JBossESB

Dr Mark Little

Director of Standards, Development Manager

Overview

- What are the aims behind JBossESB?
  - ✓ Requirements
- Architecture
  - ✓ Messages and services
  - ✓ Interoperability
    - Deployment realities
- What services will we support?
- Standards
- Current status
  - ✓ Rosetta ESB

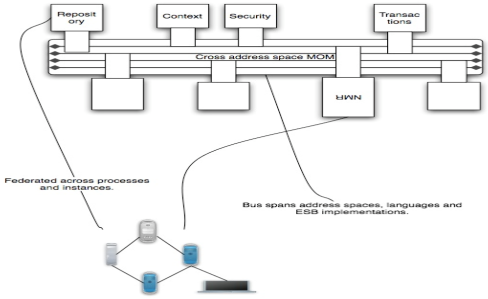
Aims

- To provide the standard OSS infrastructure for SOA
  - ✓ SOA principles first and foremost
- Use SOA principles *internally* as well as *externally*
  - ✓ Everything will (conceptually) be considered as a service
  - ✓ Everything will be replaceable
- Standards compliant
  - ✓ Though requirements live longer

Requirements

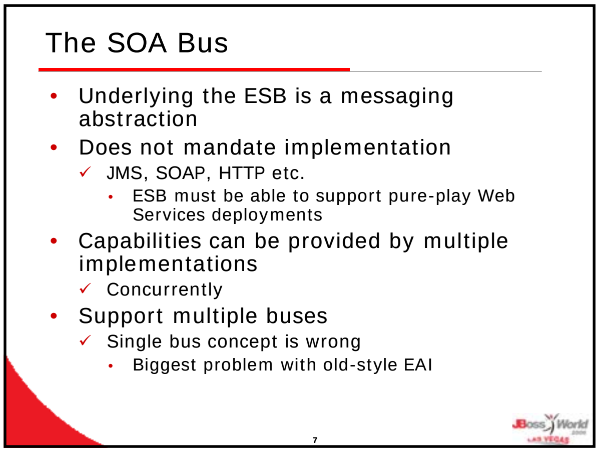
- Cannot mandate specific capability implementations
- *All* capabilities accessed as services
  - ✓ Plug-and-play
  - ✓ Extensibility
- *All* capabilities are message based
  - ✓ Including (conceptually) the container
- Standards are important
  - ✓ JBI
  - ✓ Perhaps SCA

Architecture overview


Reposit ory
Context
Security
Transac tions
Cross address space MOM
NMR
Federated across processes and instances.
Bus spans address spaces, languages and ESB implementations.

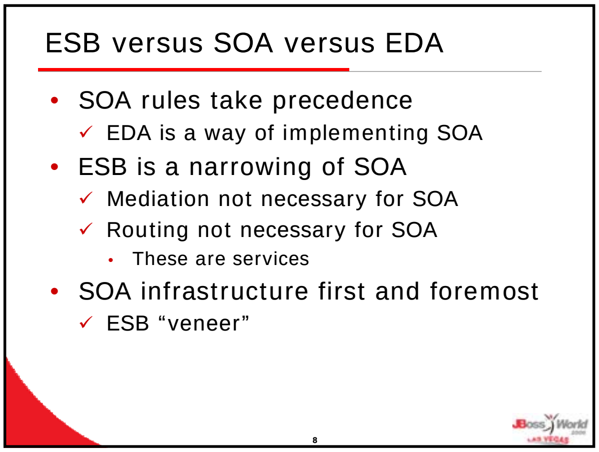
Services and messages

- Everything is a service, including the bus
  - All services are interacted with via messages.
  - Includes service lifecycles
    - ✓ Containers abstracted within architecture
      - ✓ Services plugged directly into a lifecycle bus
  - Services can be plugged into multiple buses concurrently



