, and the performance improvement and reduction of price in the consumer electronics market are at the root of this change. Moreover, the growth in network subscribers is losing pace. In order to keep their market share, network providers need to create added value for their users. A way of doing this is to promote the provisioning of a wide variety of services to the residential market. These new opportunities offer third party providers entry into the market, similar to what is happening in the mobile telephony market.

The classic *one-one-one* approach (one service, one platform, one provider) is inefficient when installing, operating, and maintaining a wide range of services, as the total cost grows linearly with the number of services deployed. A more technically efficient and cost-effective alternative is to create a suitable environment for all services. If only one platform is used, services share infrastructure and maintenance costs: this is the role of the home gateway. Even if some specific services need multiple devices, this approach eases service deployment, operation, and maintenance.

In order to provide a path toward the development and deployment of home services, we have devised and experimented with the end-to-end service provisioning framework we describe in this article. With this framework, we highlight a practical solution to the service provisioning domain, suitable for use by operators in realistic environments, meeting most of the field standards, supporting the evolution of services, their deployment in heterogeneous platforms, and their availability at low cost.

The article is structured as follows. It begins by briefly describing the residential middleware used in the home. Then it presents an overview of the end-to-end scenario. It continues by explaining the main requirements to be fulfilled by the proposed scenario, the details of which are set out in the following section. The article finishes with some overall conclusions and future work.

THE RESIDENTIAL ENVIRONMENT

One of the main challenges in the residential environment is its intrinsically open nature. This means that it is very difficult for any stakeholder to impose his/her own view on the home environment, and — as in many other information technology (IT) areas — standardization is needed. An existing hardware standardization effort is being made by the Home Gateway initiative

0163-6804/05/$20.00 © 2005 IEEE

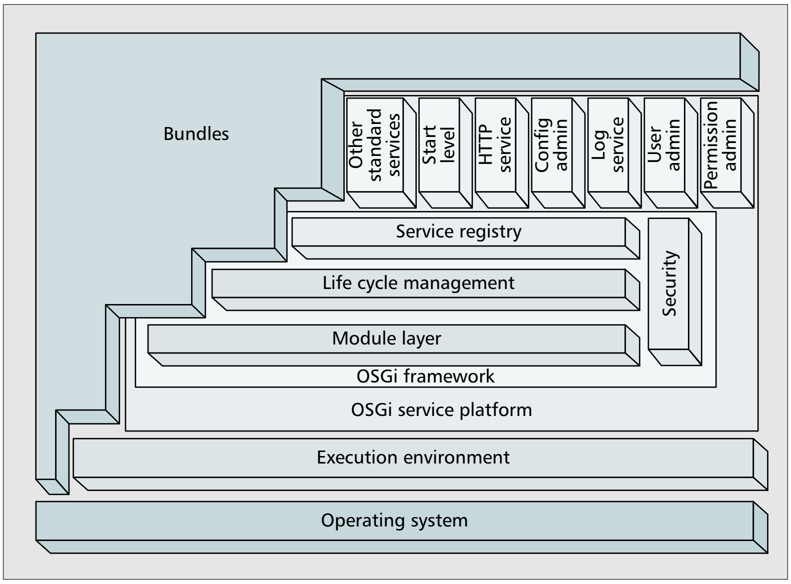
[1]. However, we believe that emphasis should be placed on middleware standardization rather than the underlying hardware platform. Although hardware approaches may offer higher efficiency, middleware ones provide better portability and adaptability to new services and technologies over time.

The OSGi Alliance [2] is an initiative that aims to bring this much-needed standardization to the provisioning of services for the embedded devices market in general and to the residential environment in particular. The OSGi Alliance defines the OSGi Service Platform, a component-based, service-oriented, standard computing environment that provides a suitable framework for the deployment and execution of applications and services. These components and services can easily be deployed from anywhere on the network to the platform. In order to be suitable for a large number of hardware and software combinations, the service platform runs on a Java virtual machine.

