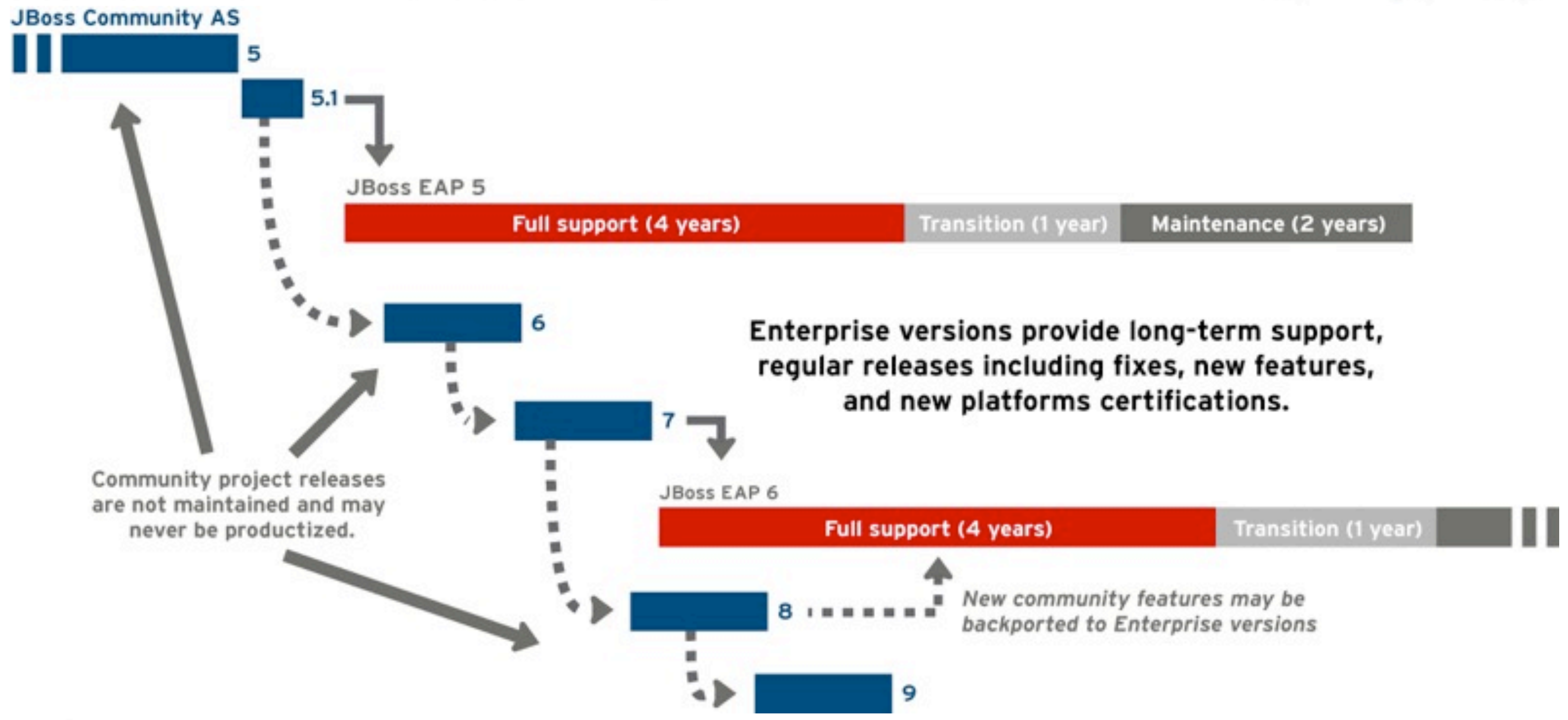
JBoss Application Server 7

JBoss AS vs JBoss EAP



Enterprise versions provide long-term support, regular releases including fixes, new features, and new platforms certifications.

