

Transactions in Eclipse-based SOA

Dimitar Valtchev, Pavlin Dobrev, Roman Roelofsen

Speaker

- Roman Roelofsen
- Software architect at ProSyst Software GmbH

- OSGi Enterprise Expert Group
- OSGi Core Platform Expert Group

Introduction 1/3

- OSGi, the runtime that underpins Eclipse, is used more and more as a deployment and runtime platform for enterprise applications on the server
- Eclipse Swordfish
- IBM, BEA moved to OSGi
- Oracle, JBoss are moving to OSGi
- Apache ServiceMix
- Transaction support by OSGi will become more important

Introduction 2/3

- Use cases
 - ◆ Life cycle operations
 - ◆ Configuration management
 - ◆ Distributed applications
 - ◆ Business process execution

- Different deployment scenarios
 - ◆ Stand alone OSGi runtime
 - ◆ OSGi embedded in JavaEE container
 - ◆ JavaEE server build on OSGi

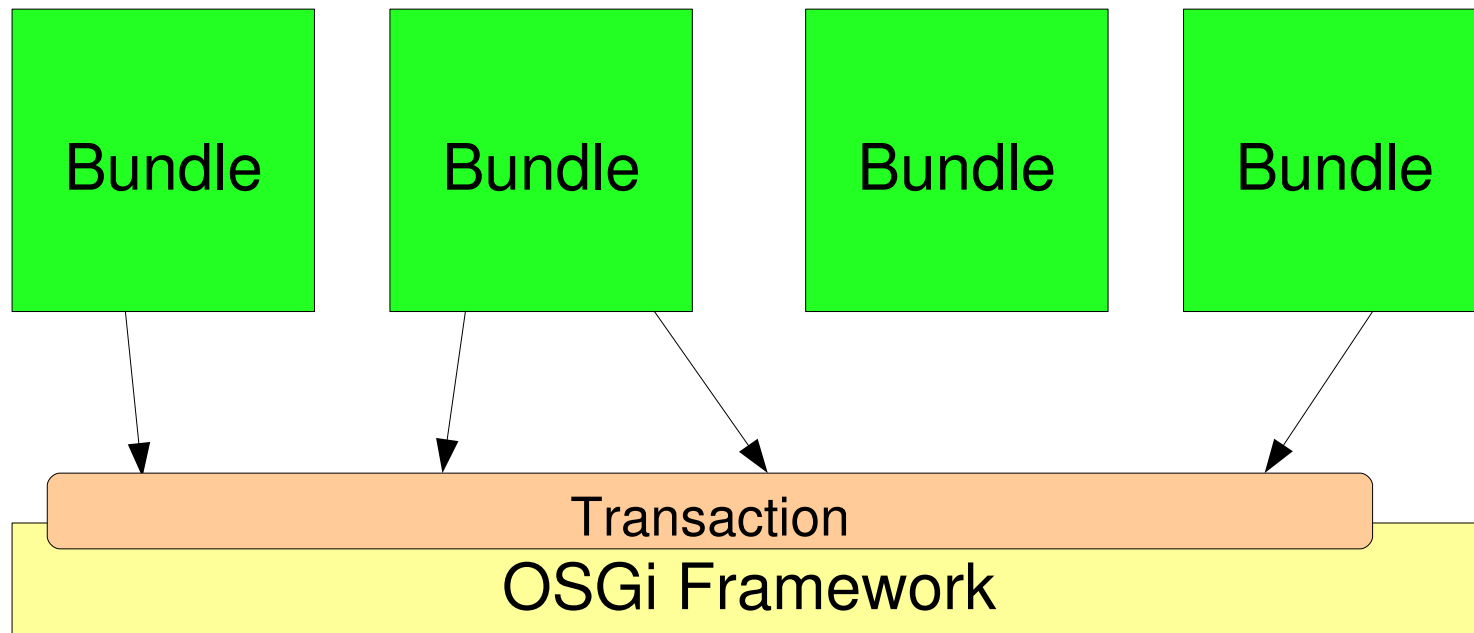
Introduction 3/3

- Basic principles
 - ◆ Applications need a well defined transaction management
 - ◆ The OSGi framework needs to be transaction-aware
 - ◆ Transaction management belongs to the application server, not the application itself
 - ◆ OSGi and the applications do not live in isolation

Transaction in the Context of OSGi

Type 1: Framework Transactional Behaviour

- Execute framework operations as a “unit of work”
 - ◆ Bundle operations, configuration changes, ...