There are other technologies targeted at the residential environment. Some of them, such as Home Audio Video Interoperability (HAVi) [3], Universal Plug n' Play (UPnP) [4], and Jini [5], can be seen as complementary [6]. In fact, the OSGi specifications already define interoperability application programming interfaces (APIs) for Jini and UPnP. The Multimedia Home Platform (MHP) [7] and recently launched Windows Media Center (WMC) [8] can be considered alternatives to OSGi. However, both the MHP and WMC are less suitable platforms for the provision of general-purpose services, being tailored to very specific domains: TV broadcasting and multimedia content management and reproduction, respectively.

The OSGi Service Platform is targeted at the embedded device market. Nonetheless, it has also proven its potential in other scenarios, such as the Eclipse Rich Client Platform [9]. However, one of the main scenarios toward which the service platform is oriented is the residential market. This model is called the *service gateway model*, in which a home gateway (possibly a broadband router or set-top box, or a separate hardware device) is the host that contains the service platform. This enables the platform to be located at the edges of both residential networks and wide area networks. In this way, the platform provides an excellent access point for service providers to deliver their services to the end user without dealing with the complexity inherent in the internal composition of residential networks, and the user's privacy is protected. Service providers can then benefit from existing infrastructures to offer high-value services that can be delivered easily to unskilled users.

As shown in Fig. 1, OSGi provides the necessary elements for building deployment and execution runtime on top of an embedded device. Its lightweight nature enables its usage in resource-constrained environments. It leverages the Java sandbox security model by adding authentication and authorization mechanisms that can be used for fine-tuned access control. Additionally, it provides an easy means of controlling application deployment and execution through its application life cycle management infrastructure, as well as prioritization of those contents thanks to its initialization level management capabilities. Moreover, the framework also provides an infrastructure for the dynamic coordination and interoperation of components. This infrastructure follows a service oriented architecture (SOA), thus allowing applications to benefit from its dynamic advantages. An interesting approach to improving the SOA support of the OSGi platform by automating runtime dynamic service dependencies resolution can be found in [6].



■ **Figure 1.** *OSGi Service Platform architecture.*

Scenario Overview

In order to understand the details of service provision, let us start by introducing some use cases. There are multiple services that are suitable for being deployed using an OSGi-based model. For instance, we can think of advanced security and surveillance services, delivery of interactive multimedia content, and remote control of domestic appliances.

Suppose you want to subscribe to a surveillance service to protect your home from intruders. Existing solutions provide a high level of reliability. However, they could easily be improved with new features such as automatic notification of intrusion detection via multimedia messaging service (MMS) and remote video streaming. First, you need to install the relevant hardware. Basic elements include surveillance devices in the home, the service provider's infrastructure, and a "hub" or "gateway," which is connected and routes the alarms to the surveillance service provider. Classic approaches use dedicated lines or plain old telephone service (POTS). A more efficient approach would be to reuse existing IP network infrastructures (cable, digital subscriber line [xDSL], etc.). Moreover, using IP connectivity would ease the process of sending multimedia alarms directly to the end user. The central point of this model is the hub or gateway that communicates with the surveillance service provider.