AS7 Key Features

- Fast and Lightweight (More later)
- Supports domain (multi-node) management
- Multiple consistent management interfaces
  - CLI, Java API, HTTP API, Console
- Unified, user focused configuration
  - No more intermixing of internal wiring and config
- Modular
  - Only APIs, no AS implementation exposure
  - True isolation (users can finally use their own XML parsers)

Two Operational Modes

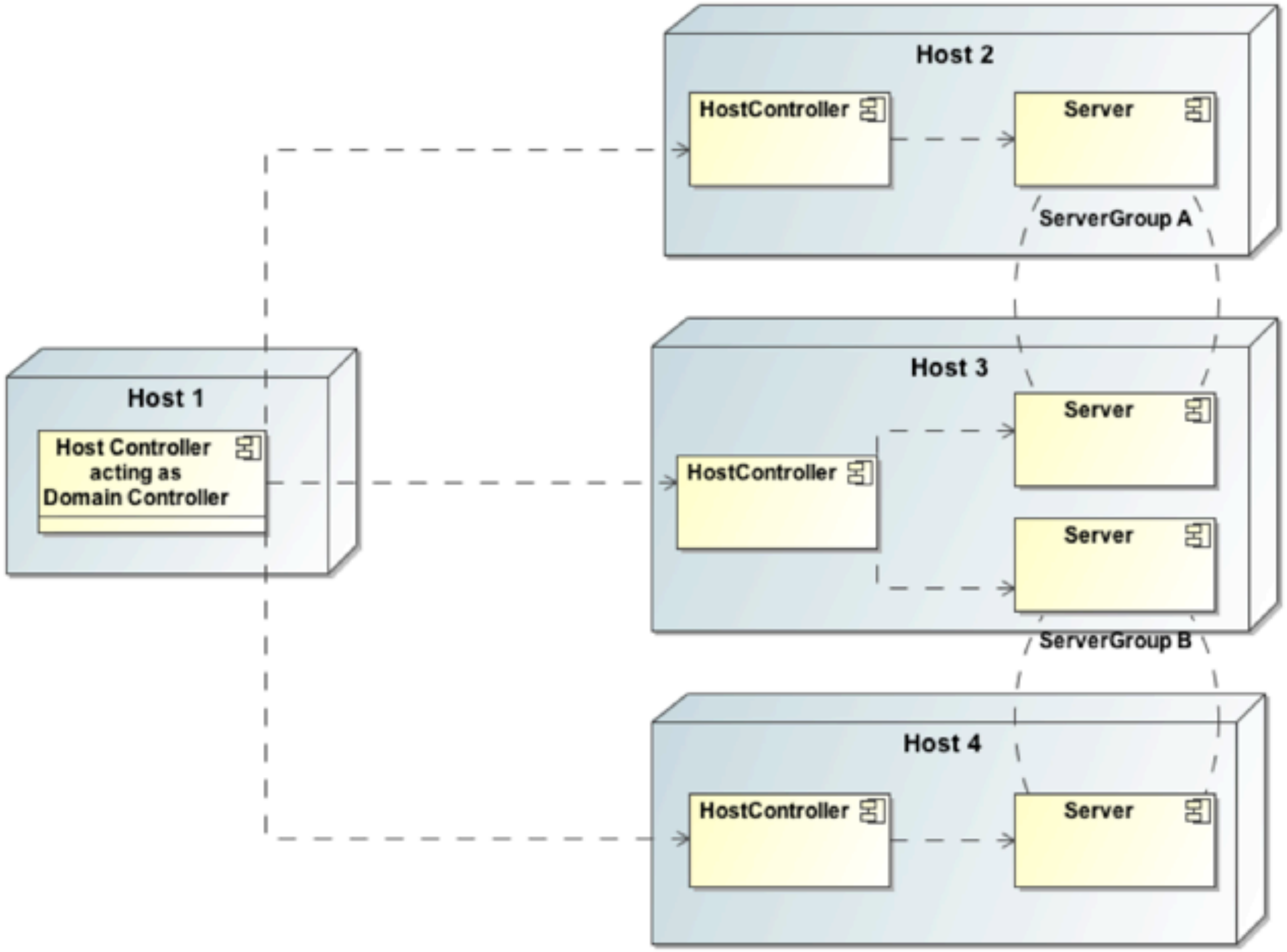
Standalone

- Traditional JBoss single JVM server
- Management facilities IN-VM
- No lifecycle management (only shutdown)