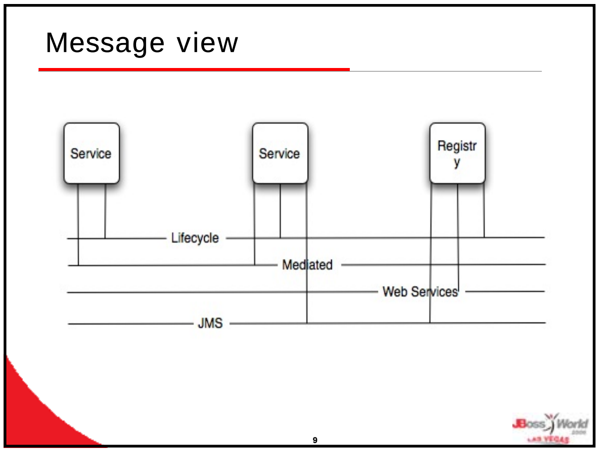
The SOA Bus

- Underlying the ESB is a messaging abstraction
- Does not mandate implementation
  - ✓ JMS, SOAP, HTTP etc.
    - ESB must be able to support pure-play Web Services deployments
- Capabilities can be provided by multiple implementations
  - ✓ Concurrently
- Support multiple buses
  - ✓ Single bus concept is wrong
    - Biggest problem with old-style EAI

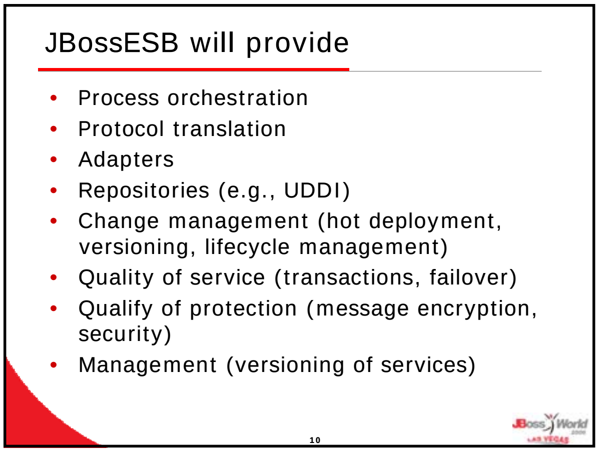


ESB versus SOA versus EDA

- SOA rules take precedence
  - ✓ EDA is a way of implementing SOA
- ESB is a narrowing of SOA
  - ✓ Mediation not necessary for SOA
  - ✓ Routing not necessary for SOA
    - These are services
- SOA infrastructure first and foremost
  - ✓ ESB “veneer”

