Evolution of an OSGi Application
Introductions

• Arum DataEye
  - www.arumdataeye.com

• Arum Systems Ltd
  - www.arum.co.uk

• Chris Brind (me)
  - www.brindysoft.com
Arum DataEye

- Dashboard for Credit Management Industry
- Flash (Flex) front end
- Java (OSGi) back end
Extension Points
Dashboard Components

Administration Console
Architecture 1/5

- Heavy weight
- OSGi-unfriendly dependencies
- Activators and Service Trackers
- Large footprint (120mb)
- Custom Ant tasks to help with CI
- Slow dev builds
- Slower CI builds
- PDE Bundles
Quick focus on db4o

- No RDBMS
- Extremely simple OODB persistence
- Saved 30% of initial dev time
- Temporary, but worked so well, we kept it.
- www.db4o.com
Architecture 2/5

- Activators and Service Trackers becoming a real problem
- Bundle start ordering problems
- Custom DI
- Composition over Inheritance

Diagram:
- JBoss
- WAR
- Equinox
- Custom DI and Composition over Inheritance
Architecture 3/5

- Custom DI too rigid, too difficult to maintain
- Moved to DS
- Optimised Dev Builds
• Removed external dependency
• Blaze DS -> AMF3OSGi
• Why is this still deployed as a WAR?
Architecture 5/5

- Much more lightweight (15mb inc. container)
- Felix -> Equinox
- With UI overhaul, became ADE 2.0
Scalability

- Service Oriented Front-end Architecture
  - i.e. a browser based RIA
- Server is basically a database for the front end
- No problems with 1GB JVM + 100s of users
The Future

- Improve scalability for 1000s, 10000s or more users?
  - OSGI clustering / cloud?
    - Paremus Service Fabric
  - Large scale JVMs?
    - Zing VM by Azul Systems
- db4o is an embedded database
  - Switch to Versant
  - Replace with RDBMS
Thanks for listening!

- **Arum DataEye**
  - [www.arumdataeye.com](http://www.arumdataeye.com)

- **Arum Systems Ltd**
  - [www.arum.co.uk](http://www.arum.co.uk)

- **Chris Brind (me)**
  - [www.brindysoft.com](http://www.brindysoft.com)