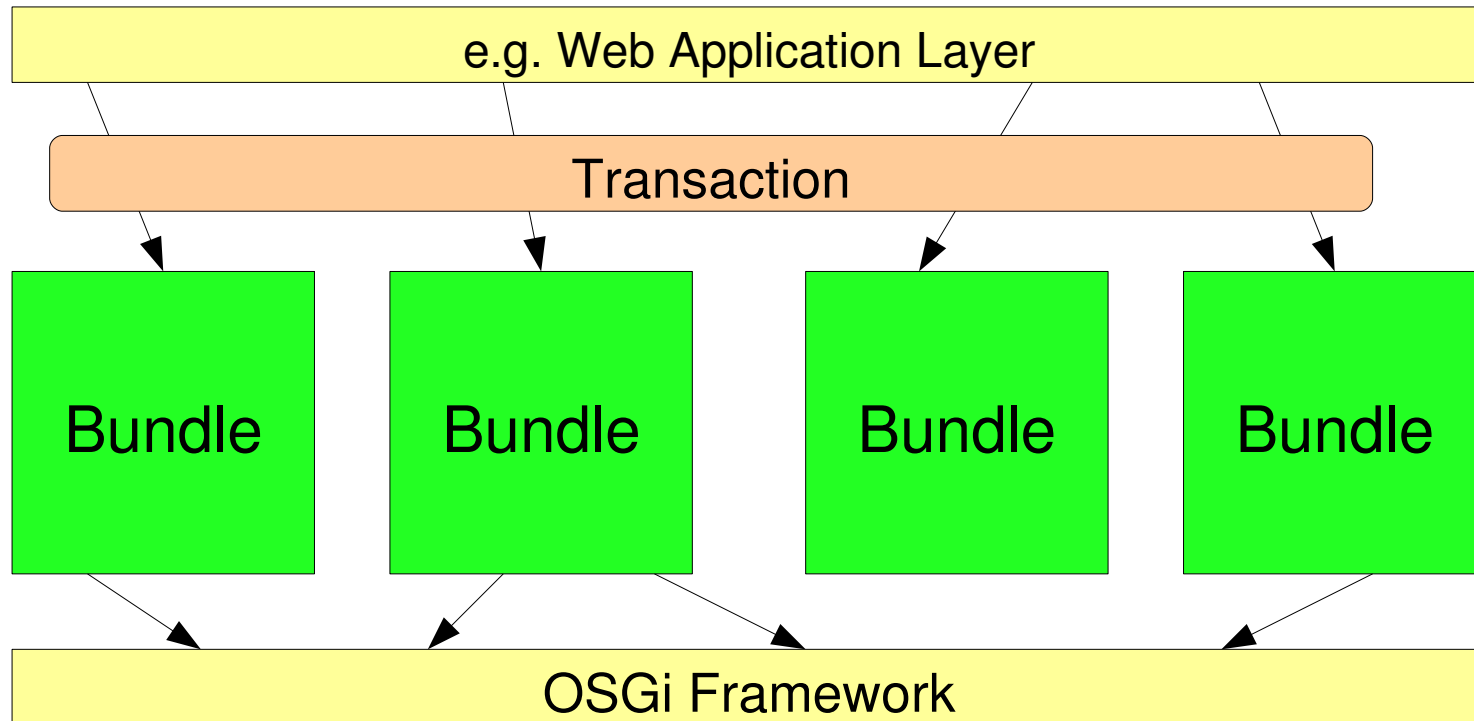


Type 1: Framework Transactional Behaviour

- Downsides:
 - ◆ Only transactional *behaviour*, no real transaction
 - ◆ Hard to group operations from different sources like
 - Declarative Services
 - Configuration Admin
 - Permission Admin
 - ◆ Duplication of code for transaction management
 - ◆ No central place for transactions management
 - Diagnosis
 - Debugging

