Agenda

- Embedded systems and future networks
- 2 example scenarios and their problems
- How do we solve these
- Demo
- Where we are

- 5 minutes for questions
Embedded Systems and Future Networks

- Embedded systems will continue to spread in end user’s environments
- The tasks of embedded systems will increasingly require communication among each other
- Public Wi-Fi hot-spots, UMTS, etc. will clear the way for new kinds of applications
- But these environments will make the future tasks of embedded systems unpredictable

- Field of study: Automotive Industry
  - Two example scenarios
Example 1/2: Car Tracking Service

- Install a service in the vehicle (PUSH)
- Expose GPS data
  - Web Service, SOAP
  - JMS
- Consume data
  - Language independent
  - Standardized format
- No Hub-and-Spoke architecture necessary!
Example 2/2: Advertising Service

- Present advertisement to the driver
  - E.g. at traffic light

- Install a client to access the advertisement and to display it

- Each area can use a different technology
  - No standardisation necessary
Problem 1/2: Lifecycle Management

- How do we install a service from the outside?
- Which data can a service access?
- We don’t know the offered services in advance
- Once we know, we can’t assume their availability in the future
Problem 2/2: Communication

- We need independent transport technologies …
- To enable unobstructed communication …
- Between highly independent parties

- Service Discovery

- Maybe several vehicles are involved in responding to a request
How to Solve These Problems?

• You know how to solve the lifecycle problem … OSGi
• Communication problem … ?
  • Can we use WS-*, JMS, … ?

• Restricted deployment targets
  • Processor: ARM, PowerPC, x86
  • JavaME CDC, JavaSE 1.4, JavaSE 5 Embedded
  • Lesser memory
Using an ESB in Embedded Systems

• Apache CXF
  • Fulfils the requirements
  • Open Source, active community
  • A lot of in-house experience

• But
  • High memory demands
  • Completely based on JavaSE 5
Adjustment Apache CXF 1/2

- Using OSGi for the bus extension mechanism
  - Register the “CXF bus” as a service
  - New bundles can add their extensions to the bus using this service

- User of CXF can bring their own extensions
  - Keep the core small
Adjustment Apache CXF 2/2

CXF
Java 5

Bytecode Retroweaving

CXF’
Java 4

Change Namespace

Deployable CXF Bundle

Apache Harmony

Bytecode Retroweaving

Apache Harmony’
Summary

- Work in progress
- We are already able to deploy on Java 1.4
- WORA?
  - More like WOTA: Sun’s RI, phoneME, J9, ...
- First, simple attempt to reduce the memory
  - Reduced by 64%
  - ~ 5 MB needed

- Almost done:
  Use your JAX-WS annotated classes in a CDC VM
Thank You!

Questions?