

The MSDA Multi-Protocol Approach to Service Discovery and Access in Pervasive Environments

Pierre-Guillaume Raverdy, Rafik Chibout, Agnès de La Chapelle, Valérie Issarny
INRIA-Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay, France,
firstname.lastname@inria.fr

1. Introduction

Pervasive computing environments bring new challenges for service discovery and service access protocols due to the heterogeneity of both the interconnected networks and the middleware platforms used. In order for clients to truly benefit from all the networked services embedded into the environment, an innovative solution is required that overcomes both the lack of interoperability of the existing discovery and access protocols, and the limited interconnectivity between the different networks available at a location. To address these issues, we propose the MSDA middleware platform (§2) that manages the dynamic composition of the networks in the environment, integrates existing middleware protocols, and provides a generic service to clients for performing service discovery and access in the environment. Our demonstrator (§3) will in particular demonstrate service discovery across multiple discovery domains, the discovery and access to services that are not IP reachable, and the use of context information to filter the available services.

2. The MSDA Middleware Platform

In MSDA, we model the pervasive environment as a dynamic composition of heterogeneous IP networks belonging to different administrative domains, and assume that global IP routing is not supported. Moreover, the Internet and cellular networks are considered as conduits enabling the interconnection of the various networks.

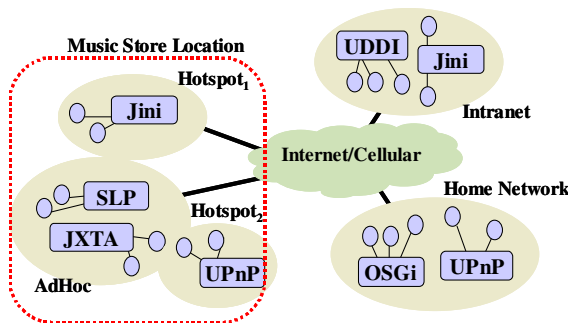


Figure 1 MSDA Pervasive Environment

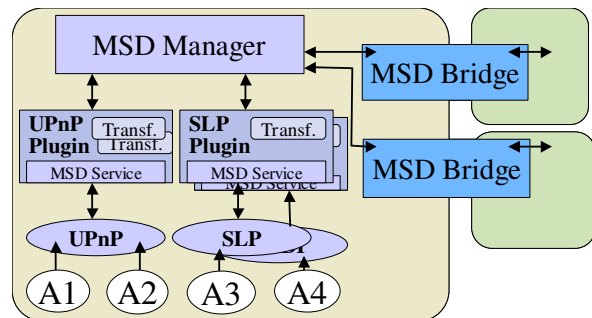


Figure 2: MSDA Platform Overview

Figure 1 illustrates such composition, with a user in a music store able to join either an ad hoc network, or 2 hotspot networks. Some devices (e.g., cellular phones) in the ad hoc and hotspot networks are connected, through cellular networks and the Internet, to remote intranets and home networks.

MSDA is designed around the following architectural principles:

- **Middleware integration:** MSDA is an additional layer on top of existing middleware, and MSDA-aware clients can interact with services in different discovery domains (through MSDA (i.e., using the format and protocols of MSDA)).

- **Dynamic configuration:** MSDA is instantiated independently in each network. MSDAs in nearby networks communicate with each other's to disseminate service information and to provide remote service access. Each network independently selects to which nearby networks to connect to, and filters service information and access requests that it receives/sends from/to these networks.
- **MSDA as a service:** Each instance of MSDA registers as a service (the *MSDA Service*), in each active discovery domain of its network. The MSDA Service provides interfaces for service discovery and service access in the pervasive environment. Therefore applications discover and interact with the MSDA Service using their preferred discovery and access protocol. Providing access to the functionalities offered by MSDA through protocols that the applications already support eases the integration of MSDA into existing applications.

As shown in **Figure 2**, each instance of MSDA in a network is composed of the following components: (i) The *MSDA Manager* that provides service discovery and access to clients within the network; (ii) *Service Discovery and Access (SDA) Plugins* and *Transformers* that interact with specific discovery domains to collect service information and perform service access (e.g., UPnP or SLP Plugins); and (iii) *MSDA Bridges* that assist MSDA Managers in expanding the service discovery and service access to other networks in the whole pervasive environment. In MSDA, services are described using the *MSDA Description* format, which contains service information originating from the initial SD-specific service description, as well as context and propagation information. In each network of the pervasive environment, device administrators configure their devices to participate (or not) in the local MSDA. MSDA-aware devices may only provide profile/status information, or also register to act as MSDA Managers and/or MSDA Bridges.

An MSDA-aware client looking for services in the pervasive environment first discovers the MSDA service that has been registered in the discovery domain (e.g., SLP) by the related SDA Plugin (e.g., SLP Plugin). The MSDA Service implements a pull-based interface for service discovery, in which the client sends a discovery request to the MSDA Service and waits for the results. The MSDA Service forwards the discovery request to the local MSDA Manager that (i) forwards it to the local SDA Plugins for processing, and (ii) forwards it to the active MSDA Bridges that propagate the request to the MSDA Managers in nearby networks. MSDA Managers receiving a remote discovery request process it as a local discovery request (i.e., forward it to local SDA Plugins and to MSDA Bridges), and return the results to the client's MSDA Manager. The client's MSDA Manager collects local and remote results, and returns them to the client through the MSDA Service.