Type 2: Transaction Support by the Framework

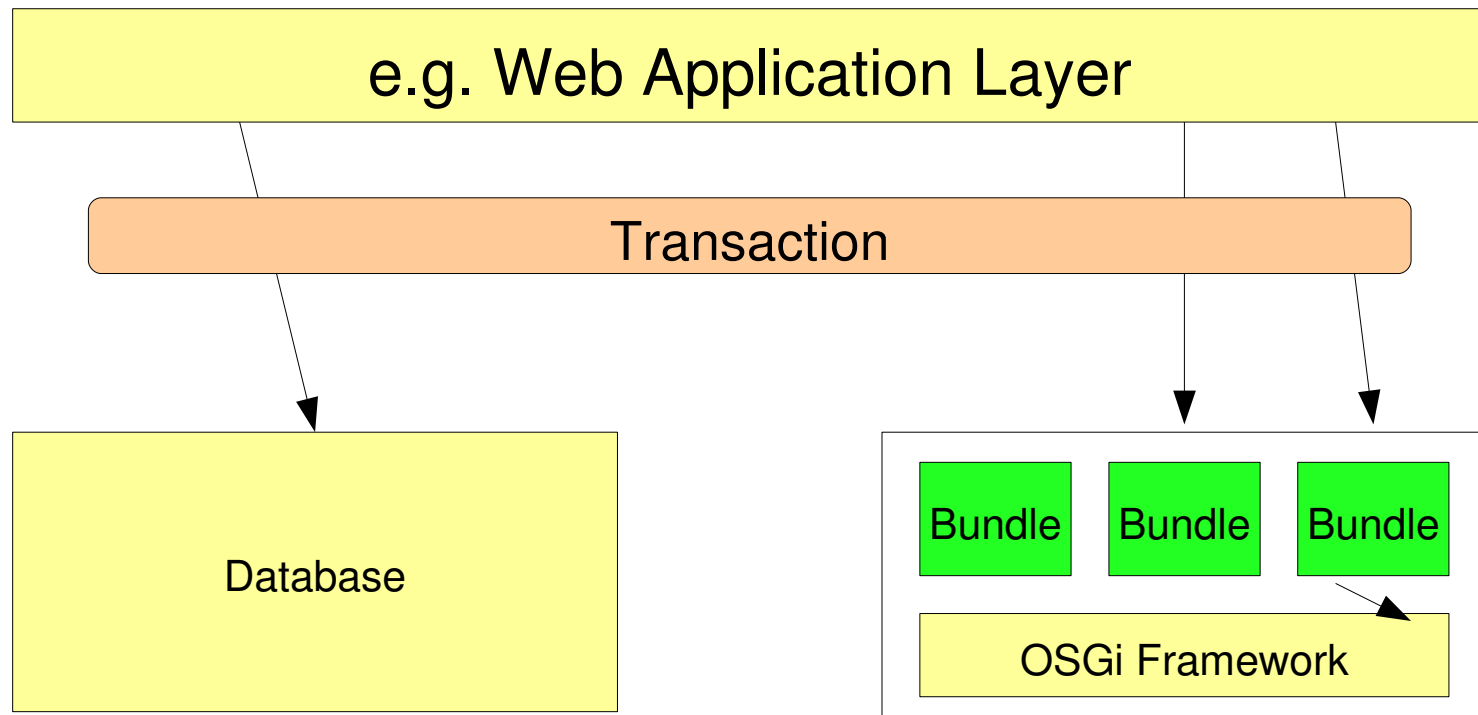
- Full transaction support for *internal* transactions
 - ◆ Transaction API, transaction manager, resource manager, ...



Type 3: Full Transaction Support

- Full transaction support for *internal* and *external* transactions
 - ◆ Distributed transactions
 - ◆ two-phase commit, XA transactions

Type 3: Full Transaction Support



Problems

Problems 1/3

- How should the framework handle *listeners* that are not aware of the transaction?
 - ◆ BundleListener, ServiceListener
- Example:
 - ◆ Transaction start
 - ◆ Install bundle -> framework will send event notifications
 - ◆ Error during bundle install
 - ◆ Transaction roll back -> event ???
- Defer the event? Send uninstall event?

Problems 2/3

- What kind of API to use?
- The OSGi burden:
 - ◆ OSGi is used in the embedded and enterprise space
 - ◆ Enterprise developers will expect e.g. @Transaction
 - ◆ Will not work on the latest JavaME CDC 1.1
 - ◆ Byte code-weaving could solve the problem partially
 - However, do we want this in the specification?

