

OSGi Release 4, version 4.2

Vad är nytt, vad är ändrat ?

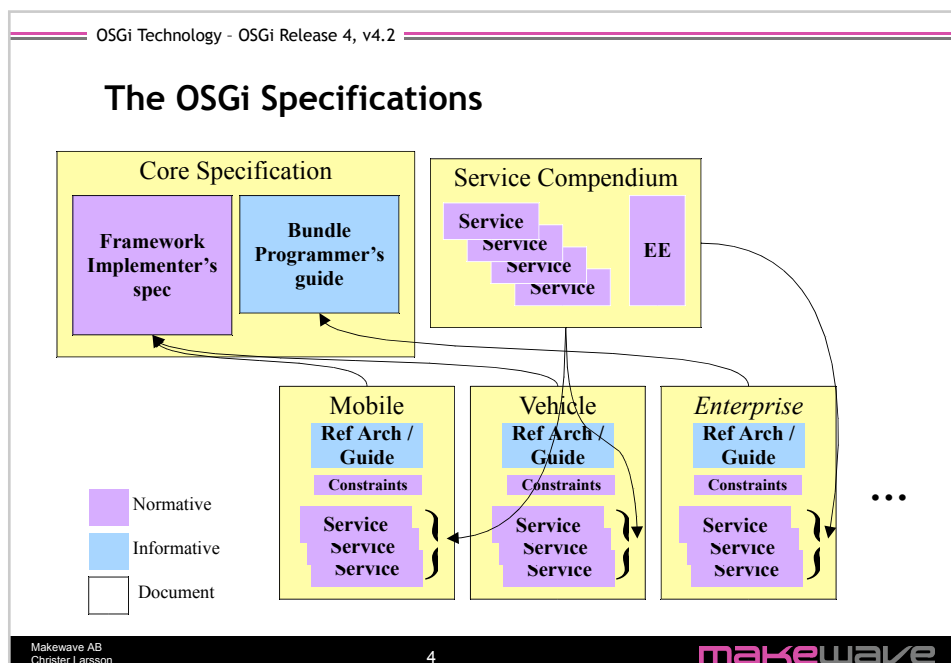
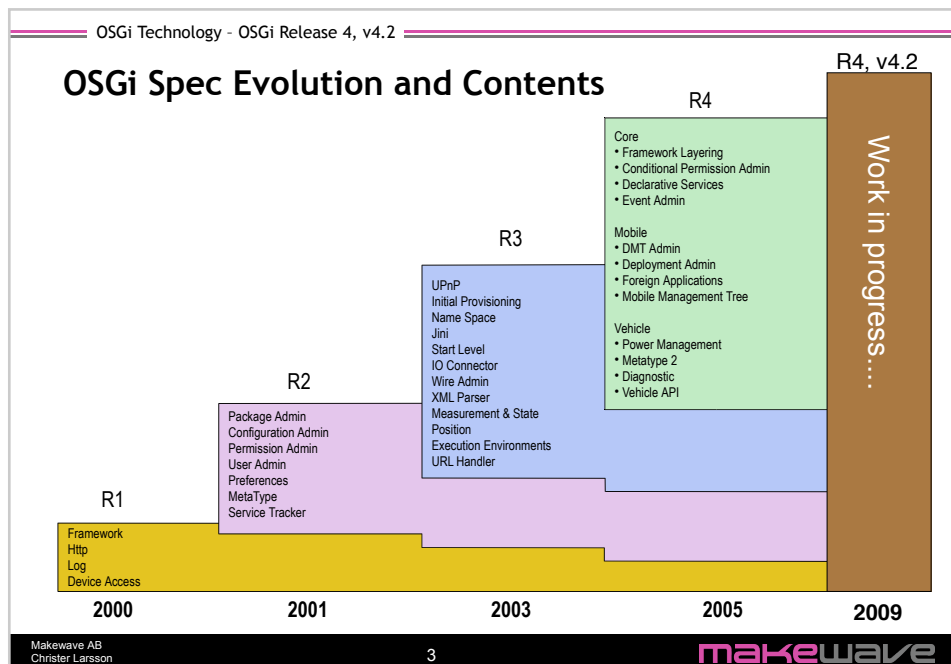
Christer Larsson
VP EMEA OSGi Alliance
CEO Makewave



