

# A UPnP™ profile to manage advanced platforms 'a la OSGi™'

January 29th, 2008

Andre Bottaro, Maxime Vincent



research & development



# Summary

- 1 Home device management today
  - Market pressure boosts standardisation
  - Device Management protocols and their limitations
  - Embedded execution platform evolution
- 2 UPnP™ assets in the remote management picture
  - Remote management functions
  - Remote management scalability
  - Home dynamicity management
  - Home market trends
- 3 UPnP™ Execution Platform Working Committee
  - A story beginning in the OSGi™ community
  - Telecom operators in the UPnP™ Forum
  - Technical objectives
  - Ongoing discussions on a UPnP™ bridge in the DSL Forum and the HGI
- 4 A straightforward profile for OSGi™ platform management on the Home Network
  - OSGi™ assets
  - RFP 101 and ongoing REG technical answer
  - A possible UPnP™ profile
- 5 An OSGi™ prototype
  - An OSGi™ client implementation
  - An OSGi™ device implementation
  - Ongoing thoughts about an implementation on a .NET platform
- 6 Conclusion  
References

# 1

## Home Device Management today

# Market pressure boosts standardization efforts

## ■ Trends

- Number of networked devices at home
- User demand for self-care and remote assistance



## ■ Telecom operators participation in standardization efforts

- DSL Forum: TR-069 spreading in Home device management servers
- OMA DM: clients installed on phones. Phones coming into the home.
- HGI (Home Gateway initiative): shaping the DM big picture
- OpenIPTV: looking for the right DM standard
- UPnP™ Execution Platform Working Committee: A profile for DM
- OSGi™ Residential EG:
  - TR-069 Device Management Tree,
  - Reference architecture for device management from distinct service providers

# Device Management protocols and their limitations

## ■ DM protocols

- Client-server protocols
- Examples: TR-069, OMA DM, SNMP, JMX, etc.
- Main issue: Scalability. Few methods, rich device management tree.

## ■ Limitations

- No adaptation for self-care and autonomous network scenarios
- Home dynamicity ignored (disconnections, dynamic state monitoring)
- No bridge towards other home device control protocols

# Embedded execution platform evolution

- Embedded environments
  - Operating systems: Linux, Windows Mobile, Symbian, VxWorks, etc.
  - Virtual Machines: Java (CLDC / CDC), .NET, etc.
- Platform-specific definitions
  - Software entities, e.g., firmware, modules, services
  - A lifecycle for every entity and related actions, related events
  - A security model: user authentication, software validation, access rights
- Classic Device Management today
  - Firmware upgrade and reboot actions
  - Device description and configuration: Describe rich (extensible) device data model
  - Basic diagnostic: Monitor device connectivity and usage
- Advanced Device Management tomorrow
  - Modular Software deployment
  - Modular application lifecycle management
  - Dynamic device discovery, instant diagnostic and continuous monitoring

# 2

## UPnP™ assets in the remote management picture

# Device Management functions

- Asset inventory
  - Device discovery and description
- Provisioning & Configuration
  - Device configuration actions (get/set)
  - Provisioning: Initial configuration
  - Backup and restore actions
- Firmware & Software
  - Software lifecycle management actions
- Device Monitoring, Diagnostic, Tracking
  - Device numbers monitoring (get/listen): CPU, battery, signal strength...
- Security
  - DM server authentication, access right management, transferred content validation

# Remote management scalability

- 3 main actors
  - Devices: network auto-organization
  - Users: self-care scenarios
  - Remote DM servers: Operator maintenance
  
- Objectives of operators and service providers
  - Avoid manual installation and maintenance
    - Simplicity for the user => make the devices smarter
    - Avoid costly hotline calls => provide simple user tools to diagnose, repair their network
    - Avoid costly manual tasks in sales shop => spread management protocols on devices
  - Increase Remote Management servers scalability
    - Keep most of actions on the user side => bridge WAN protocols with LAN protocols
  
- UPnP™ assets
  - Plug-n-Play mechanisms makes the devices smarter
  - UPnP™ simplifies user services and tools
  - UPnP™ technology is already in the Home

# Home dynamicity management

- Objectives of operators and service providers
  - Instant diagnostics on running devices
  - Continuous performance monitoring
  
- UPnP™ assets
  - UPnP™ provides dynamic device inventory on the LAN
  - UPnP™ enables instant diagnosis and continuous monitoring
  - UPnP™ Forum standardizes home device features (AV, IGD, etc.)