Each SDA Plugin collects service information in a specific discovery domain (e.g., SLP, Jini, UPnP) on behalf of the MSDA Manager. Depending on the SD protocol, the SDA Plugin either registers for service announcements (i.e., push-based SDP), or directly performs service discovery (i.e., pull-based SDP). SDA Plugins generate the initial MSDA Description based on the collected SD-specific service information. This MSDA Description is then forwarded to the Transformers for further processing. Transformers are external components provided by companies, consortiums, or interest groups that extend the initial MSDA Description. For example, a Transformer can dynamically communicate with services supporting a specific interface to collect status information (e.g., service available or overloaded) and include this status information into the MSDA Description.

In the case of push-based protocols, MSDA Description are generated asynchronously by SDA Plugins, and sent to the MSDA Manager that stores them in a local repository. For these SDA Plugins, the MSDA Manager does not forward incoming discovery requests, but directly evaluates the request against the repository, and returns the matching MSDA Descriptions.

3. Demonstration Scenario and Prototype Infrastructure

The demonstrator will illustrate the following scenario:

Ben and his friend stop at Tower Records to buy a new music DVD. The store is full of people using various networked services hosted on the store's intranet, on stand-alone hotspots

installed by music labels, or in ad hoc networks dynamically created by customers. The mobile devices of Ben and his friend autonomously join Tower Records wireless intranet and start to share content with other customers. Ben can also access his home network through the cellular connectivity of his friend's smartphone. As they reach the music section, Ben finds the DVD he was looking for and scans its RFID with his device. He then tries to discover services in the environment that accept a product RFID as input parameter and finds two: a Web service provided by Tower Records and a UPnP service provided by a music label's hotspot. Ben then selects the music label's service and starts to download photos and other promotional content of the artist.

Figure 3 shows the infrastructure of the prototype that implements the above scenario. The figure also shows the interactions among the different networked devices and components, in terms of service discovery and interaction.

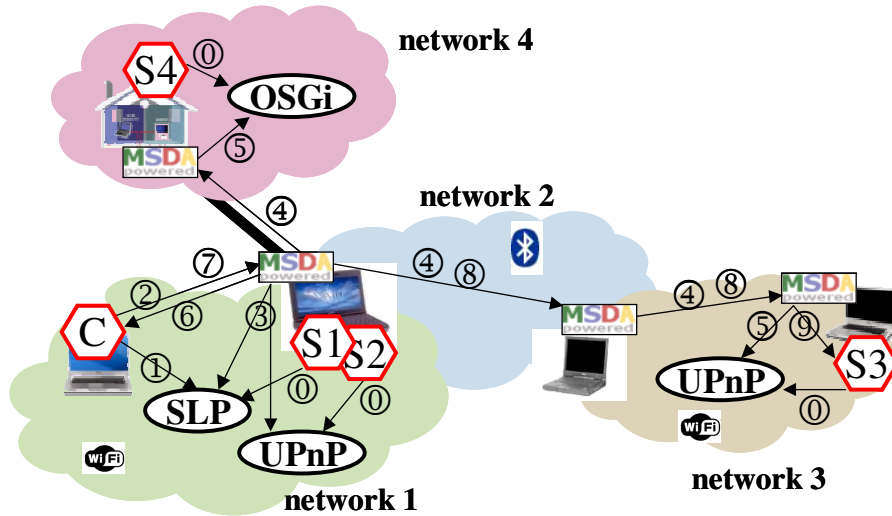


Figure 3: Prototype infrastructure

In our prototype, three independent wireless networks coexist at the location: two WiFi networks (network 1 and network 3) and a Bluetooth network (network 2). The home network (network 4) is accessed remotely through the Internet. Two devices are acting as MSDA Bridges (one between networks 1,2 and 4, and one between networks 2 and 3).

Each network assigns private IP addresses to its devices, and IP routing is not supported. We assume that all services have already registered (step 0) in their respective discovery domains and that the MSDA components have already been started.

Demonstration of MSDA capabilities is achieved through the following steps:

1. The client application issues an SLP service discovery request for the MSDA Service and gets its description.
2. The client issues an MSDA service discovery, searching for services accepting an RFID.
3. The MSDA Manager forwards the request to the local SLP and UPnP Plugins that return the MSDA Descriptions for S1 and S2.
4. The MSDA discovery request is forwarded to the remote MSDA Managers for networks 2, 3, and 4.
5. The MSDA Manager for network 3 forwards the request to the local UPnP Plugin that returns the MSDA Description for S3, and the MSDA Manager for network 4 forwards the request to the local OSGi Plugin that returns the MSDA Description for S4.
6. The MSDA Manager for network 1 collects the results and return them to the client.
7. The client issues an MSDA service access request for S3.

8. The service access request is propagated to the MSDA Manager for network 3.
9. The MSDA Manager for network 3 forwards the request to the MSDA Service that execute the service request. The result is then sent back to the client following the same path.

4. Acknowledgements

This work has been done within the project "UBISEC", running under the 6th framework program performed by Information Society Technologies (IST) under the contract n° 506926.