makewave

OSGi Release 4, version 4.2

- OSGi R4, v4.2 är nästa version av OSGi specifikationen.
- Den innehåller framförallt funktionalitet ämnat för enterprise världen, framtagna av EEE inom OSGi.
- Det finns en preview tillgänglig
- Ytterligare en preview kommer att släppas i samband med OSGidevcon (Eclipsecon)
- En final release är inplanerad till halvårsskiftet 2009.
- Innan vi går in på nyheterna i 4.2 några allmänna ord om OSGi specifikationen , historia, uppbyggnad mm.



The OSGi Expert Groups are creating the specs

- Core Platform Expert Group (CPEG)
 - Responsible for the Framework and core services as well as overall architecture
- Vehicle Expert Group (VEG)
 - Responsible for vehicle related requirements and designs
- Mobile Expert Group (MEG)
 - Responsible for mobile device related requirements and designs
- Enterprise Expert Group (EEG)
 - Responsible for enterprise/server related requirements and designs
- Residential Expert Group (REG)
 - Responsible for residential related requirements and designs

OSGi Release 4, version 4.2

- Följande RFC:er utgör basen i den kommande releasen
- Det finns en preview tillgänglig att ladda ner: OBS det är en preview och räkna med förändringar!
 - 98 - Transactions
 - 119 - Distributed OSGi
 - 120 - Security Enhancements
 - 121 - Bundle Tracker (done)
 - 124 - Blueprint Component Model
 - 125 - Bundle License header (done)
 - 126 - Service Registry Hooks (done)
 - 128 - Accessing ApplicationAdmin Exit Values (done)
 - 129 - Initial Provisioning Update (done)
 - 132 - Command Line and Framework Launching
 - 134 - Declarative Services Update (very near done)
 - 138 - Multiple Frameworks in a VM

Transactions - RFC 98

- An increasing number of service specifications in the OSGi Service Platform rely on some form of transactional behavior.
- Other service specifications could improve if they had transactional behavior.
- This RFC defines a transaction model and identifies Java transaction APIs for use in OSGi environments, including embedded and constrained environments.
- Is build on JTA – Java Transaction Architecture

Distributed OSGi - RFC 119

- Defines a minimal level of feature/function for distributed OSGi processing, including service discovery and access to and from external environments.
- Does not preclude any other solution and is not intended as an alternative to Java EE, SCA, JBI, or any other external API set that may be mapped onto OSGi.

Distributed OSGi - RFC 119

- The RFC aims at enabling the following:
 - An OSGi bundle deployed in a JVM to invoke a service in another JVM, potentially on a remote computer accessed via a network protocol
 - An OSGi bundle deployed in a JVM to invoke a service (or object, procedure, etc.) in another address space, potentially on a remote computer, in a non OSGi environment)
 - An OSGi service deployed in another JVM, potentially on a remote computer, to find and access a service running in the “local” OSGi JVM (i.e. an OSGi deployment can accept service invocations from remote OSGi bundle
 - A program deployed in a non OSGi environment to find and access a service running in the “local” OSGi JVM (i.e. an OSGi deployment can accept service invocations from external environments)

Security Enhancements - RFC 120

- In pre 4.2 you can only grant privileges
- RFC address how to enhance the model by introducing the ability to also deny privileges
- Will give simpler administration in several use cases