Domain

- Multi-JVM, multi-server model
- Management coordinated by Domain Controller Process
- Multiple server instances (JVMs) per Host
- Full lifecycle managed by Process Controller

Domain Physical Topology



```
jboss-7.0.0.Beta3
    bin
        standalone.conf          Standalone Mode JVM Parameters
        standalone.sh            Standalone Mode
        domain.sh                Domain Mode
        jboss-admin.sh           Command Line Interface
    modules      Static JBoss Module Definitions
    standalone
        configuration
            standalone.xml         Standalone Unified Configuration
        deployments                File System Deployment
        logs
        data                       Internal Data (includes repository)
```

```
jboss-7.0.0.Beta3
    domain
        configuration
            domain.xml          Domain Wide Unified Configuration
            host.xml            Host Controller Configuration
        servers
            server-one          Server “One” JVM instance data
                logs
                data
            server-two          Server “Two” JVM instance data
                logs
                data
```

JBoss Community

User-focused Config

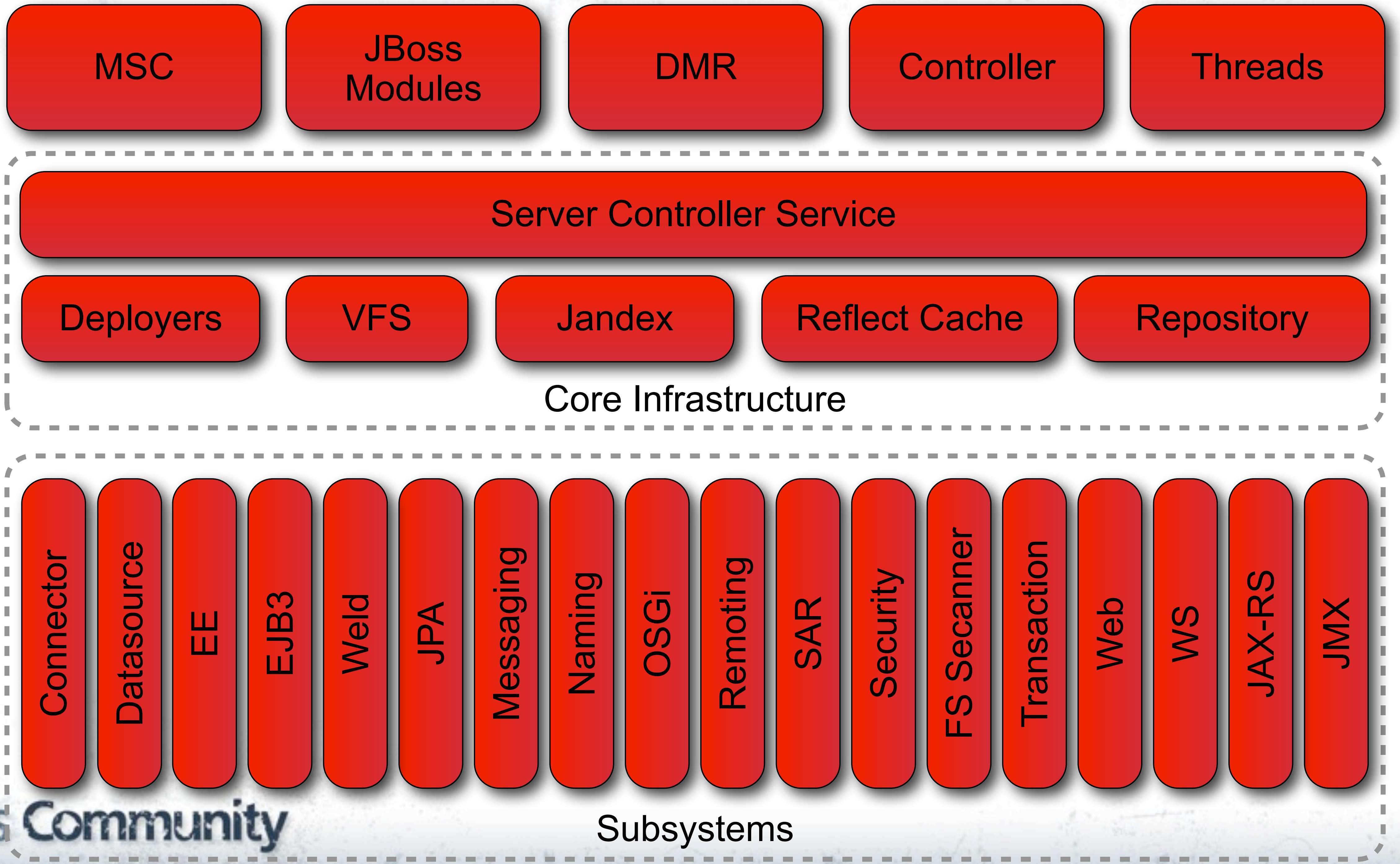
```
<bean name="TransactionManager"
class="com.arjuna.ats.jbossatx.jta.TransactionManagerService">
        <annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX
(name="jboss:service=TransactionManager",
exposedInterface=com.arjuna.ats.jbossatx.jta.TransactionManagerServiceMBean.class,
registerDirectly=true)</annotation>
        <annotation>@org.jboss.managed.api.annotation.ManagementObject
(name="TransactionManager",componentType=@org.jboss.managed.api.annotation.ManagementCompone
nt(type = "MCBean", subtype =
"JTA"),targetInterface=com.arjuna.ats.jbossatx.jta.TransactionManagerServiceMBean.class)
        </annotation>

        <property name="transactionTimeout">300</property>
        <property name="objectStoreDir">${jboss.server.data.dir}/tx-object-store</property>
```

```
<subsystem xmlns="urn:jboss:domain:transactions:1.0">
            <recovery-environment socket-binding="txn-recovery-environment"
                                  status-socket-binding="txn-status-manager"/>
            <core-environment socket-binding="txn-socket-process-id"/>
</subsystem>
```