Another possible service is the provisioning

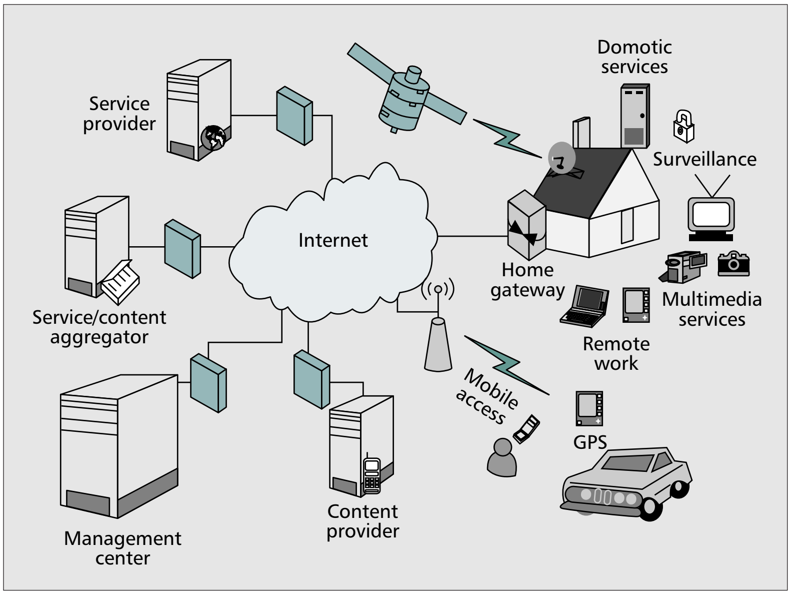


Figure 2. *Stakeholders involved in the scenario.*

of multimedia personalized interactive content to residential environments. Multimedia content can easily be delivered using different technologies and approaches. However, in order to include interactive applications with the contents, you need to provide a framework suitable for their execution. A smart set-top box would be able to handle the multimedia content and its related applications. This box would be connected to both the internal audio-visual (A/V) network and the content provider's network (satellite, cable, xDSL, etc.). The personalization server can match the multimedia contents to user preferences, using XML metadata that describes contents and user preferences, such as TV Anytime and MPEG-7 [10, 11].

Now that you have subscribed to both the surveillance and multimedia content provisioning services, and installed and configured both the security hub and the set-top box, you may be tempted to subscribe to a new service that enables you to monitor and control your home appliances remotely. You will then be able to turn your microwave oven on and off or control your garden irrigation system from any part of the planet. However, you need to install and configure yet another device that acts as an intermediary or gateway between your existing home networks and devices and the external network and service providers. Moreover, further extensions to your smart home's infrastructure — or even upgrades — will require time and effort.

The need for a more cost-effective approach is quite evident. Furthermore, the fragmented solutions that exist today require the user to worry about obscure configuration details and tricky installation processes that discourage the majority of potential clients. We propose to leverage the OSGi Service Platform by creating an end-to-end service delivery scenario that enables all involved stakeholders to interact transparently to provide valuable services to the end user.

The ultimate goal of this scenario is to create an environment suitable for all actors. End users should be able to browse all available services and subscribe to those in which they are interested in, without having to worry about any configuration issues. Service providers should be able to add, remove, and update their services without the end user noticing. Accounting and billing activities should be carried out efficiently. All operation and management activities should be transparent to the end user.

Figure 2 shows the conceptual roles involved in the scenario, although a stakeholder could eventually play more than one role simultaneously. Service aggregators, service providers, content providers, and the operator — represented in Fig. 2 as the control center — make up a business ecosystem. The operator is in charge of all the administrative tasks surrounding the scenario. It must include suitable elements for the other entities to add their contents and services, and present them to the user in a proper manner. It must also provide mechanisms to detect and correct problems on the user side.

At the other end, we have the residential environment, which is the target for the delivery of services. There are a number of networks, classified [12] into domotic networks (e.g., X10), data networks (Ethernet, IEEE 802.11x, HomePNA), and multimedia networks (IEEE 1394, etc.), although there is a convergence tendency among them. The home is connected to the Internet through a broadband connection managed by a home gateway. The home gateway is the natural access point for the rest of the stakeholders.

Now that we have seen the main entities involved in the scenario, let us look at the aforementioned surveillance service use case.

Let us suppose that the user has all the necessary hardware installed at home, including a correctly configured home gateway he/she has bought, for instance, to remotely control his/her appliances remotely as well as cameras and motion detection devices. Also suppose that a security company has developed a surveillance service. After negotiating, the company reaches an agreement with the operator through which the latter adds the service to its repository. This negotiation and addition process is possibly automated and carried out through Web services. Eventually, the user browses the operator's service catalog, decides that the surveillance service looks interesting, and buys it. At that moment, the operator deploys to the home gateway — which includes an OSGi Service Platform — the surveillance application and all its related components. The operator also informs the service provider that a subscription has been taken out. At this point, the service is activated. The user can configure his/her preferences, including the format of notifications (MMS, email, etc.), whether the police should be immediately informed of any break-in, or whether user confirmation is needed.

Eventually, the surveillance system (Barking Intruder Detection System, BarkIDS, in Fig. 3) detects an intrusion. It then informs the service provider of the event. The service provider, in turn, makes a decision based on the user preferences. This decision might involve some video streaming to the user's cellular phone. Another option would be simply to send an MMS with the intruder's photograph.