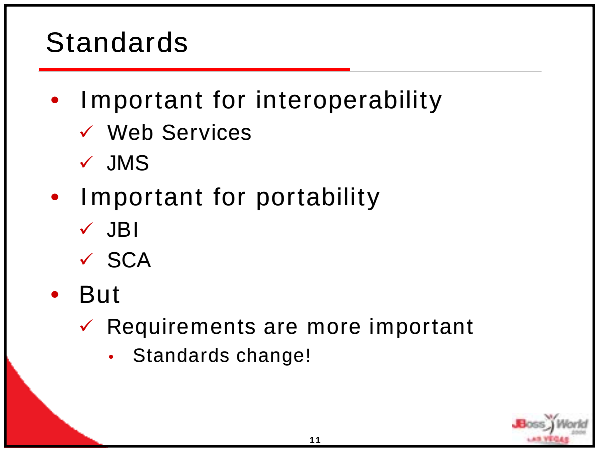


Message view



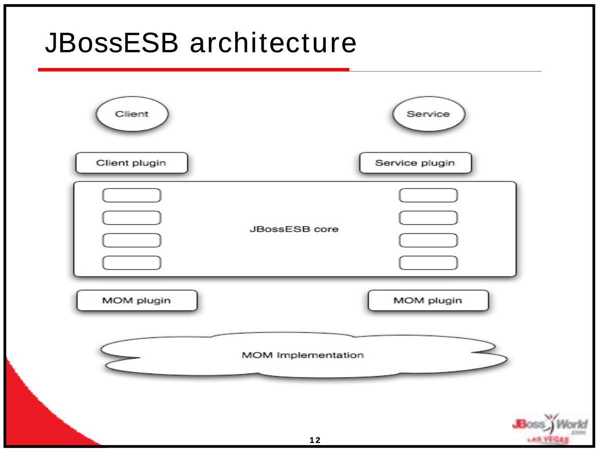
JBossESB will provide

- Process orchestration
- Protocol translation
- Adapters
- Repositories (e.g., UDDI)
- Change management (hot deployment, versioning, lifecycle management)
- Quality of service (transactions, failover)
- Qualify of protection (message encryption, security)
- Management (versioning of services)

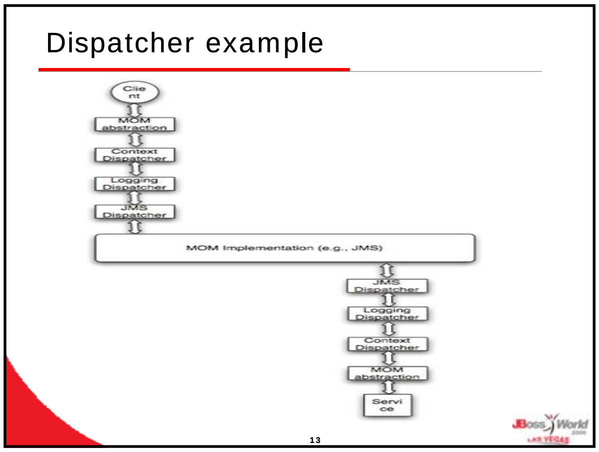


Standards

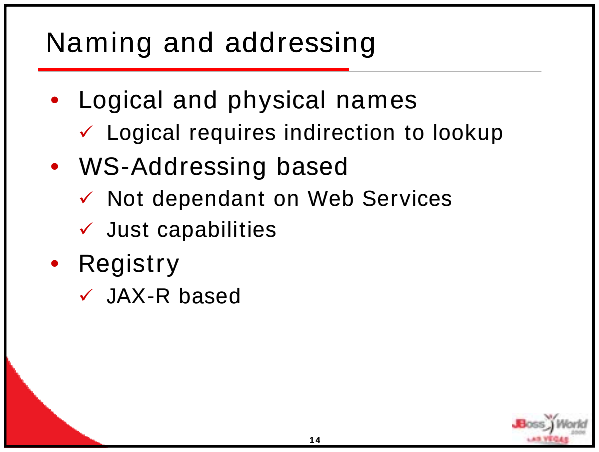
- Important for interoperability
  - ✓ Web Services
  - ✓ JMS
- Important for portability
  - ✓ JBI
  - ✓ SCA
- But
  - ✓ Requirements are more important
    - Standards change!