# Home market trends

- Objectives of operators and service providers
  - Benefit from the user demand for networked services
  - Insure QoS and service evolution
  
- UPnP™ asset: UPnP™ is already in the Home
  - Media devices market:
    - Most of network attached storages (NAS) and Game consoles
    - some players, recorders and video cameras.
  - IP connectivity market: Most of Home Internet gateways
  - Imaging market: some printers and scanners.
  - Certification
    - 233 certified devices: <http://www.UPnP™-ic.org/certification/default.asp>
    - UPnP™ AV certification, 1st step of DLNA certification process

# 3

## UPnP™ Execution Platform Working Committee

# A story beginning in the OSGi™ community

- 2004: RFP 49 A UPnP™ profile for the OSGi™ platform
  - Authors: Jooyeol Lee, Kiran Vedula, Jae Lee, Hyun-Gyoo Yunk, Samsung, Peter Kriens, aQute.
- 2005: An OSGi™ prototype in the ObjectWeb oscar community
  - Author: Didier Donsez, University Joseph Fourier.
- 2006: some email discussions
- March 2007:
  - First audio calls: Jooyeol Lee, Andre Bottaro, Didier Donsez, Kai Hackbarth
  - Main topic: Service lifecycle management
- June, 28<sup>th</sup> 2007: Technical presentation to the UPnP™ SC, Munich
  - Entering the UPnP™ incubation process: New Projects sub-committee
  - New attendees and enlargement of the topic in the general Device Management domain
- October 13<sup>th</sup> 2007: Execution Platform Working Committee creation approval
  - Use case selection and Technical requirements phase
  - An active Working Committee: 12 active companies, 1 audio call/week, 1 F2F meeting/2 months

# Technical objectives insert UPnP™ in the Remote Management picture

- Specify a UPnP™ profile for classic Device Management
  - Basic diagnostic
    - Monitor device connectivity and usage
  - Device description and configuration
    - Describe rich (extensible) device data model
  - Firmware upgrade and reboot actions
  
- Go over today Device Management limits
  - Dynamic device discovery
  - Modular Software deployment and lifecycle management
  - Instant diagnostics and continuous performance monitoring

# Hard technical issues

## ■ Genericity

- Configuration model agnosticism to existing DM data models
- Modular lifecycle management genericity

