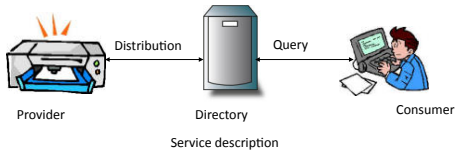
Distributed Systems Challenges: Service Discovery

Overview

- Key functions in peer-to-peer networking
  - Automatic configuration
  - Service discovery
  - Service delivery
- Examples of standards and systems for service discovery and delivery
  - Jini, Service Location Protocol (SLP), and UPnP

What is Service Discovery?



Service vs resource discovery

Service Discovery Scenarios (1)

- You are in a taxi without your wallet…
  - Your cell phone "sees" a service of the Anytown Cab Service
  - The phone "downloads" an electronic payment application to your cell phone that you use to pay your fare
- You bring a handheld to a client site to show your company's new products and options…
  - Your handheld "discovers" a printer service when you step into the client's office
  - You are able to print the document that is customized to your client's needs

Service Discovery Scenarios (2)

- You bring an iPAQ loaded with a PowerPoint file to a meeting…
  - Your iPAQ "discovers" that there is an intelligent projector in the room
  - You "upload" the PowerPoint file stored on your iPAQ to the projector
  - You start and control the presentation from your iPAQ

Service Discovery Scenarios (3)

- You bring an iPAQ to a meeting of folk musicians…
  - Your iPAQ "discovers" audio files on other devices that are available
  - You retrieve the files of the musician that you just heard perform
- You bring your digital camera home from a trip…
  - The camera "discovers" a photo-quality printer
  - The camera sets the printer's parameters and prints pictures that you selected

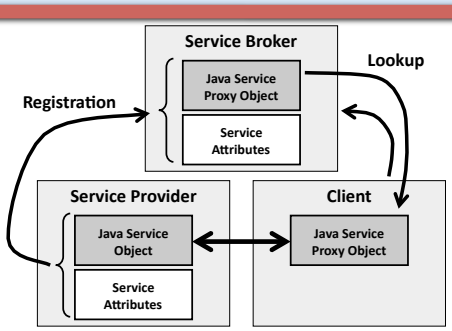
Key Functions for P2P Computing (1)

- Auto-configuration
  - The device must configure itself to participate in a network
  - IP address assignment
    - Use of Dynamic Host Configuration Protocol (DHCP)
    - Selection of an address from the reserved link-local block 169.254.0.0/16
    - Explicit auto-configuration support in IPv6 with link-local addressing

Key Functions for P2P Computing (2)

- Service discovery
  - A server must be able to advertise its services
    - Service
    - Invocation and interfaces
  - A device must be able to locate services
  - Reliability and scalability are important issues
  - Three examples: Jini, SLP, and UPnP
- Service delivery
  - Device must be able to communicate with server, invoke service, pass parameters to the service, and receive reply from the service
  - Authentication and authorization are often issues

Sun Microsystem's Jini (1)



Sun Microsystem's Jini (2)

- A service provider discovers one or more lookup services (brokers)
- A service provider registers a service object (Java) and its service attributes with the lookup service (broker)
- A client requests a service from service attributes and a copy of the service object moves to the client at runtime
- The lookup service can notify clients registered in a service when there is a state change in the service
- The client interacts directly with the service via the service object

Reliability in Jini (1)

- Network failures are handled by leases
  - Jini leases a resource to a client for a fixed amount of time
  - After the lease period expires, the client must renew the lease to continue accessing the service
  - The lease automatically expires for all users when a server goes down or is removed from the network

Reliability in Jini (2)

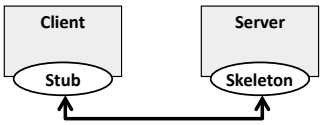
- Jini supports redundancy in the infrastructure and resilience against failure
  - The network may have several lookup services
  - Service providers register their proxy service objects with all lookup services they can discover using the discovery protocol
  - Clients can obtain a proxy object of the desired service from any of the lookup services with which the service provider has registered

Scalability in Jini

- Scalability is addressed by "communities" or "federations"
  - Groups of Jini services can form a community
  - Jini communities can link together into a larger group
  - The lookup service of a community can register in other communities

Other RPC Models (1)

- Common Object Request Broker Architecture (CORBA) and Distributed Component Object Model (DCOM) use a remote procedure call (RPC) model
  - Require a client-side "stub" process and a server-side "skeleton" process
  - Stub and skeleton agree upon a protocol for information exchange – set at compile time



Other RPC Models (2)