JBossESB architecture

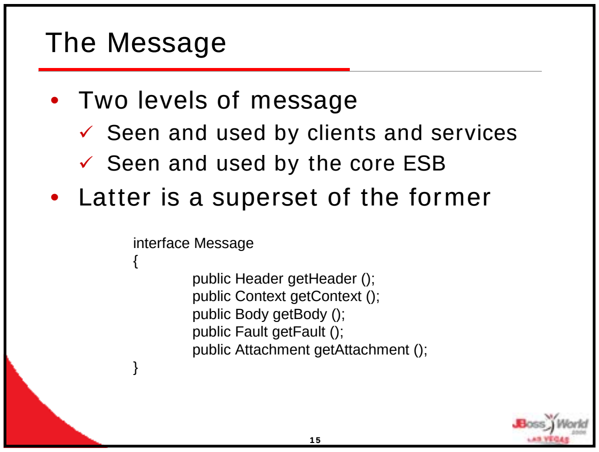


Dispatcher example



Naming and addressing

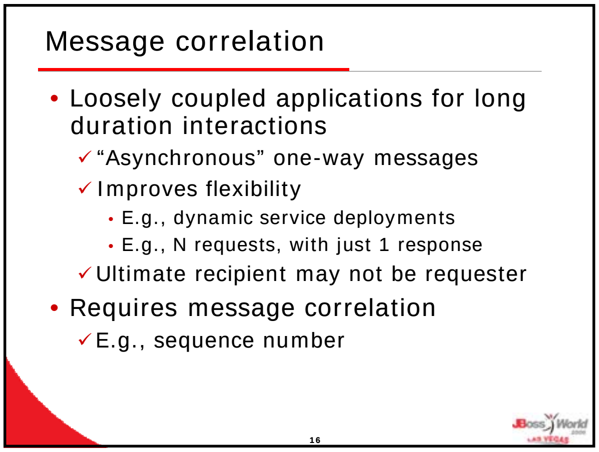
- Logical and physical names
  - ✓ Logical requires indirection to lookup
- WS-Addressing based
  - ✓ Not dependant on Web Services
  - ✓ Just capabilities
- Registry
  - ✓ JAX-R based



The Message

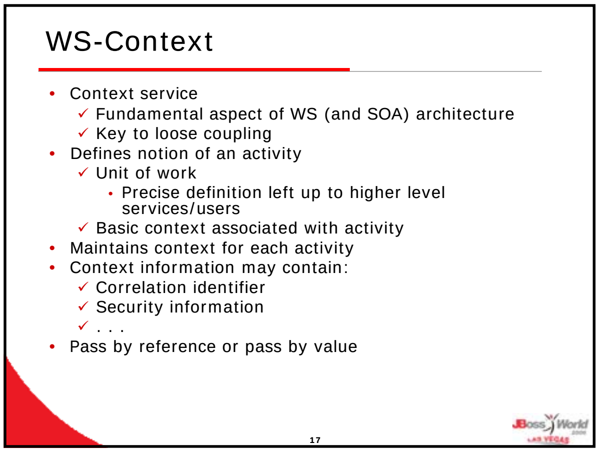
- Two levels of message
  - ✓ Seen and used by clients and services
  - ✓ Seen and used by the core ESB
- Latter is a superset of the former

```
interface Message
{
        public Header getHeader ();
        public Context getContext ();
        public Body getBody ();
        public Fault getFault ();
        public Attachment getAttachment ();
}
```



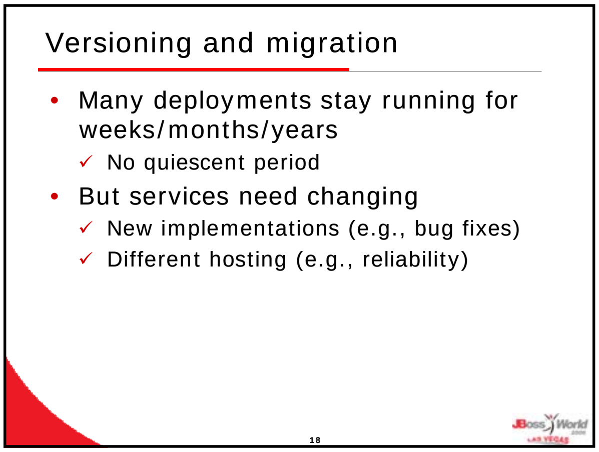
Message correlation

- Loosely coupled applications for long duration interactions
  - ✓ “Asynchronous” one-way messages
  - ✓ Improves flexibility
    - E.g., dynamic service deployments
    - E.g., N requests, with just 1 response
  - ✓ Ultimate recipient may not be requester
- Requires message correlation
  - ✓ E.g., sequence number



