Services in a multi-device world
Why End to End Architectures matter
Presentation Format

- What are users looking for?
- An example user service; Navigation
- Where OSGi fits amongst all of this
- An example End to End Architecture – GST
- Using GST to implement the navigation example
What do users want?

- Users want to have their application data presented to them consistently on whatever devices they have available;
  - How many phone books do you have right now? (PC, phone, PDA, handwritten….)
  - Are you constrained to a single machine for reading and writing your e-mail?
  - ....
- They recognize that capabilities might be limited by the nature of the devices they have available (c.f. Don Norman and the Swiss Army Knife) but are happy to live with those constraints.
- Application data often needs to be processed locally before it can be presented to the user as useful information – we need an application platform as well as a data platform.
- They really don’t care about underlying platforms or technologies and, in fact, too much information about these scares them and switches them off!
Navigation as an example portable service
Implemented the ‘traditional’ way…
Implemented using portable services...
Things that help implement Service Portability…

• Common data representations
• Common management interfaces
• Common execution environments
• Dynamic collaboration
• Automatic deployment and provisioning

… all of these things, when commonly specified, contribute towards creating an End to End Architecture
Where does OSGi fit?

- OSGi provides an environment in which a significant proportion of the diversity of individual platforms is abstracted away – the same code can run in many environments.
- Code can be created for one environment and can still operate in others, despite the heterogeneity in the deployment.
- From a user perspective, code can migrate from one device to another, even though the reality (screen size, I/O devices etc.) is that the code being executed may change significantly.
What does OSGi Enable?

• A service platform where code can execute independently
• Application updates with no platform downtime
• Recognition of a dynamic world with capabilities entering and leaving the environment
• Implicit Multi-environment support (home, automotive, mobile etc. etc.)
An Example E2E Architecture : GST
(Work in Progress!)
What is GST specifying…?

<table>
<thead>
<tr>
<th>OSGi, .NET.....</th>
<th>Operating Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>C#, Java, C, C++...</td>
<td>Programmatic APIs</td>
</tr>
<tr>
<td>IETF RFCs etc.</td>
<td>Functional</td>
</tr>
<tr>
<td>HTTP, SOAP, ASN.1, CORBA...</td>
<td>Messaging</td>
</tr>
<tr>
<td>802.3, Bluetooth, ZiBee, WiFi, HSCSD, GPRS.....</td>
<td>Line Protocols</td>
</tr>
</tbody>
</table>

© copyright 2004 Telcordia Technologies Inc. All rights reserved.
The Goal of GST...
The GST System

Operational System
- Content Centre
- Service Centre
- Control Centre
- Client System
- User Credential Store
- Payment Centre
- Service Application
- Vehicle Management Centre
- Software Repository
- Billing Centre
- Telematic Control Unit
- I/O Device
- Service Platform
- Payment Service Centre
- Clearing Centre

Realization System
- Development Centre
- Certification Centre
- GST Standards and specifications

Entities in each system Interact with each other to animate
End to End Architecture of GST

- Service Centre
  - Service Consumption Interface (SC-I)
  - Service Authorization Interface (SA-I)
- Payment Centre
- Billing Centre
- Authentication & Authorization
- User Subscriptions

- IO Device
  - IO Device Interface (IOD-I)
- Telematics Control Unit
  - Vehicle Interface (V-I)
  - System Management Interface (UA-I)
- Vehicle to Vehicle Interface (VV-I)
- Vehicle
- Client System Management
  - User Authorization Interface (UA-I)
What it means in the real world…

Service Centre
- Service Creation
- Data Sources
- Administration

Control Centre
- Offline
- Realtime
  - Authentication
  - Payment and Billing
  - Subscription Management
  - Client System Management

Consumer
- Deployment
  - Rendering a service to the end user
  - Infrastructure Providers
  - Mobile Terminals
  - Vehicle Users
  - Emergency Services

© copyright 2004 Telcordia Technologies Inc. All rights reserved.
What’s Novel about an E2E Architecture?

• End to End Architectures require **multiple players** to collaborate to create an ecosystem...in the case of GST;
  – Car Manufacturers & their OEMs
  – Communication infrastructure providers
  – Mobile Handset manufacturers
  – ISPs and billing infrastructure providers
  – ...

• These ecosystems are most easily fostered either by the emergence of a leader creating de facto standards or via industry collaboration (OSGi, GST...)
Navigation: Route Planning Step

- Service Centre
- Payment Centre
- Billing Centre
- Authentication & Authorization
- User Subscriptions
- Client System Management
- Service Authorization Interface (SA-I)
- Vehicle Interface (V-I)
- Vehicle to Vehicle Interface (VV-I)
- User Authorization Interface (UA-I)
- Service Consumption Interface (SC-I)
- System Management Interface (SM-I)
- Data Logic

© copyright 2004 Telcordia Technologies Inc. All rights reserved.
Navigation: Vehicle Step

- Service Centre
- Payment Centre
- Billing Centre
- Authentication & Authorization
- User Subscriptions
- Client System Management
- System Management Interface (UA-I)
- Service Authorization Interface (SA-I)
- Service Consumption Interface (SC-I)
- IO Device Interface (IOD-I)
- Vehicle Interface (V-I)
- Telematics Control Unit

© copyright 2004 Telcordia Technologies Inc. All rights reserved.
Navigation : Pedestrian Step
To recap…

- End to end architectures allow the functionality of a service to be rendered onto many available devices, depending on what devices happen to be local to the user.
- In general, an E2E architecture requires the collaboration of many corporations to achieve the objectives…with the intended end result that a larger market is created for each than would otherwise be possible.
- OSGi allows execution, and not just data, to migrate too – thus saving costs and enhancing reliability when communication links are involved.
- GST is an example of an End to End Architecture which will support applications running on diverse and dynamic platforms.
...finally

The GST project is a work in progress - our understanding of the world will change and develop over the next two and a half years – come and give your input at the next GST Forum meeting in April in Munich.

http://www.gstforum.org/

.. any questions??