Case Study – An Oracle Coherence Data Grid in OSGi Containers
Background

• Investment bank client
• Post-trade Services
• Several distinct application services layered on a set of common data services
• High Volumes (up to 3m events/day)
• Low Latency
Oracle Coherence

• Clustered in-memory data cache
• API extends java.util.Map
• Redundant storage of data – safe against node or machine outage
• Primary data store in this application – not 2\textsuperscript{nd} level cache
• Internal use of “Portable Object Format” – POF
Hardware Topology

- Each node runs in a container
- All nodes communicate with all others by multicast
- Data nodes have 8Gb heaps
- Application and Proxy nodes are full grid members but with local storage of data disabled
Application Structure
Data Model Hierarchies
Conventional Coherence Classpath Structure

- Stored object and serialisation classes normally on classpath – accessible from Coherence code
- Support for specifying ClassLoader – used in JEE, could be used in OSGi
Object Models as OSGi Services

- All touch points where Coherence needs to be aware of a data object can be intercepted and delegated to OSGi services.
- Cache configurations can be provided to Coherence via an OSGi service.
Data Node Deployment

- On data nodes, coherence must still have access to serialisation and cache configuration
POF Evolvable and code releases

- POF supports an “Evolvable” model for converting objects between old and new versions both ways without data loss.
- Coherence “rolling upgrade” may take considerable time for a large grid.
- Serialisation via OSGi allows upgrade in-place with no node restarts.
The Hard Bits – Coherence

- Coherence cache instances traditionally obtained via static factory methods from a single global configuration.
  - `NamedCache cache = CacheFactory.getCache(cacheName);`
- Static access methods used internally in some places
  - Proxy service
  - Messaging pattern
The Hard Bits – OSGi

- Managing one container per grid node
- Development tooling
- Version dependencies
- Embedding Jars within bundles
- Anomalously long startup time for DM server under Eclipse on Windows
- Incomplete cleaning of state on restart
- Controlled shutdown – order of service removal
- Logging