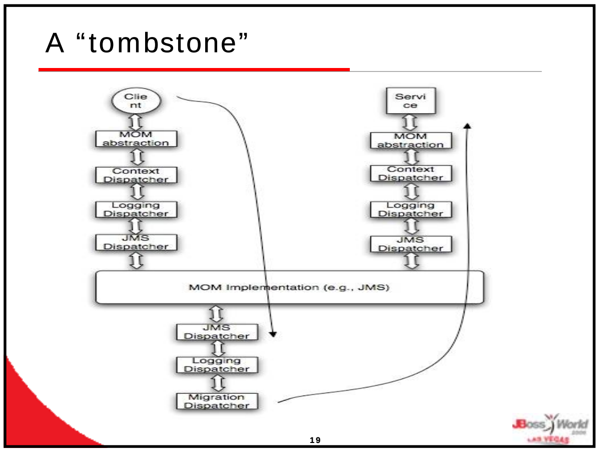
WS-Context

- Context service
  - ✓ Fundamental aspect of WS (and SOA) architecture
  - ✓ Key to loose coupling
- Defines notion of an activity
  - ✓ Unit of work
    - Precise definition left up to higher level services/users
  - ✓ Basic context associated with activity
- Maintains context for each activity
- Context information may contain:
  - ✓ Correlation identifier
  - ✓ Security information
  - ✓ . . .
- Pass by reference or pass by value



Versioning and migration

- Many deployments stay running for weeks/months/years
  - ✓ No quiescent period
- But services need changing
  - ✓ New implementations (e.g., bug fixes)
  - ✓ Different hosting (e.g., reliability)



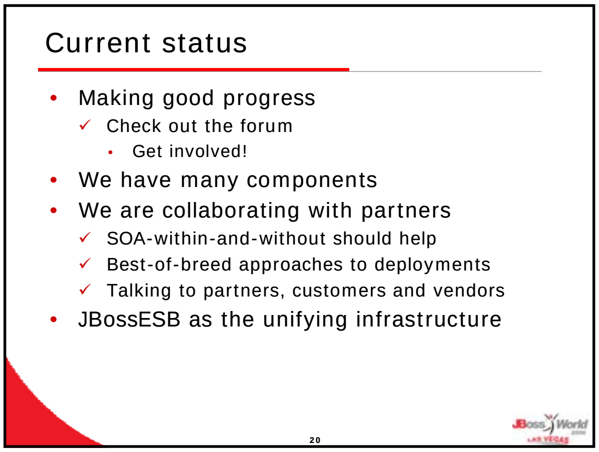
A “tombstone”

Client
MOM abstraction
Context Dispatcher
Logging Dispatcher
JMS Dispatcher

Service
MOM abstraction
Context Dispatcher
Logging Dispatcher
JMS Dispatcher

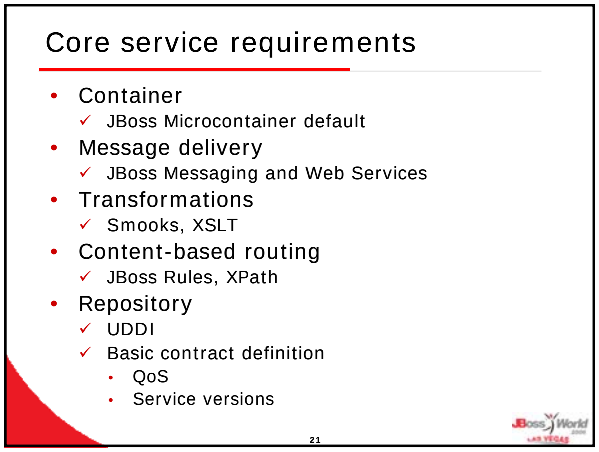
MOM Implementation (e.g., JMS)

JMS Dispatcher
Logging Dispatcher
Migration Dispatcher



Current status

- Making good progress
  - ✓ Check out the forum
    - Get involved!
- We have many components
- We are collaborating with partners
  - ✓ SOA-within-and-without should help
  - ✓ Best-of-breed approaches to deployments
  - ✓ Talking to partners, customers and vendors
- JBossESB as the unifying infrastructure