- The client-server interface is defined by a language-neutral language
  - Interface definition language (IDL)
- An RPC compiler is used to automatically generate client stub and server skeleton
- Tight coupling between client stub and server skeleton is a major disadvantage
  - Stub and skeleton are jointly developed – simultaneously created by the RPC compiler
  - Any design change in one must be reflected in the other

Jini's Dynamic Service Proxy Object

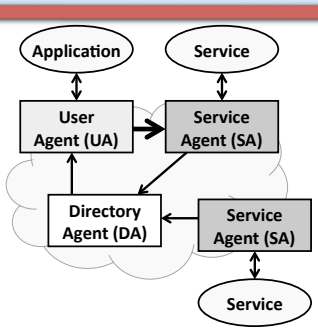
- Jini overcomes the disadvantage of the compile-time RPC stub/skeleton creation by allowing a client to obtain the stub from the server at runtime
- The stub is the service proxy object that the server uploads to the lookup service
- The proxy service object lets clients use the service without knowing anything about it
  - For example, a user interface (UI) can be embedded in the code, no need for a device driver if the service provided is a device

Additional Capabilities of Jini

- Remote events
  - Notify an object when specified changes occur in the system
  - Can be triggered by newly published services or state changes in an existing service
- Transactions
  - Provide a way to implement atomic commitment

Service Location Protocol (SLP)

- SLP is the Internet Engineering Task Force's approach to service location
- SLP makes services visible via "service URLs"



Three Agents in SLP

- Service agent
  - Broadcasts advertisements
  - Registers advertisements via service URLs
- Directory agent (optional)
  - Caches advertisements
  - Processes discovery queries from user agents by returning URLs of matched service agents
- User agent
  - Discovers services
  - Browses and selects services using URLs returned

Universal Plug-and-Play (UPnP)

- Microsoft-initiated standard
- Objective is…
  - To enable the advertisement, discovery, and control for networked devices, services, and consumer electronics in ad hoc environments
- UPnP leverages
  - UDP and the TCP/IP protocol suite
  - HTTP
  - XML and SOAP

A UPnP Device Can…

- Dynamically join a network
- Obtain an IP address
- Convey its capabilities on request
- Learn about the presence and capabilities of other devices
- Dynamically leave a network

UPnP Supports…

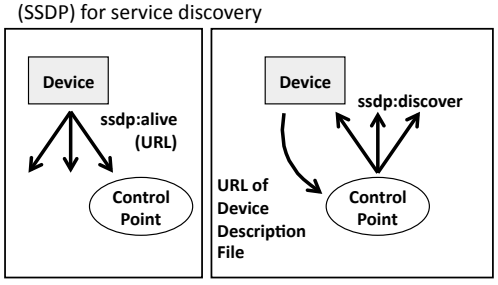
- Automatic configuration of IP
- Service discovery
- Service description
  - XML-based
- Service control
  - SOAP-based
- Eventing
  - Generic Eventing and Notification Architecture (GENA)
- Presentation
  - HTML interface

IP Addressing in UPnP

- UPnP uses Auto IP to let a device join a network without any explicit administration
- When a device connects to a network, it tries to acquire an IP address from a DHCP server, if one exists
- In there is no DHCP server, an IP address is claimed automatically from a fixed, reserved range for local network use
  - An IP from the link-local range (169.254.0.0/16) is randomly selected
  - An Address Resolution Protocol (ARP) request is sent to see if anyone has claimed the address

Service Discovery in UPnP (1)

- UPnP uses the Simple Service Discovery Protocol (SSDP) for service discovery



Service Discovery in UPnP (2)

- A device (e.g., a capable projector) can multicast an advertisement message (ssdp:alive) to advertise its services to control points (e.g., iPAQs)
- A control point (e.g., a device looking to use a projector) can multicast a search message (ssdp:discover) to the network
  - Any device that hears the multicast message can respond by replying with a unicast response message
- The URL of the XML Device Description File is returned to the control point

Service Description in UPnP

- UPnP uses XML for service description
- An advertisement message contains a URL that gives the address URL of an XML Device Description File
- Device Description File describes the advertised device's capability
- A control point can retrieve and inspect an advertised device's XML device description file using HTTP
- A device may provide multiple services

<service> Element

- The service type
- The service ID
- The service address URL for invoking the service via SOAP
- The event subscription URL for subscribing to event notifications
- The Service Description File of the service for describing more specific details of the service that is provided

Device Description File for a Projector

projector-desc.xml

```
<?xml version="1.0" ?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
<device>
 <deviceType>urn:schemas-upnp-org:device:projector:1</deviceType>
 <UDN>uuid:UPnP-Projector</UDN>
 <serviceList>
 <service>
 <serviceType>urn:schemas-upnp-org:service:control:1</serviceType>
 <serviceId>urn:upnp-org:serviceId:control</serviceId>
 <controlURL>isapictl.dll?control</controlURL>
 <eventSubURL>isapictl.dll?control</eventSubURL>
 <SCPDURL>projector-scpd.xml</SCPDURL>
 </service>
 </serviceList>
</device>
</root>
```