JBoss Community

AS7 Architecture



Key MSC Attributes

Small, lightweight, and efficient

- 216K JAR
- Tiny memory overhead

Highly concurrent & scalable state machine

- State transitions are “tasks”

Services are primarily interface based

- No reflection or XML required!

Only two non-error, non-transition states

- UP & DOWN
- N-State systems are implemented with N services

Other notable features

Multiple startup modes

- Active, Passive, Lazy, On-Demand, Never

Strong typing

Listeners

- Capable of modeling advanced cases

Child services

Async start & stop

Advanced Cycle Detection and Error Detection

Immediate & Deferred resolution

JBoss Modules

Small, lightweight, and efficient

- O(1) Dep resolution
- Concurrent class loading (lockless on most VMs)
- 207k JAR

“Pure” modular class loading

- Modules only see what they import (includes JDK classes!)

External module definitions

- Don’t have to break open the JAR

Dynamic

- Modules can be redefined

Extensible

- JBoss OSGi implemented on modules

JBoss Modules

Small, lightweight, and efficient

- O(1) Dep resolution
- Concurrent class loading (lockless on most VMs)
- 207k JAR

“Pure” modular class loading

- Modules only see what they import (includes JDK classes!)

External module definitions

- Don’t have to break open the JAR

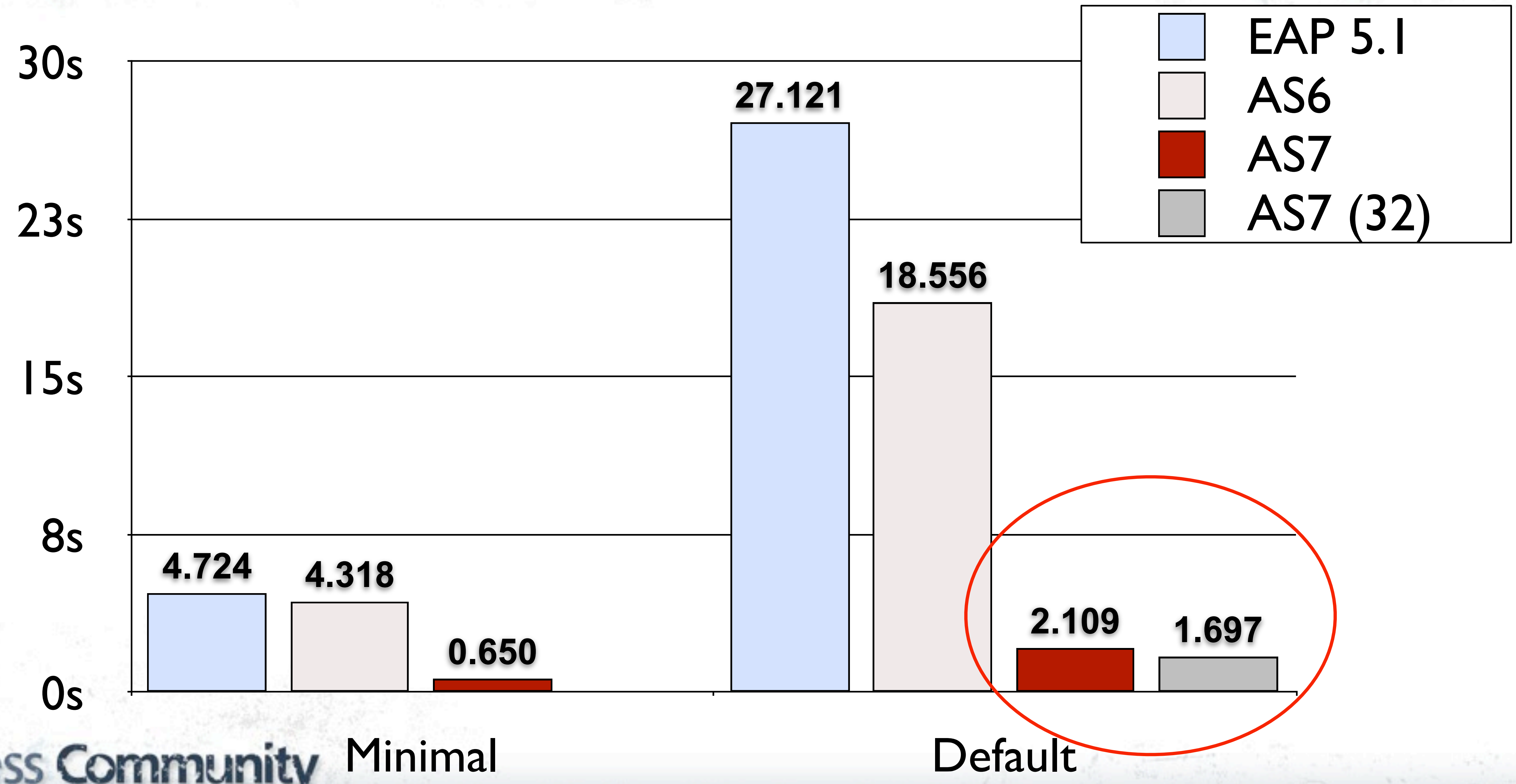
Dynamic

- Modules can be redefined

Extensible

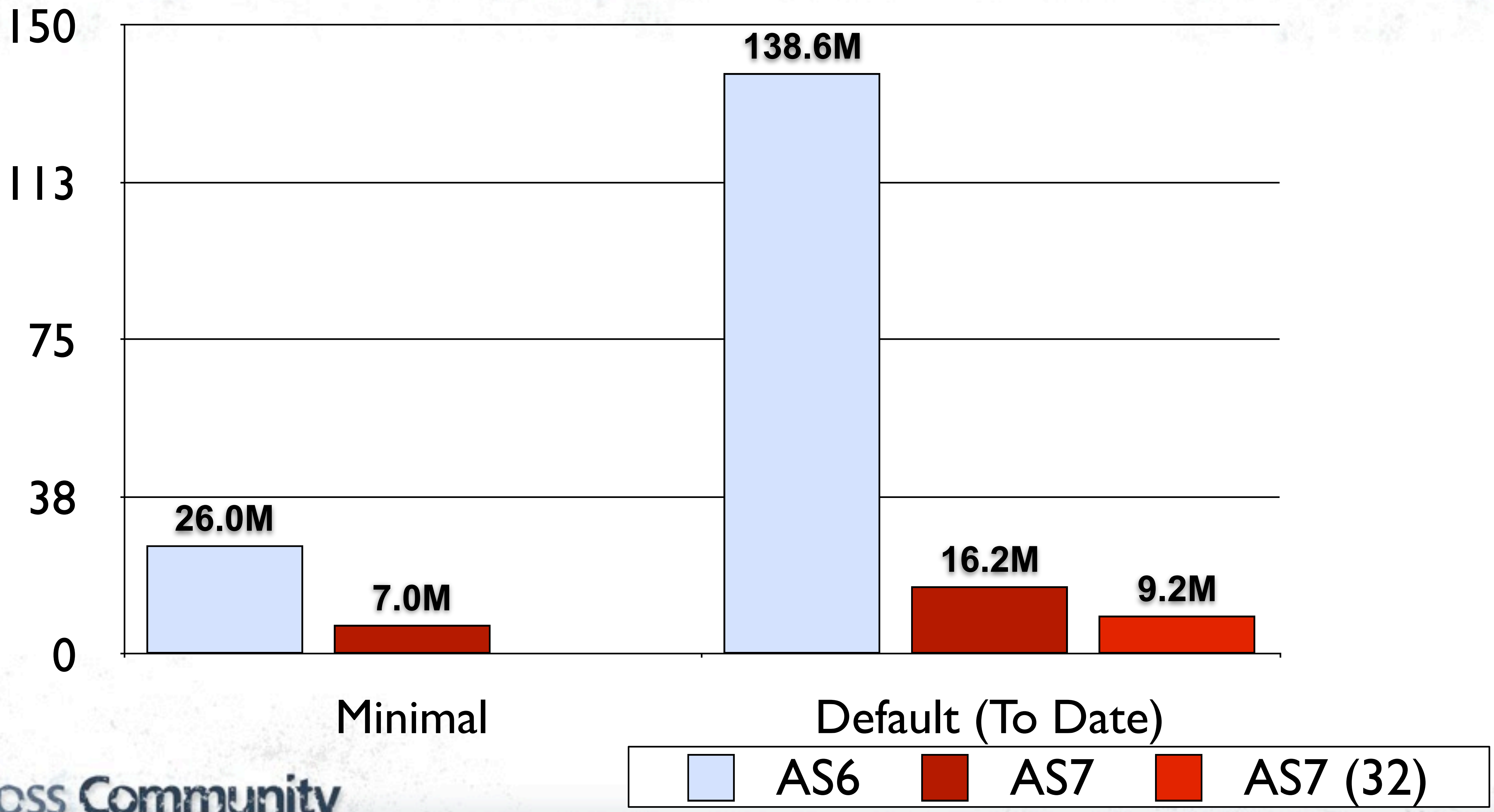
- JBoss OSGi implemented on modules

Boot Time Results



JBoss Community

Memory Consumption



JBoss Community

New Possibilities

- Execute an entire Java EE container in a unit test!
- Use an appserver VM per Application!
- Run in constrained environments (light cloud, plug computers, mobile devices)
- Develop faster than you ever have before!
- Run multiple staging servers on your laptop!

CR1 “White Rabbit” Released!

Get it:

http://jboss.org/jbossas/downloads

Code it:

http://github.com/jbossas

JBoss Community