

Integrating IMS Presence Information in a Service Oriented Architecture

Martin Strohbach¹, Ernő Kovacs¹, and Laurent-Walter Goix²

¹ NEC Europe Ltd., Network Laboratories, Germany
{martin.strohbach, ernoe.kovacs}@netlab.nec.de

² Telecom Italia, Italy
laurentwalter.goix@telecomitalia.it

Abstract. Successful service delivery platforms will need to build on service oriented architectures and provide interoperability with the IP Multimedia Subsystem (IMS). In this extended abstract we describe how presence information that is maintained in IMS can be integrated into a service oriented architecture.

1 Introduction

The 3GPP standardized IP Multimedia Subsystem (IMS) [1] is widely accepted as the future architecture of converging fixed and mobile networks. IMS defines the functional architecture of managed IP-based networks and has been designed to clearly separate transport and signalling functions. For signalling, IMS uses the SIP protocol providing functionalities such as event subscription and notification, and terminal mobility. IMS systems are currently experiencing widespread adoption by mobile operators.

While IMS works on top of heterogeneous network technologies and supports roaming end-users, it does not fully address the challenges of quickly developing personalized and dynamically composable services that are assumed to generate new sources of revenue. In particular, it is important that services can be developed in a cost-effective way and with a fast time to market.

Within the EU project SPICE (Service Platform for Innovative Communication Environment) [3] we address these challenges by developing a service oriented architecture for the rapid development of mobile services. A central topic within SPICE is the integration and evolution of IMS to a service oriented architecture that addresses the needs of tomorrow's service platforms [4].

In this extended abstract we briefly present our ongoing work of a SPICE service that offers user presence information stored in the IMS system. For a general overview of the IMS role and its functions within the SPICE platform we refer to Tarkoma et al. [4] and to our project website [3].

2 SPICE Architecture

Figure 1 shows a high level overview of the SPICE architecture. It consists of multiple services that are logically divided into *Component Services*, *Knowledge*

Services, and *Value Added Services*. The SPICE basic component layer provides facilities for component-based development and deployment. This layer provides services like message routers, resource adapters, and managers for common services. Spice basic components implement a set of defined interfaces allowing SPICE basic components to (a) be deployed into various runtime environment and (b) to participate in SPICE system services such as service monitoring and logging, service discovery, service roaming and others.

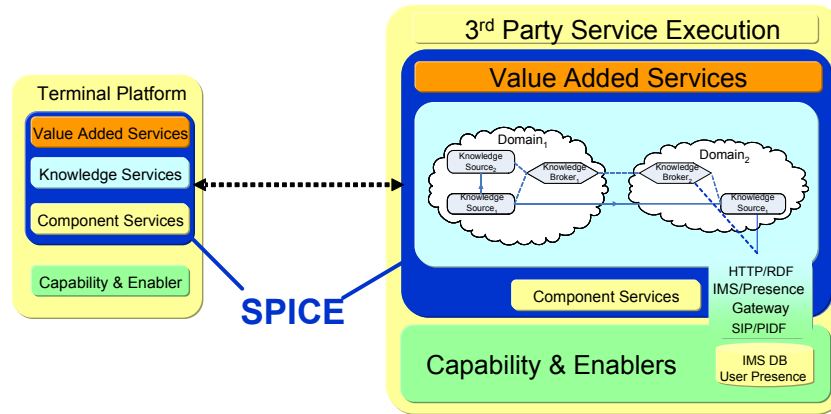


Fig. 1. SPICE Architecture Overview

Knowledge services provide access to various knowledge gathered from the web, network operators, user profiles and information about local resources. Local resources are typically discovered by the user terminal and may include nearby devices, accessible networks, local services and sensor data gathered from wireless sensor networks. The generic form of a knowledge service is called a *knowledge source* and provides an interface for querying and subscribing to knowledge and register with a *knowledge broker*. A knowledge broker is used to find knowledge sources that are able to answer a specific query. Specialized knowledge services include reasoners and recommenders that derive knowledge for personalized end-user services and offer more specific interfaces. The knowledge layer is in itself a service oriented architecture whose components are used by value added services.