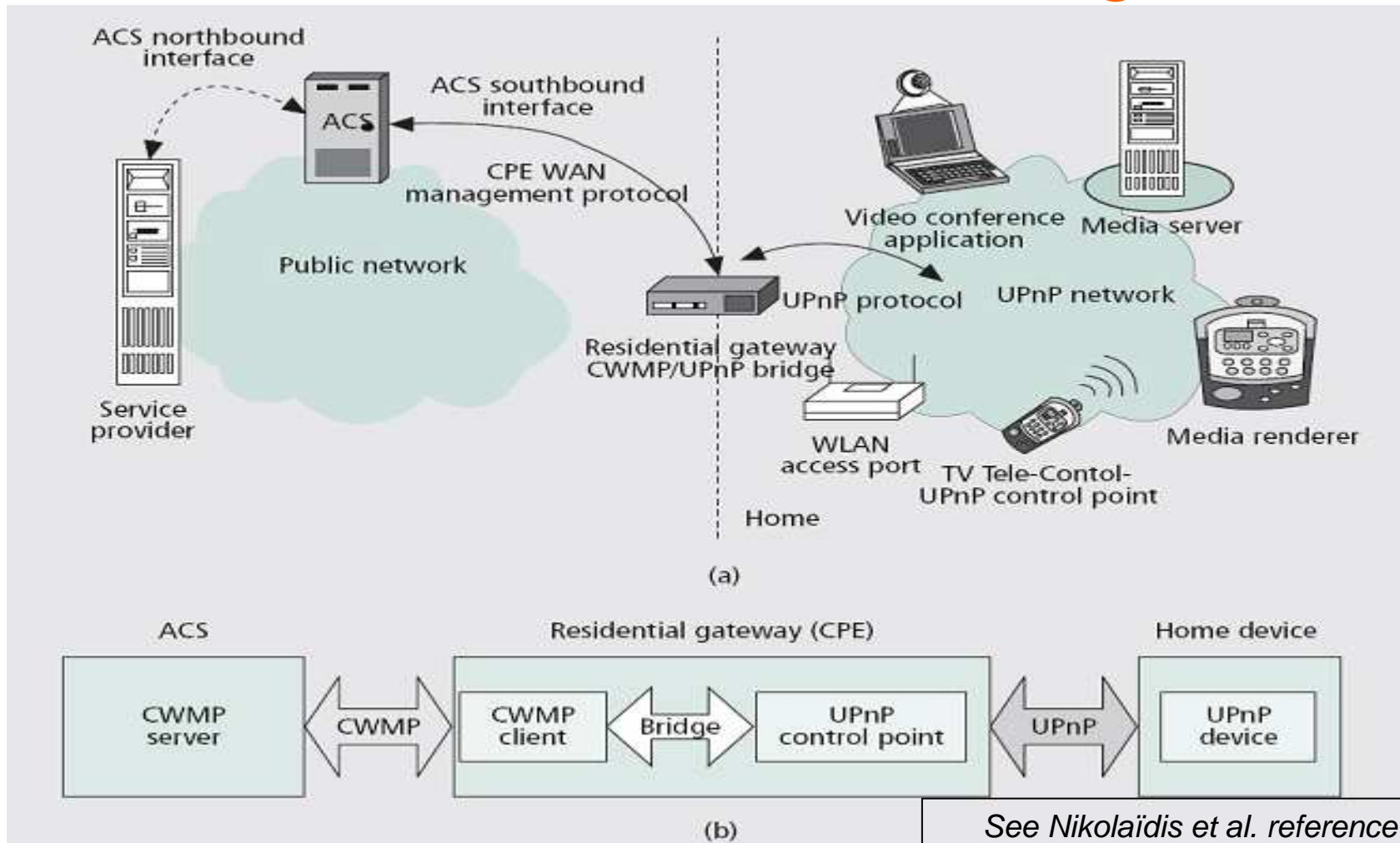
## ■ Security

- Device authentication, message integrity and confidentiality.

## ■ Distributed deployment consistency

- Configuration consistency of related managed devices on a local network

# A profile complementary to a DSL Forum technical bridge



# 4

A straightforward profile for OSGi™ platform management

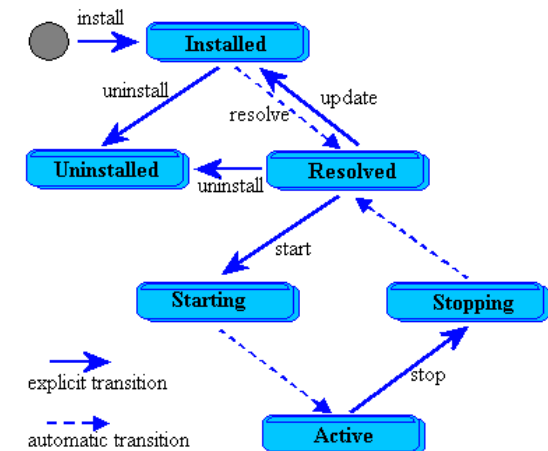
# OSGi™ assets

## ■ A generic deployment model in the core specification

- Defined deployment units with dependencies
- Defined lifecycle management
- Security, eventing, configuration

## ⇒ An ideal UPnP™ Execution Platform:

- ⇒ The bundle: the manageable software entity
- ⇒ An acceptable minimal set of management actions
- ⇒ Security, Eventing, Configuration: matching features



