Discover Remote Services

ECF- Eclipse Communication Framework
Wim Jongman

ECF, e4 and Nebula

@wimjongman

wim.jongman@remainsoftware.com
Discover Remote Services

- OSGi Services
- Remoting Concepts
- Discovery
- Demo
- Price Ceremony
Everything you always wanted to know about OSGi services (but were afraid to ask)
OSGi whiteboard pattern recap

Service provider

Implicitly uses

API

Service consumer

Depends on
OSGi whiteboard pattern recap

Service provider

Looks up all Listeners of type X

Implicitly uses

Service Registry

Registers Listener of type X

Service consumer

(Type X is defined by provider or additional API bundle)
Fallacies of Distributed Computing

The network is reliable.
Latency is zero.
Bandwidth is infinite.
The network is secure.
Topology doesn't change.
There is one administrator.

“A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable”

[Lamport]
Remoting Concepts
Discovery
Time for a Demo
What have we cooked?

Chat Application
Server for Discovery (Zookeeper and DNS)
Remoting (R_OSGI, ECF Native)
ChatBot for some content

Two modes: Centralistic and p2p
# ECF Architecture

<table>
<thead>
<tr>
<th>local</th>
<th>discovery</th>
<th>manager</th>
<th>transport</th>
<th>remote</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File</td>
<td></td>
<td>r_osgi</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Zookeeper</td>
<td></td>
<td>ECF</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>DNS</td>
<td>RemoteService Admin (RSA)</td>
<td>JMS</td>
<td>Service-proxy</td>
</tr>
<tr>
<td></td>
<td>Bonjour</td>
<td></td>
<td>REST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SLP</td>
<td></td>
<td>XMPP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Demo
Demo
Instructions

git clone https://github.com/ECF/Chat
cd Chat
mvn verify
chat_win32_x64
Downloads

Chat Client: http://goo.gl/G3IH7f

Chat Server: http://goo.gl/bCSW3j
ECF’s asynchronous RS

package org.eclipse.ecf.example.chat.model;

import java.util.concurrent.Future;

import org.eclipse.ecf.remoteservice.IAsyncRemoteServiceProxy;

// Interface only needed on caller side
public interface IChatServerListenerAsync extends IAsyncRemoteServiceProxy {
    /**
     * Async for \{@link IChatServerListener#getHandle()\}
     */
    Future<String> getHandleAsync();
}
ECF’s RSA implementation

Passes OSGi R5 compliance tests*

* MOD known CT failures
Discover Waldo?!
Price Ceremony
THANKS!!!

PLEASE RATE THIS TALK
Distributed Eventing

Uses ECF’s distributed eventing to sync up servers

[http://blog.osgi.org/2013/06/distributed-eventing-rfp-158-now.html]
[http://wiki.eclipse.org/EIG:Distributed_EventAdmin_Service]