Let us look now at the advanced multimedia service example detailed in Fig. 4. The initial

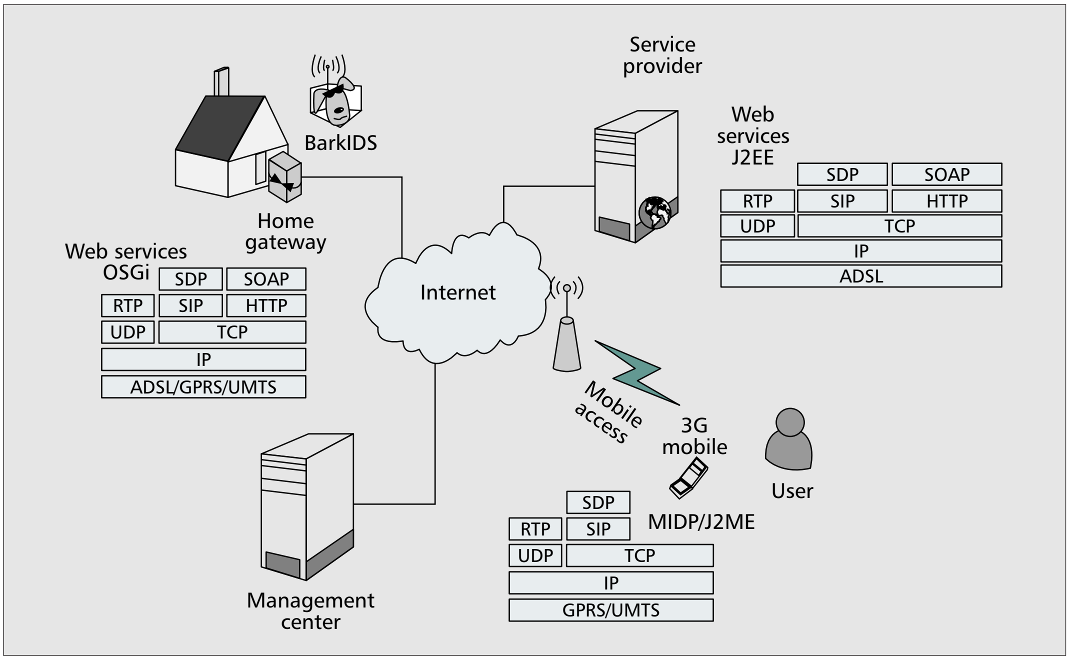


Figure 3. *Example surveillance service.*

process is similar. The content provider contacts the operator and adds the service to the repository. The user subscribes to the service. At this moment, the deployment process starts, and all the necessary elements — including codecs and a state-of-the-art player — are deployed to the home gateway. Eventually, the user searches for available content. The personalization server deals with content organization and filtering to provide a view adapted to the user's profile. When the user selects specific content, its information (i.e., content URL and other related metadata) and associated applications are automatically deployed, installed, and configured. Then the interactive content is played.

In both the surveillance service and the interactive content example, we can see that the central point of the scenario is the home gateway. It constitutes a single point of access and execution of services. It integrates the networks and smart devices at home, and thus promotes the development and provision of highly attractive services to the user.

Stakeholders' needs are satisfied in this scenario. Network providers benefit from new users that subscribe to high-bandwidth connections to enjoy services at home. Service providers have new business opportunities in the massive residential market. The end user, in turn, benefits from a digitally powered home, relieved from PC-related inconveniences, with improved usability, reliability, and stability, only concerned about browsing, selecting, and enjoying services at home.

Main Needs of the Scenario

The presented scenario is attractive. However, it is also very challenging. To make it possible, many technical issues need to be addressed. The main requirements of the scenario can easily be elicited from the presented use cases. These include administration, deployment, security, operations support, and integration of heterogeneous systems.