# RFP 101: An ongoing REG technical answer

- REG: Residential Expert Group
  - Telecom operators, manufacturers and software editors
  - Created at the end of 2007
  - RFPs available during the creation phase
- RFP 101: The OSGi™ platform as a UPnP™ device
  - RFP 49 rewritten with new ideas and references to OSGi™ R4 chapters
  - Ongoing RFC edition in the Expert Group

# A possible UPnP™ profile

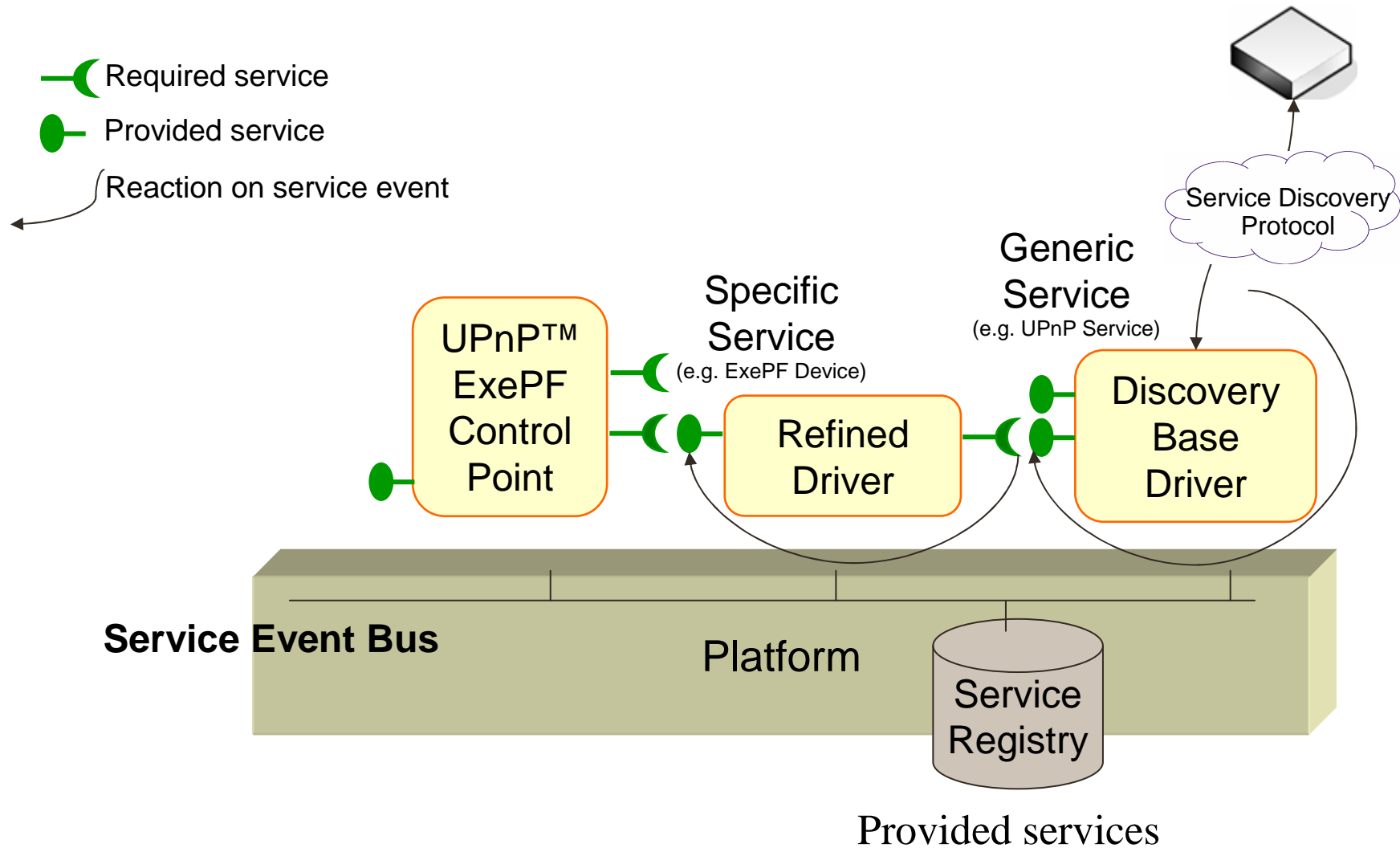
- Written in Java format and deduced from the technical objectives
  - Module lifecycle management related actions
    - **void** installModule ( **string** URN, **string** optionalParameters )
    - **void** updateModule ( **integer** moduleId, **string** optionalNewURN )
    - **void** startModule ( **integer** moduleId )
    - **void** stopModule ( **integer** moduleId )
    - **void** uninstallModule ( **integer** moduleId )
    - **string** getModuleState ( **integer** moduleId )
    - **string** getModuleInformation ( **integer** moduleId )
    - **string** getModuleList ( )
  - Platform related actions
    - **void** updateFirmware ( **string** URN )
    - **string** getCapabilities ( )
    - **string** getParameter ( **string** parameterName )
    - **void** setParameter ( **string** parameterName, **string** value )