Problems 3/3

- Technical constraints
- E.g. recovery needs a log of all operations
 - ◆ Resource limitations on embedded systems

What do we have now?

Deployment Admin Specification 1/3

- A deployment in OSGi can include
 - ◆ bundles
 - ◆ configuration objects
 - ◆ resources (e.g. images)
 - ◆ ...
- Support needed for tracking the fine-grained dependencies
 - ◆ Which bundles and resources belong together
 - ◆ ...

Deployment Admin Specification 2/3

- Deployment Admin Specification standardizes the access to some of the responsibilities of the management agent
 - ◆ Installing/Un-installing
 - ◆ Customizer
 - ◆ Fix Package
 - ◆ ...
- Chapter 114
 - ◆ OSGi R4 Service Compendium
 - ◆ OSGi R4 Mobile Specification

Deployment Admin Specification 3/3

- All operations take place in a *session*
- Sessions are *committed* or *rolled back*
- Two-phase transactional behaviour (type 1)

DMT Admin Service Specification 1/3

- API for managing a device using concepts from the OMA DM specification
- Management model for mobile devices
 - ◆ Like JMX for JavaEE
 - ◆ But not Java-specific
- Stores configuration data in the device management tree
 - ◆ Informations are stored in nodes
 - ◆ ACLs, Meta nodes

DMT Admin Service Specification 2/3

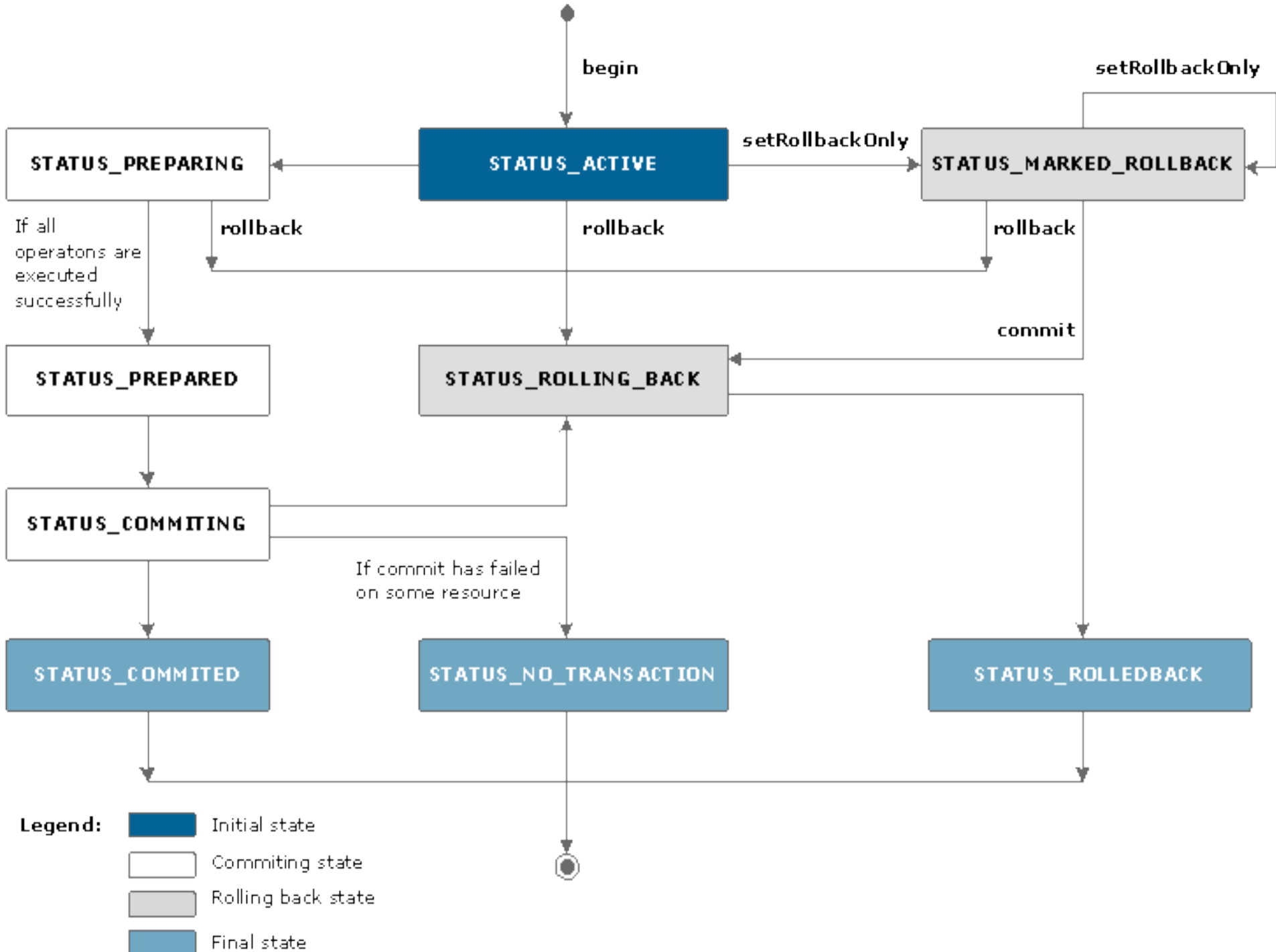
- Access to the DMT is session based
 - ◆ Multiple Readers Session
 - Any number of read only session
 - ◆ Exclusive Update Session
 - Exclusive lock on sub-tree
 - ◆ Atomic Session
 - Exclusive Update Session
 - Can be rolled back