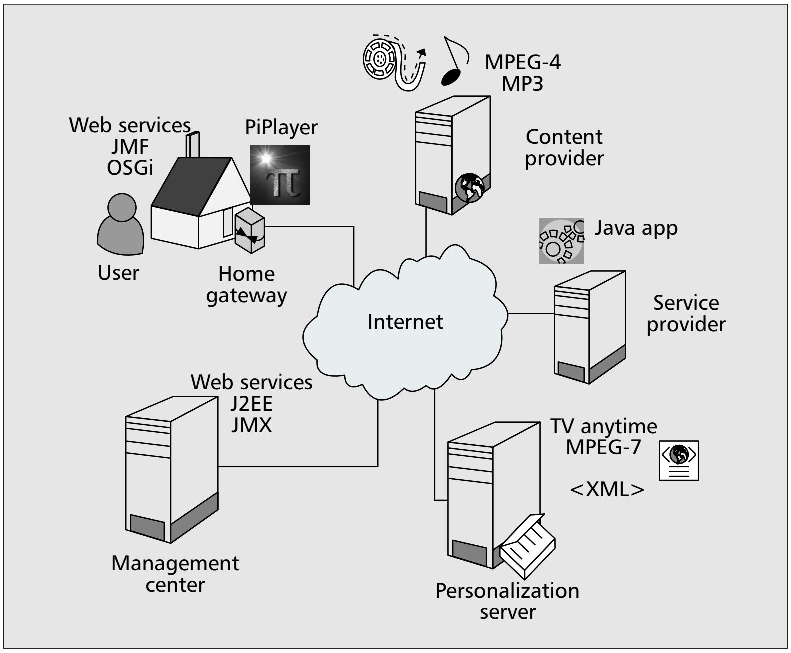


Figure 4. *Personalized interactive multimedia service.*

The resulting solution must provide:

- Automated deployment: Once the user is interested in subscribing to a service, the deployment of software assets must be carried out transparently. In brief, the process should seem to the user to be a click-to-deploy task.
- Zero administration: The user should not have to bother with the configuration and management of the complex environment involved in the scenario; many services are seamlessly carried out through heterogeneous networks and smart devices. This task will have to be done automatically and

> *We envision a tendency from centralized to distributed situations. This would eventually end up being a kind of P2P model where anyone, including end users, would act as a service/content provider.*

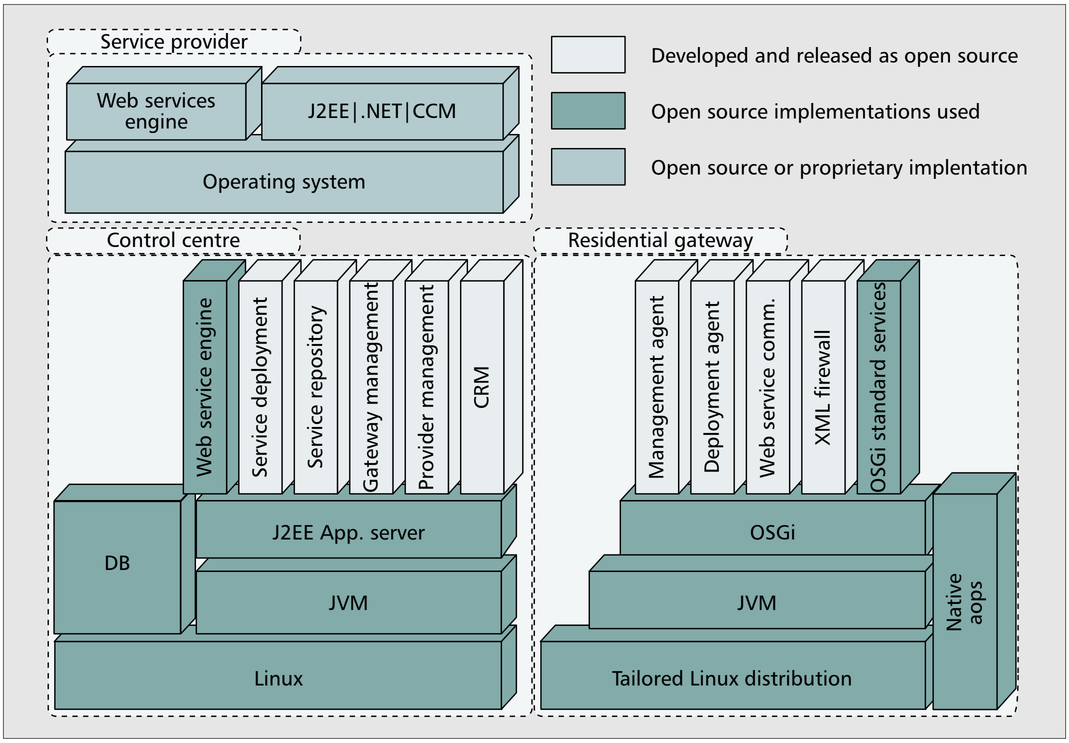


Figure 5. *Components of the scenario.*

possibly also remotely, with the aid of advanced tools and middleware.