# 5

## An OSGi™ prototype

# Implementing a UPnP™ OSGi™ client

# Service proxies populating the registry



# Refined transparency

## ■ UPnP XML Description

```
<action>
  <name>Play</name>
  <argumentList>
    <argument>
      <name>InstanceID</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_InstanceID
      </relatedStateVariable>
    </argument>
    <argument>
      <name>Speed</name>
      <direction>in</direction>
      <relatedStateVariable>
        TransportPlaySpeed
      </relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

## ■ Writing a control point with the Base Driver API

```
Dictionary dictionary = new Hastable();
Dictionary.put("InstanceID", "0");
Dictionary.put("Speed", "1");
upnpDevice.getDescription(null)
    .getService("AVTransportService")
    .getAction("play").invoke(dictionary);
```

## ■ Writing a control point with a Specific Driver API

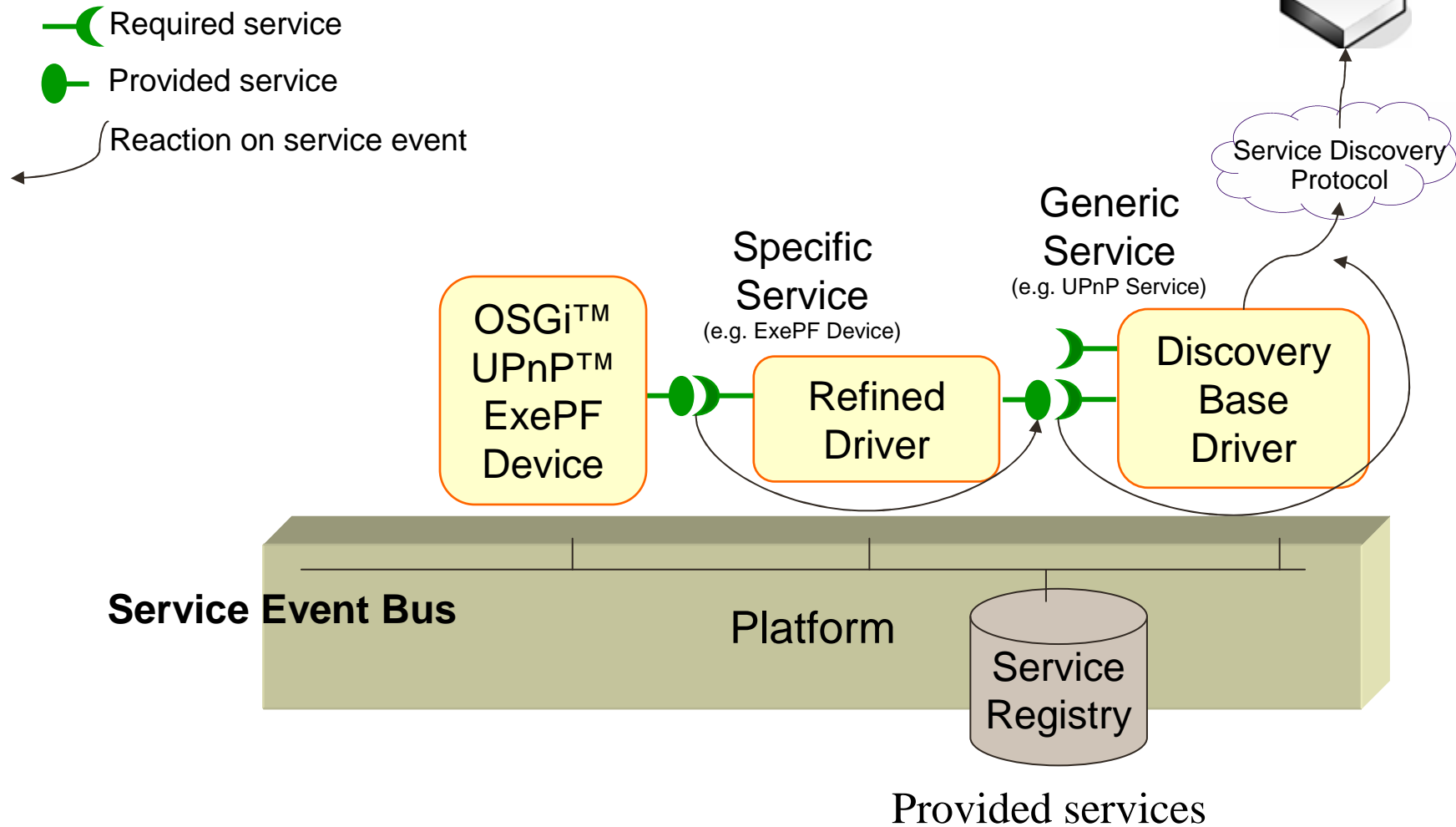
```
mediaRenderer.getAVTransportService()
    .play(0,1);
```

## ■ Generated specific Driver API

- FT, UJF, Siemens or other tools
- See OSGi RFP 72 – Extended Mapping for UPnP Discovery Transparency

# Implementing a UPnP™ manageable OSGi™ server

# Exporting an OSGi™ service into a UPnP™ device



# Device Implementation

## ■ UPNP XML Description