DMT Admin Service Specification 3/3

- Chapter 117
 - ◆ OSGi R4 Service Compendium
 - ◆ OSGi R4 Mobile Specification
- Transactional behaviour (type 1)
- Support is not mandatory in the specification
- Therefore no transactional guarantees in all implementations

ProSyst mBedded Server Professional 1/3

- OSGi Framework implementation by ProSyst Software
- OSGi R4 certified
- Full support for transaction (type 2)
- Transaction support includes
 - ◆ API for transaction management
- API
 - ◆ TransactionManager, Transaction, TransactionContext, TransactionResource



ProSyst mBedded Server Professional 3/3

- System recovery support
 - ◆ Recovery after major crashes
 - ◆ Transaction manager keeps a record of all active transactions
 - ◆ Transaction manager checks at start-up if there are any interrupted transactions
- Deadlock recovery mechanism
 - ◆ Based on transaction timeouts
 - ◆ Transaction can be committed or rollback after predefined timeouts

Custom Solution based on JTA

- `javax.transaction.TransactionManager`
- `javax.transaction.xa.XAResource`

- Requires external transaction manager
 - ◆ e.g. JavaEE server
- Will not work in a stand-alone OSGi runtime
- Full support for transactions (type 3)

Future Developments

RFC 98 – Transactions in OSGi 1/2

- OSGi spec: RFC 98 – Transactions in OSGi
- Handled by the OSGi Core Platform Expert Group
- Requirements
 - ◆ Provide an API for transactions
 - ◆ May provide locking facility
 - ◆ Must allow implementation on embedded systems
 - ◆ No crash recovery

RFC 98 – Transactions in OSGi 2/2

- Requirements (cont'd)
 - ◆ May support recovery from power fails
 - ◆ Must be compatible with JTA as much as possible
 - ◆ ACID
 - ◆ Isolation between participants of a transaction
 - ◆ Set of permissions to restrict the use of transactions

ProSyst mBedded Server Professional (next version)

- Transaction support based on RFC 98
- Transaction resources
 - ◆ OSGi Framework
 - ◆ Configuration Admin Service
 - ◆ Preferences Admin Service
 - ◆ Deployment Admin Service
 - ◆ ...

ProSyst's Future OSGi Enterprise Runtime

- Framework implementation targeted at JavaEE use
- Work in progress, several new features including...
- Full transaction support
 - ◆ Distributed transactions
 - ◆ Framework will participate in external transactions

Summary

- Transaction support is currently fragmented in OSGi
 - ◆ Deployment Admin Spec, DMT Admin Spec
- Custom solutions build on e.g. JTA possible
- Commercial products
 - ◆ ProSyst's mBedded Server Professional
- Ongoing standardisation process
 - ◆ RFC 98
- Future prospects
 - ◆ ProSyst's Future OSGi implementation for enterprise use

Thank you!

Questions?

Roman Roelofsen

r.roelofsen@prosyst.com

<http://www.prosyst.com>

<http://dz.prosyst.com>