Blueprint Service - RFC 124

- Adds a component model for declaring components within a bundle and for instantiating, configuring, assembling and decorating such components when a bundle is started.
- Describes a set of core features required in an enterprise programming model and that are widely used outside of OSGi today when building enterprise (Java) applications.
- LARGELY inspired by Spring Dynamic Modules!
- Improves OSGi framework capabilities in: dependency injection, aspect-like functionality and declarative declarations for enterprise applications.
- In short:
 - Adds *configuration* and *assembly* information to a bundle thereby creating a *managed bundle*. This is done declarative.
 - An *extender bundle* is responsible for observing the life-cycle of these bundle. The extender bundle creates a *module context* for the managed bundle

Multiple OSGi Frameworks in One JVM - RFC 138

- Defines a set of requirements defining the behavior of an OSGi framework
- Special care has to be taken where there are JVM limitations such as JVM singletons, e.g.
 - URLStreamHandlerFactory and ContentHandlerFactory
 - Only one Security Manager reachable via the java.lang.System class. Impacts CPA.
 - System properties
 - System.in,out,err
- Adds service / classes for managing a so called child framework started within a framework.

Minor changes

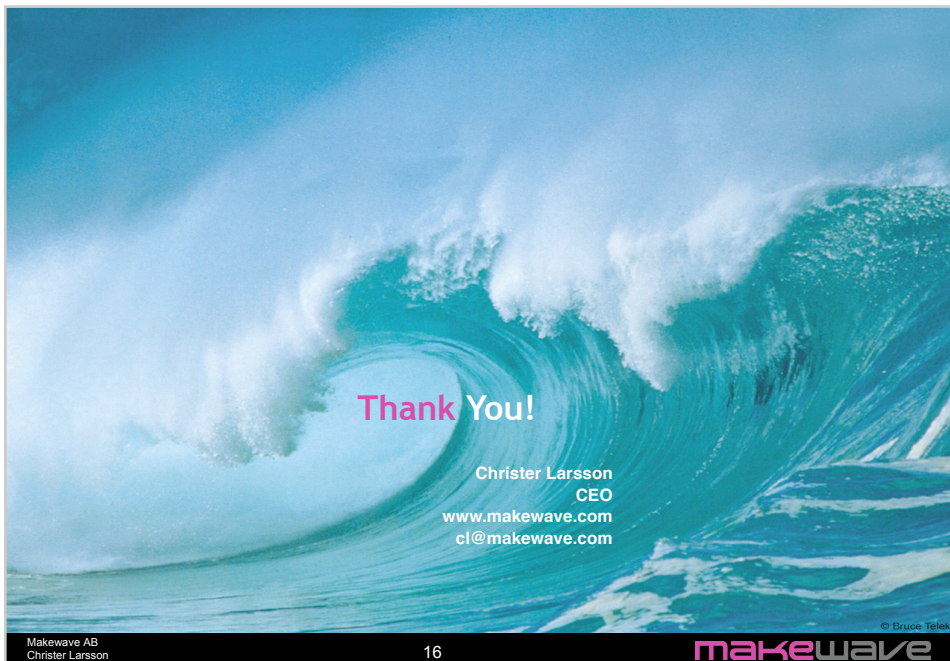
- Bundle Tracker - RFC 121
 - Only `SynchronousBundleListener` supported
- Bundle License – RFC 125
 - New MANIFEST header, Bundle-License
- Service Registry Hooks – RFC 126
 - Adding hooks (or listeners) in the service layer that will be notified on operations in the service registry (publish, find, listen)
- Accessing exit values from applications – RFC 128
 - means for accessing the exit value from an application launched using an Application Admin
- Initial Provisioning update – RFC 129
 - Remove dependency on a special field in zip archive.

Minor Changes, cont.

- Command Line Interface – RFC 132
 - An OSGi specified command line interface service.
- DS updates - RFC 134
 - Minor updates on Declarative Services.

Resources

- The R4 v4.2 preview is available at:
 - <http://www.osgi.org/Specifications/Drafts>
-
- Want to get involved in the spec. work?
 - Please join the Alliance!



Thank You!

Christer Larsson
CEO
www.makewave.com
cl@makewave.com