```
<action>
  <name>getModuleState</name>
  <argumentList>
    <argument>
      <name>moduleID</name>
      <direction>in</direction>
      <relatedStateVariable>A_ARG_TYPE_ModuleID
      </relatedStateVariable>
    </argument>
    <argument>
      <name>moduleState</name>
      <direction>out</direction>
      <!--<retval/>-->
      <relatedStateVariable>A_ARG_TYPE_ModuleState
      </relatedStateVariable>
    </argument>
  </argumentList>
</action>
```

## ■ Generated Interface

```
public interface IModulesAdminService {
    public String getModuleState(int moduleID) throws
        Exception;
    ...
}
```

## ■ Device implementation

```
public class DeviceImpl implements IModulesAdminService {
    public String getModuleState(int moduleID) throws Exception {
        try {
            Bundle[] bundles = bundleContext.getBundles();
            ...
        }
    }
}
```

# Implementing a UPnP™ manageable .NET server

# .NET™ entities

## ■ Assembly

- .NET™ basic deployment units
- Classes, resources & manifest file
- A process or a library
- Sharing and isolation features
- Possible management actions :
  - Install/uninstall from the GAC (Global Assembly Cache)
  - Load in application
  - Cannot be unloaded independently from its application domain

## ■ Application Domains

- .NET™ intra-application isolation concept
- Allow loading assemblies dynamically
- Communication intra application Domains needs network mechanisms
- Possible management actions
  - An application could create as many Application Domains as it wants
  - Could be destroyed (Only way to unload assembly)

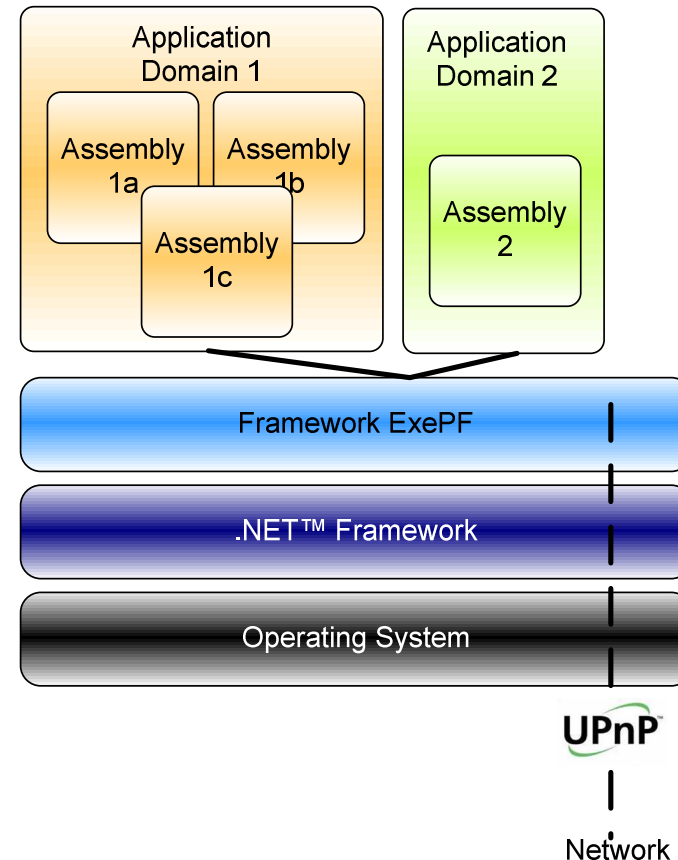
# Ongoing thoughts about an implementation on .NET platform

## ■ Current suggestions

- 2 module types with distinct available management actions
  - Assembly
  - Application domain