- Security: Understood in a very broad sense, it comprises home surveillance as well as privacy and confidentiality in communications between the outside world and the home — particularly in the presence of wireless networks. This is of particular importance because of the existence of payment transactions.
- Operations support: This includes client relationship management (CRM) functionalities, stocking, and service subscription mechanisms and automatic service deployment and management support.
- Integration of heterogeneous systems: For example the control center will have to interoperate with the rest of the stakeholders in a standard way, being the hub of the scenario. Service providers should be able to publish their assets easily, while accessing accounting and billing information or updating their services. Several approaches are possible in this respect, from a centralized catalog placed at the control center to a highly distributed but federated network of repositories. We envision a tendency from centralized to distributed situations. This would eventually end up being a kind of P2P model where anyone, including end users, could act as a service/content provider.

Implementation of the End-to-End Scenario

In such a complex environment, the use of open standards promotes an easy interaction between the involved stakeholders. This is particularly true in those interactions concerning the control center and service/content providers/aggregators. Residential environment standardization can easily be achieved by adopting the OSGi Alliance's Service Platform. The most effective way to enable communication between companies is Web services technology. SOA is the best option for promoting interoperability with independence of the specific technologies and platforms available at different sites.

To satisfy the needs defined in the previous section, we have defined a suitable architecture, designed the required components, and implemented them. Our solution follows an open standards-driven approach, leveraged by the usage of open source components.

The implementation of the scenario has been divided into three phases: selection of the underlying middleware platforms, Jonas (an open source J2EE certified application server) for the control center and Oscar (an open source OSGi R3 implementation) for the home gateway; implementation of the required components at each end of the scenario; and validation of the scenario with the development and deployment of proof-of-concept services [13]. Web services enable service providers to choose the platform that best meets their particular business needs and technical requirements. As a result, there is no need to impose a model for the service provider architecture. Figure 5 shows the system design. In the following subsections we describe how these components fulfill the requirements.

Automated Deployment

From the users' point of view, service deployment must be a one-click task. However, services can be made up of several components and may also bring about dependencies on other related services and resources. Traditional deployment approaches focus on packaging together all the necessary elements, and are therefore fairly limited. In addition, the duplication of components may appear in the medium term for each platform (so far, the sharing of components or dynamic link libraries in evolving platforms has proved ineffective).

However, these approaches make service updating too laborious and error-prone. To overcome this problem, we have used automated dependency resolution techniques. Dependencies cut across different layers. System level dependencies coexist with middleware and application levels, so they must be handled consistently. The aforementioned personalized interactive multimedia service is a good example of this situation.

The deployment infrastructure has been designed based on the Object Management Group (OMG) distributed component deployment and configuration standard [14]. The service deployment component in the control center and the deployment agent at the home gateway provide this functionality. The deployment agent implements context-aware algorithms to automate the resolution of dependencies at both the system and service platform levels. The deployment agent has been released as the open source project Jbones [13]. The service deployment component at the server side deals with service subscriptions by sending Web service messages to the deployment agent to launch the deployment process.

Administration

Administration tasks such as failure detection, configuration management, and performance analysis are crucial for the correct functioning of the system. Operator-based administration requires a remote management infrastructure to be deployed.

We have used the classic management architecture based on the remote manager-management agent pairing, following the OSGi reference section on remote management. The management agent, JMood [13], is also available as an open source project.

In this management model the agent gathers information and carries out operations on behalf of the remote manager, and asynchronously sends events. These administration facilities enable the operator to control aspects such as application and service performance monitoring, configuration management, logs, permissions, and lower-level monitoring (e.g., monitoring of the underlying Java virtual machine performance). Its autonomous management capabilities improve scalability by reducing the burden of the remote manager and the network load. The management agent provides support for the definition of operational policies based on different parameters, such as memory overload or network congestion.

