Pervasive Spontaneous Composition

André Bottaro, Anne Gérodolle, Philippe Lalanda
PhD candidate Grenoble University

The present document contains information that remains the property of France Telecom. The recipient’s acceptance of this document implies his or her acknowledgement of the confidential nature of its contents and his or her obligation not to reproduce, transmit to a third party, disclose or use for commercial purposes any of its contents whatsoever without France Telecom’s prior written agreement.
Outline

- Pervasive Computing
- Hiding complexity
- Realisation
Pervasive Computing
Pervasive Computing

Mark Weiser in the 90s
- Numerous and numerous smart networked devices
- Natural user interfaces
  => Fading into the background (the factory metaphor)

Interconnection and self-organisation
- Device discovery and description
- Distribution and mobility
- Location-awareness and context-awareness
Our Vision

Provided and required features

Provided and required resources

Standard devices

Local and remote Discovery / Communication Middleware

Plateform
Software engineering

- **Object orientation**
  - Clear intracomponent architecture
  - Evolutivity

- **Component orientation**
  - Clear architecture
  - Separation of concerns

- **Service orientation**
  - Service abstraction above a component implementation
  - Service discovery with provider/requester/registry model
Hiding service availability management

**OSGi R4 Declarative Services**

- Fine grain component above OSGi bundle model
- Self-description: service dependency description
  - Developers do not write error-prone code any more
Hiding distributed protocol heterogeneity: Service Discovery protocols

Many existing technologies

- on IP networks
  - UPnP (SSDP, GENA, SOAP, etc)
  - Bonjour (DNS-SD, mDNS)
  - DPWS (SOAP, WSDL, ws-*)
  - IGRS (SSDP+, GENA+, SOAP, WSDL, etc.)
  - CORBA (IIOP, CORBA Services)
  - Jini
  - SLP
  - Salutation
  - ...

- And on ad hoc networks ...
  - ZigBee
  - Bluetooth SDP
  - ...

Hiding distributed protocol heterogeneity: Service Discovery protocol generic layers

- Addressing layer (e.g. DHCP, AutoIP)
- Naming layer (e.g. DNS, mDNS)
- Discovery layer (e.g. SSDP)
- Description layer (e.g. WSDL)
- Control layer (e.g. SOAP)
- Eventing layer (ws-eventing)
- High-level layers: Presentation, Security, QoS, etc.
**Lookup Service**

One unified interface to access external service directory

- String registerService(Service s)
- void unregisterService(String uuid)
- Service[] lookup(String interFace, String LDAPFilter)
- void addListener(RemoteServiceListener)

Various possible instances

- SLPLookupService Directory Agent or multicast requests
- SSDPLookupService multicast requests
- WSDLookupService Discovery Proxy or multicast requests
- ...
Export / Binding

- soap stub
- soap binding service
- soap proxy bundle
- soap binding "client"

- rmi stub
- rmi binding service
- rmi proxy bundle
- rmi binding "client"

- rmi skel.
- exportable service
- rmi exportservice
- soap servlet
- rmi export
- soap export
- "server"
Discovery infrastructure

- Pervasive Service Binder or Discovery Base Driver
- Lookup Service
- Export Factory
- Binding Factory
- Export-Binding
- Discovery

OSGi Framework
An extended Service Binder

Pervasive Service Oriented Middleware
- Adapt to dynamic service availability
- Adapt to Service Discovery protocol multiplicity
- Make the use of local and distant services transparent and keep priority on local binding

Technological choice and achievements
- OSGi R4 Declarative Services (Service Binder)
- Extension: distribution with standard protocols
  - Architecture with plugins (generic interfaces)
    - Discovery: WS-Discovery, SSDP, SLP
    - Communication: SOAP, RMI
Declarative description

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bundle SYSTEM "metadata-amicom.dtd">
<bundle>
  <component class="com.francetelecom.amicom.app.AudioComposer">
    <requires service="com.francetelecom.AudioPlayer"
      filter="(location=*)"
      cardinality="1..1"
      policy="dynamic"
      bind-method="bindAudioPlayer"
      unbind-method="unbindAudioPlayer"
      local-only="false"
    />
  </component>
</bundle>

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bundle SYSTEM "metadata-amicom.dtd">
<bundle>
  <component class="com.francetelecom.amicom.impl.AudioPlayerImpl">
    <provides service="com.francetelecom.amicom.AudioPlayer"/>
    <property name="plateforme" value="g-babord" type="string"/>
    <onregister registry="*"/>
  </component>
</bundle>
```
Realisation
Technical architecture

provided and required services

provided and required packages

SLP, WS-Discovery, SSDP, etc.
RMI, SOAP, etc.

standard devices

extended SB

extended SB

extended SB

extended SB

OSGi

OSGi

OSGi

OSGi

france telecom
research & development

distribution of this document is subject to france telecom’s authorization
D17 - 29/06/2006
Follow-Me application

- Pervasive Service Binder
- OSGI Framework
- Device in the kitchen
- B-Light Client
- Pervasive Service Binder
- B-Light Server
- OSGI Framework
- C
- S
- Loc
- Device in the room
- Device in the corridor
- Device in the living room
- Device in the hall

Paul: move

Distribution of this document is subject to France Telecom’s authorization
D18 - 29/06/2006
Acknowledgements

- RNRT Pise project

Thanks for your attention