- An OSGi-like platform above .NET for module lifecycle management (see excoffier et al. reference)

## ■ Using Intel® UPnP™ tools

- UPnP™ control API on the .NET™ framework
- Automatic generation of .NET™ classes from xml service descriptors



# 6

## Conclusion

# Conclusion

- Remote Management: an important topic today
  - Protocols: DSL Forum TR-XXX, OMA DM, OpenIPTV, ...
  - Platforms: JCP JSR 232, JSR 277, JSR 320, OSGi™ Residential Expert Group, ...
  
- UPnP™: a relevant protocol set for device management
  - Management scalability: Self-care scenarios and autonomous networks
  - Home context dynamicity: instant diagnostics and continuous monitoring
  - Home market: Home device control protocols today dominated by UPnP.
  
- OSGi: a relevant model of a Home advanced platform
  - Bringing a generic model for modular applications with a flexible lifecycle
  - Shaping an open Remote Management Reference Architecture

# Questions ?

# References

- André Bottaro, Anne Gérodolle, Philippe Lalanda, *Pervasive Service Composition in the Home Network*, 21st Int. IEEE Conference on Advanced Information Networking and Applications (AINA-07), Niagara Falls, Canada, May 2007
- OSGi™ Alliance, *RFP 72 Extended Mapping for UPnP Discovery Transparency*, April 2006. Author: André Bottaro.
- OSGi Alliance, *RFP 101 The OSGi service platform as a UPnP device*, October 2007. Authors : André Bottaro, Peter Kriens, Jae Shin Lee, Kiran Bharadwaj Vedula, Hyun-Gyoo Yook
- Apostolos E. Nikolaidis, Serafeim Papastefanos, Gregory A. Doumenis, George I. Stassinopoulos, Marios Polichronis K. Drakos, *Local and Remote Management Integration for Flexible Service Provisioning to the Home*, IEEE Communications Magazine, Oct. 2007
- Clement Escoffier, Didier Donsez, and Richard S. Hall, *Developing an OSGi-like Service Platform for .NET*, IEEE Consumer Communications and Networking Conference, 2006

# Backup