The final set of components are the so-called value added services (VAS). These components are the result of a service creation process. They typically encapsulate logic of a new service created for SPICE users. In the VAS layer basic component services and knowledge services are orchestrated to provide the required functionality. VAS components are created using orchestration languages such as Business Process Execution Language (BPEL) in conjunction with advanced techniques like semantic service discovery and matchmaking, automatic service composition, and dynamic control of Web Service requests using

an Enterprise Service Bus, e.g. request enrichment, request redirect, content-based routing, and others.

Components of all layers use OWL-based ontologies to make sure that services interpret the data in the same way and that services of various application and administrative domains can be re-used to compose new services. Various sub-ontologies, e.g. for services, user profiles and presence information are being specified. Knowledge sources therefore provide their data using standardized RDF/OWL (Resource Description Framework/ Web Ontology Language) [2] documents.

3 IMS Presence Integration

An important design goal of the SPICE architecture is to provide highly personalized end-user services by taking the users current situation into account. The current situation of a user can be derived from presence information. As presence data is primarily intended to help users make decisions about how and when a person can be contacted, it will also be necessary to combine user presence with other information. Presence information is therefore integrated into the knowledge layer so that various knowledge sources can be used to compose value added services.

The technical integration is achieved by the IMS presence gateway depicted in Figure 2. The task of the gateway is to provide a bidirectional mapping of data formats and protocols between IMS and SPICE. Presence data originating from IMS is accessed by implementing a presence watcher interface that subscribes to a presence server using the SIP protocol. Change notifications are sent in standardized XML documents as defined in RFC 3863, RFC 4479 and RFC 4480. The mapper within the gateway transforms these presence documents to instances of a presence sub-ontology and offers these RDF/OWL documents to SPICE via the knowledge source web service interface. Vice versa the gateway implements a web service client to access SPICE knowledge sources. Received RDF/OWL instances are transformed to presence documents and pushed to the presence server via the presence source interface of the gateway.

Figure 3 shows an example of how a presence document can be mapped to OWL individuals. The classes of the corresponding sub-ontology is shown on the right. The information shown states that the presentity can be contacted via her assistant using the specified eMail address.

4 Summary

In this extended abstract we have presented how presence information that is maintained in IMS can be integrated into a service oriented architecture such as the SPICE platform. We have argued that future service delivery platforms will need to build on service oriented architectures and provide interoperability with IMS.

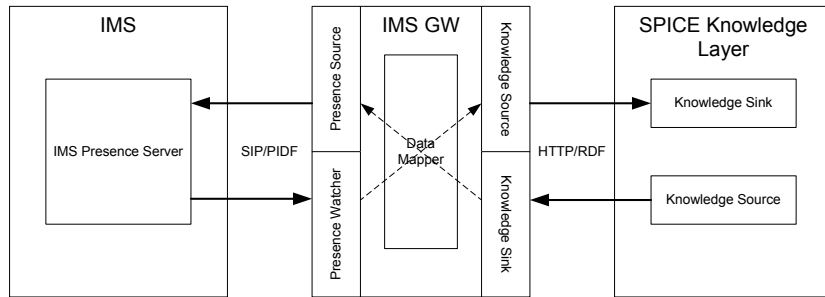


Fig. 2. IMS Gateway



Fig. 3. Mapping example between PIDF and OWL

Acknowledgements

The work presented in this paper is executed as part of the SPICE project and partly funded by the European Union under contract IST-027617. We would like to thank all of our partners for their insightful discussions and contributions.

References

1. IP Multimedia Subsystem Specification (IMS) [online, cited 29/09/2006]. Available from: <http://www.3gpp.org/specs/numbering.htm>.
2. Semantic Web Homepage of the W3C [online, cited 20/12/2006]. Available from: <http://www.w3.org/2001/sw/>.
3. Spice project website [online, cited 13/10/2006]. Available from: <http://www.ist-spice.org/>.
4. S. Tarkoma, C. Prehofer, A. V. Zhdanova, K. Moessner, and E. Kovacs. SPICE: Evolving IMS to Next Generation Service Platforms. In *The 2007 International*

Symposium on Applications and the Internet (SAINT2007), 2007. Accepted for
Publication.