Service Description File for the "projector control" service

Service Control in UPnP (1)

- The XML service description file of an advertised service (e.g., the "projector control" service) contains the following information
  - An action list listing actions to which the service will respond
  - A service state table listing the set of state variables (and their data types) representing the service state at runtime

Service Control in UPnP (2)

- To invoke a specific service control advertised by a device, a control point (e.g., an iPAQ controlling the projector) sends a SOAP message to the service at the specified service address URL
  - The control point can query or update state variables listed in the state service table
- The advertised service then executes the specified control action and returns action-specific values to the client via a SOAP message

Service Description File for Projector Control Service (1)

projector-scpd.xml

```
<?xml version="1.0" ?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
<actionList>
<action>
 <name>SetPower</name>
   <argumentList>
    <argument>
        <name>Power</name>
        <relatedStateVariable>Power</relatedStateVariable>
        <direction>in</direction>
    </argument>
   </argumentList>
</action>
 … Other actions …
</actionList>
```

Service Description File for Projector Control Service (2)

```
…
<serviceStateTable>
  <stateVariable sendEvents="yes">
        <name>Power</name>
        <dataType>Boolean</dataType>
        <defaultValue>0</defaultValue>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>File</name>
    <dataType>string</dataType>
    <defaultValue>default.ppt</defaultValue>
  </stateVariable>
 … Other state variables …
</serviceStateTable>
</scpd>
```

UPnP Event Subscription Services

- A control point can "subscribe" to receive an event notification message from an advertised service (e.g., the "projector control" service) when state variables change their values
  - The subscription URL is given in the device description file
- These event messages are expressed in XML and formatted using the General Event Notification Architecture (GENA)

UPnP GENA Example

- For example, the projector service can send an event message to a control point that subscribes to the notification service when one of the following events occurs
  - Page up/down (changes the variable pageNumber)
  - Power on/off – turning on/off the projector emulator
  - Files – list of PowerPoint presentation files
  - File – current PowerPoint file for presentation
- UPnP does not allow a control point to subscribe to individual state variables, so a control point has to determine which state variable has been changed when receiving an event notification message

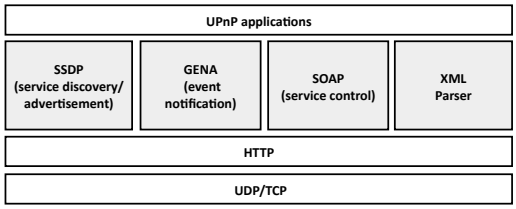
UPnP HTTP-based Presentation (1)

- A device can advertise a user-interface presentation URL from which the control point can access services provided by the device such as…
  - Retrieve a page from the URL
  - Load the page into a browser
  - Let the user control the device
  - View the device's status

UPnP HTTP-based Presentation (2)

- Because of the use of XML for data definition and exchange, a UPnP service *potentially* can deal with a wide variety of small devices (e.g., WAP, i-mode, iPAQ, etc.) as control points by performing XML transformation based on XSLT
  - Currently UPnP supports only HTML browsers

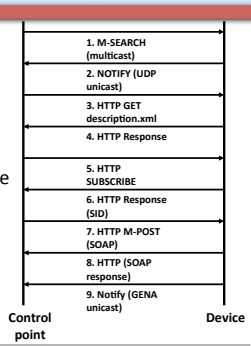
UPnP Architecture



- SSDP uses HTTP over multicast and unicast UDP
- SOAP and GENA use HTTP over TCP

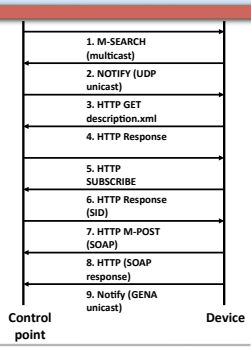
Service-Control Point Interaction (1)

1. The control point sends out a SSDP search request
2. The device issues a unicast UDP NOTIFY response with the URL to the device's XML description document
3. The control point requests the XML description document via HTTP
4. The web server contained in the device responds to the request and returns the XML description document



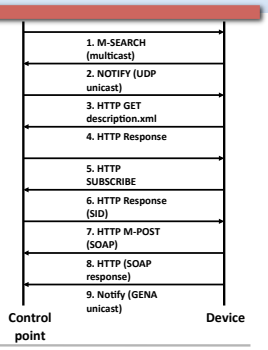
Service-Control Point Interaction (2)