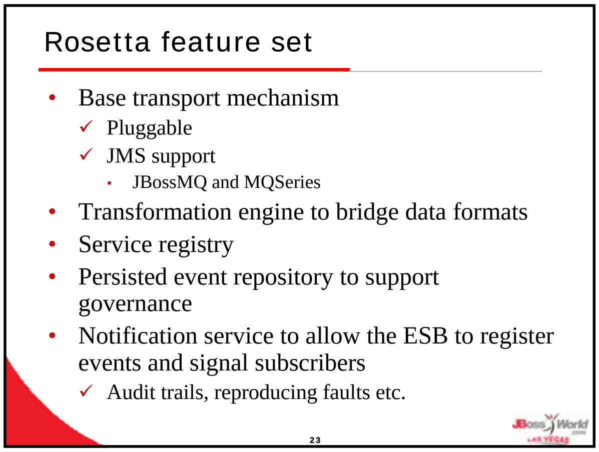


Core service requirements

- Container
  - ✓ JBoss Microcontainer default
- Message delivery
  - ✓ JBoss Messaging and Web Services
- Transformations
  - ✓ Smooks, XSLT
- Content-based routing
  - ✓ JBoss Rules, XPath
- Repository
  - ✓ UDDI
  - ✓ Basic contract definition
    - QoS
    - Service versions

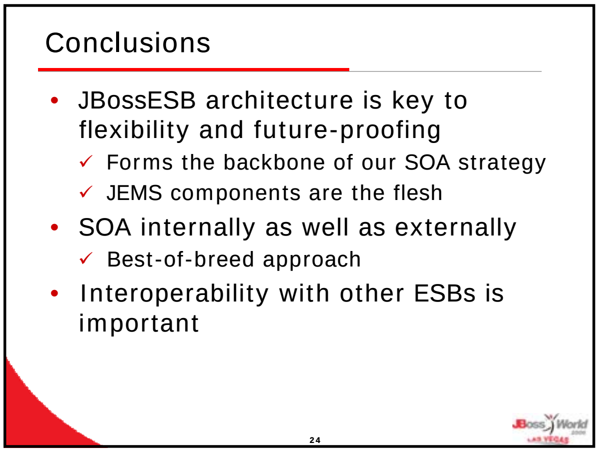
Recent news

- JBoss received donation of enterprise-ready ESB
  - ✓ Rosetta
- Used to run 2nd largest Canadian insurance firm
  - ✓ In use continuously for 3 years
    - Build to handle > 10K real-time events/day
      - ✓ Legacy and JEE applications
    - Very few other ESB implementations can say the same thing



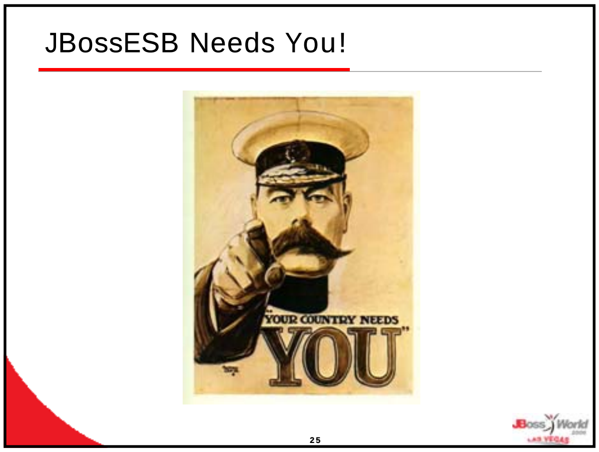
Rosetta feature set

- Base transport mechanism
  - ✓ Pluggable
  - ✓ JMS support
    - JBossMQ and MQSeries
- Transformation engine to bridge data formats
- Service registry
- Persisted event repository to support governance
- Notification service to allow the ESB to register events and signal subscribers
  - ✓ Audit trails, reproducing faults etc.



Conclusions

- JBossESB architecture is key to flexibility and future-proofing
  - ✓ Forms the backbone of our SOA strategy
  - ✓ JEMS components are the flesh
- SOA internally as well as externally
  - ✓ Best-of-breed approach
- Interoperability with other ESBs is important



JBossESB Needs You!