From the operator's point of view, Simple Network Management Protocol (SNMP) as the management technology has advantages related to widespread support from network equipment vendors. However, SNMP is not well suited for dynamic service instrumentation. For this reason, we have used Java Management Extensions (JMX) [15] technology for gateway management. JMX is a Java Community Process standard broadly used in J2EE application servers' management. JMX makes application and service instrumentation painless, while at the same time decoupling the control plane from the services plane.

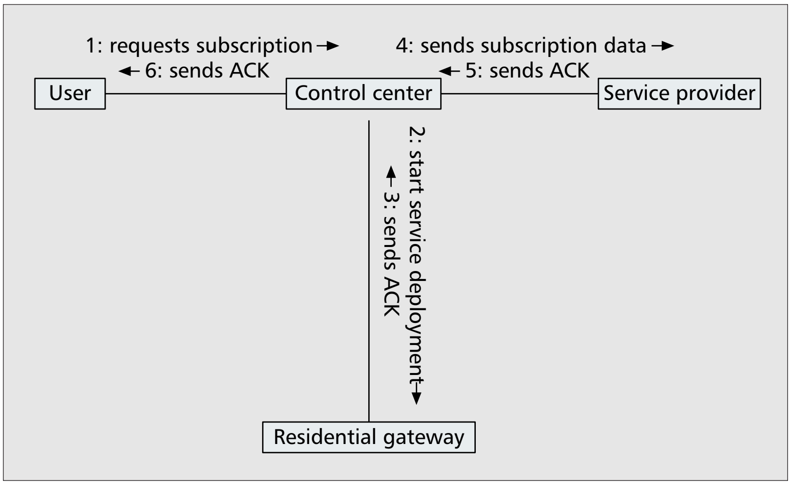


Figure 6. *Service subscription process.*

Operations Support

The Operations Support Systems through Java (OSS/J) initiative [16] aims to improve interoperability between different OSS products. OSS/J builds on current J2EE standards to provide API specifications to ensure compatibility between different solutions. A reference implementation for each API is also made available, which enables rapid development of OSS-based services. OSS/J has provided us with the conceptual framework to build the control center architecture. We have implemented the OSS/J service activation API in the service deployment component.

A key element in the proposed scenario that represents a step forward in services management is integration of the remote manager with the enterprise operation support environment defined by emerging industrial management standards [17], which has chosen OSS-J as the technical infrastructure for management business processes.

The service repository component is based on some of the ideas included in the OSS service inventory API. The stocking of the repository is carried out by service providers, who are registered and managed by the provider management component as detailed in Fig. 5.

Service providers populate the service repository using the provider management interface. Figure 6 shows the interactions that take place during the subscription process. Users registered in the CRM component can log into the OSS and browse the service catalog for interesting services. Eventually, the user subscribes to a service. This makes the service deployment process start. Once the deployment process has been successfully carried out, the service deployment component notifies the service provider via Web services. This finishes the deployment process, and the service is ready to be used.

Future work includes the deployment of our scenario in a test bed with a large number of home gateways for feedback on non-functional attributes such as reliability, response to high traffic workloads, scalability and security.

Security Issues

In order to provide a coherent security model for the home gateway, a generic approach is needed. Access control and information protection tasks have to be carried out effectively. OSGi access control mechanisms include a permission model based on Java's sandbox model, which assigns execution permissions to applications depending on their URL. It also provides a simple authentication and authorization model.

Our scenario relies heavily on Web services as the transport mechanisms for sensitive management information. Sealing this channel is very important. Therefore, classic lower-level security systems such as personal and network firewalls are complemented by an application-level firewall. In Fig. 5 the XML firewall provides these security functionalities.

Conclusions

We have reported our experience in defining and implementing an end-to-end service provisioning scenario capable of handling numerous types of end-user services and providing an operational solution that integrates open standards in order to fulfill the requirements of the involved stakeholders: end users, service providers, and network operators.

We started the article by presenting two sample advanced services: personalized interactive multimedia service and home surveillance services. We have developed these services to validate the scenario. The resulting implementations have been released as open source projects, and are called Personalized Interactive Player (PiPlayer) and BarkIDS [13].

Selection of the basic platforms has been a key point for the definition of the architecture. We have chosen the OSGi services platform for the residential gateway and J2EE for the operator's side. Once the architecture was defined, we implemented its components. The strategic points of the architecture are the deployment and administration infrastructure. The former automates the context-aware deployment operations, and the latter provides a policy-based service management agent for remote operation and maintenance.