5. To receive automatic notifications of changes in the device, the control point subscribes to the services in which it is interested via HTTP
7. The device acknowledges the subscription request and returns a unique Subscription Identifier (SID)



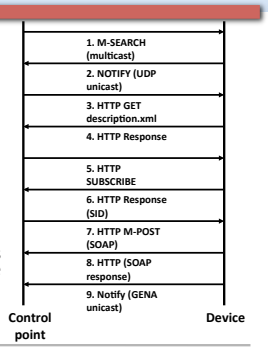
Service-Control Point Interaction (3)

7. The control point can instruct the device to perform actions by changing one of the state variables
   - The URL to send control requests is contained in the device's XML description document
   - The control point issues a SOAP action over HTTP



Service-Control Point Interaction (4)

8. The device changes the state of the internal variable and issues a SOAP response message
9. The device can notify clients of changes in its state either because of explicit actions (step 8) or implicit changes in the device itself
   - Device notifies all subscribers via a unicast NOTIFY message over HTTP



Middleware Functionality in UPnP (1)

- Service Discovery
  - UPnP provides functionality for service discovery in P2P environments
- Data Transformation
  - Advertised service can potentially perform data transformation between XML and other control-point-specific languages

Middleware Functionality in UPnP (2)

- Adaptability
  - IP addresses can be dynamically allocated
  - Changes of state information (contexts) of subscribed services are made known to mobile devices through event notifications
  - No support yet for service routing and selection based on the client's location
- Transparent support for communication
  - UPnP provides transparent P2P communication based on Internet standards
  - No multi-hop ad hoc communication support

Summary

- Basics of peer-to-peer networking
- Three example standards and systems for service discovery and delivery
  - Jini
  - Service Location Protocol (SLP)
  - Universal Plug and Play (UPnP)
- UPnP in more detail

Mainstream protocols

	Jini	**Salutation**	**SLP**	**UPnP**	**Bluetooth**
Main Entities	Lookup Service, Client, Service	Salutation Manager, Transport Manager, Client Server	Directory Agent, Service Agent, User Agent	Control Point, Devices (Services)	SDP Client, SDP Server (or both)
Service Repository	Lookup Service	A set of SLMs	DA (directory agent)	None	SDP Server
Service Announcement	Discovery/ Join protocol	Registering with local SLM	Service Registration	Multicast advertisement	Not Supported
Access to Service	Service proxy object based on RMI	Service Session Management	Service type for discovered service	Invoking Action to service	Not Supported
Service Description	Interface type and attribute matching	Functional Unit and attributes within it	Service type and attribute matching	Description in XML	Attribute ID and Attribute Value

Mainstream protocols

	Jini	Salutation	SLP	UPnP	Bluetooth
Service Group	Group	No	Scope	No	Service Class
Event Notification	Remote Events	Availability Checking (periodic & automatic)	SLP extension for event notification	Service publishes event when state variable changes	Not Supported
Other Features	Java-centric architecture	Transport independence	Authentication security feature	Automatic configuration	Services could be browsed from a hierarchy
Usage	CNN and sprint Web/ Directory Servers, E-mail, Calendar, Collaboration Servers		Novell Netware	WinXP for gateways, Internet connectivity and NAT	Bluetooth access points, print adaptors, Palm OS bluetooth system

Location-Aware Service Discovery

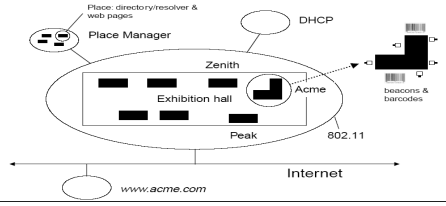
Where is the closest Italian restaurant to me?

- Location sensing
- Nomadic users
- Handy devices

CoolTown, Splendor, Agents2Go

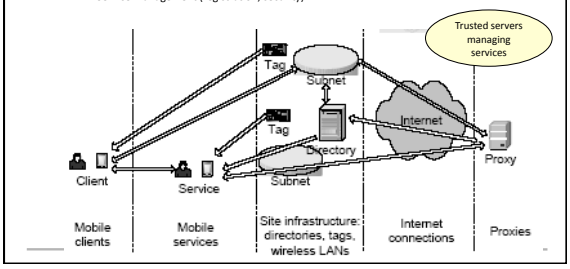
CoolTown

- Every service (e.g., printer) has a web server, and a tag.
- Users walk in and receive URL from tag beacons
- Connect to the URL which describes the service
- Place manager maintains resources in a place, with web interface (HTML and XML)
  - Resources can be grouped.
  - Also acts as a resolver for some places



Splendor

- Tag-based location aware
  - Tag label location and people
- Proxies
  - Enhanced security
  - Service management (registration, security)



Agents2Go