The home gateway is the central element of the scenario. It is a common platform for the deployment and execution of services, thereby enabling a technically efficient and cost-effective solution for their provision to the residential environment. However, it also has drawbacks: the home gateway is a single point of failure for all services in the home, making it unsuitable for safety-critical domains. The existence of fault detection mechanisms reduces the impact of failures. Nevertheless, depending on the kind of services deployed, performance monitoring measures might be necessary.

We have already proven the main functional aspects: automated deployment, administration, and operation support. Future work includes the deployment of our scenario in a testbed with a large number of home gateways for feedback on nonfunctional attributes such as reliability, response to high traffic workloads, scalability, and security. The adaptation of the scenario from the residential environment to mobile network services is also underway.

Acknowledgments

We would like to thank Jesús Bermejo and the Spanish company Telvent for their encouraging support in the development of this work in the project Eureka ITEA-OSMOSE. We would also like to thank the Spanish Ministerio de Educación y Ciencia, Dirección General de Investigación, which partially funded this work under grant TIC2002-12427-E.

References

[1] Home Gateway Initiative, http://www.homegatewayinitiative.com

[2] The OSGi alliance, http://www.osgi.org

[3] Home Audio Video Interoperability, http://www.havi.org

[4] Universal Plug 'n Play, http://www.upnp.org

[5] Jini Specs. and API Archive, http://java.sun.com/products/jini/

[6] R. Hall and H. Cervantes, "Challenges in Building Service-Oriented Applications for OSGi," *IEEE Commun. Mag.*, May 2004, vol. 42, no. 5.

[7] ETSI TS 102 812, "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP)," Spec. 1.1.1, v. 1.2.1, 2003.

[8] Microsoft Windows XP Media Center Edition 2005, http://www.microsoft.com/windowsxp/mediacenter/default.mspx

[9] The Eclipse Rich Client Platform, http://eclipse.org

[10] ETSI TS 102 822-3-1, "Broadcast and Online Services: Search, Select, and Rightful Use of Content on Personal Storage Systems ("TV-Anytime Phase 1"); Part 3: Metadata; Subpart 1: Metadata Schemas."

[11] MPEG-7, pt. 5, ISO 15938-5:2003, "Multimedia Description Schemes."

[12] P. Utton and E. Scharf, "A Fault Diagnosis System for the Connected Home," *IEEE Commun. Mag.*, Nov. 2004, vol. 42, no. 11.

[13] Open Source for Open Systems, http://www.os4os.org/

[14] "Deployment and Configuration of Component-Based Distributed Applications Specification," OMG-PTC 2003-07-08.

[15] Java Management eXtensions, http://java.sun.com/products/JavaManagement/

[16] Operations support systems through Java initiative, http://www.ossj.org/

[17] TeleManagement Forum, http://www.tmforum.org/

Biographies

JUAN C. DUEÑAS (jcduenas@dit.upm.es) is an associate professor at the Department of Telematics Engineering at ETSI Telecomunicación, Universidad Politécnica de Madrid, Spain. He received his Ph.D. degree in telecommunication engineering in 1994. His thesis was awarded with the Spanish Engineers Association prize and UPM doctoral prize. Since then he has worked as a technical researcher and research manager on several Europe-wide projects in the IST and Eureka ITEA programs in the areas of services engineering, the Internet, and software architectures for distributed applications. He is currently deputy director of the Department of Telematics Engineering.

JOSÉ L. RUIZ (jlruiz@dit.upm.es) is a Ph.D. candidate at the Universidad Politécnica de Madrid, and is involved in several European projects as a technical researcher. He received an M.Eng. degree in telecommunication engineering from that university in 2001. His research interests are in services engineering and deployment and open source approaches to these topics.

MANUEL SANTILLAN (santillan@dit.upm.es) received his M.Eng. degree in telecommunication engineering from Universidad Politécnica de Madrid in 2004. He joined this research group in 2003, where he developed his Master's thesis in the area of services management. Since then he has been involved in several European projects as a technical researcher while conducting his Ph.D. work. His research interests include services engineering and management and multimedia services.