Messaging Security using GlassFish and Open Message Queue

OWASP AppSec USA 2011 Conference (@appsecusa / hashtag: #appsecusa)
Srini Penchikala (@srinip)
09.23.11

Goals and Scope

- **Goals:**
  - Messaging security architecture & design considerations and best practices
  - How to use NoSQL Databases for some messaging security use cases
- **Is Not:**
  - Security vulnerabilities talk
- **Is:**
  - Focus on application security in messaging:
    - Authentication & authorization
    - Message encryption
    - Logging and monitoring
  - Code examples on messaging security aspects (Java based)
- **Target Audience:**
  - Architects, Application Developers, and Security Ops
- **Format:**
  - 45 min presentation + 5 min Q&A
  - Demo Application (Java)

About the Speaker

- Security Architect
- Certified Scrum Master
- Author, Editor (InfoQ)
- IASA Austin Chapter Leader
- Detroit Java User Group Leader (past)
- Working with Java since 1996, JEE (2000), SOA (2006), Security (2007) & PPT since 03/2011
- *Current:* Agile Security Architectures, NoSQL Security, Domain-Driven Security Design, Security Architecture Enforcement, Model Driven Development
- *Future:* Role of DSL in Architecture Enforcement, NoSQL Security Tools and Frameworks

Before We Start

- How many are currently using messaging in their applications?
- How many are currently working in security architecture or development?
- Any regulatory compliance requirements (Federal, State, Local, or Finance related)?

Background

- Financial services
- J2EE security model
- Agile software development
- Regulatory compliance and its impact on IT
- Software architecture

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Messaging Architecture

- Asynchronous Communication
- Message-Oriented Middleware (MOM) pattern
- Java Message Service (JMS)
- Architecture Components:
  - Broker
  - Connection
  - Destination (Queue or Topic)
  - Transactions
  - Message
- Enterprise Integration Patterns*

***Source:** http://www.eaipatterns.com/toc.html

Messaging Security Considerations

- Current state: Not enough focus on the middleware
- Authentication
- Role Based Access Control
- Message Encryption
- Transport Layer Security
- Message Persistence
- Secure Logging and Auditing
- Secure Message Monitoring
  - Standards based administration (JMX)
- Availability
  - Broker clustering
  - Automatic reconnect
  - Connection event notification

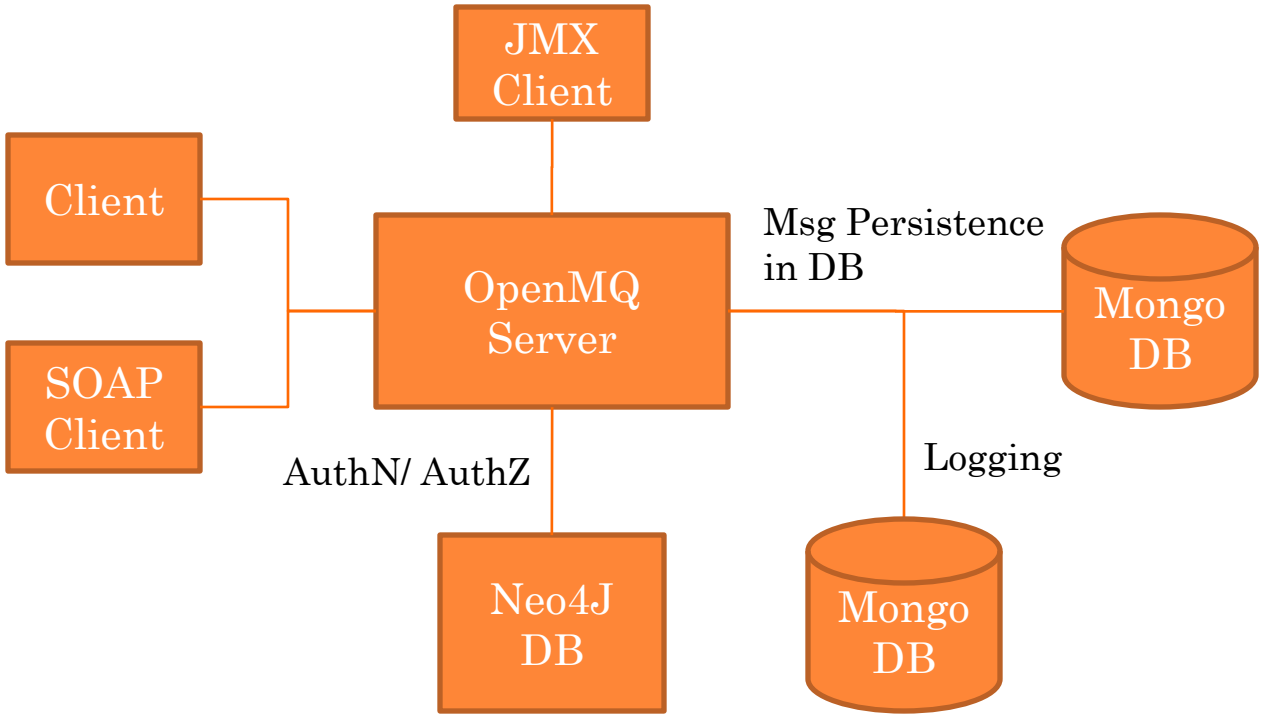
Technologies

- GlassFish (v3.1)
  - Java EE 6 compliant application server
  - Open source and commercial versions
- Open Message Queue (v4.5)
  - Implements JMS standard
  - Stand-alone service or deployed in an application server
  - Standard JMS Provider in GlassFish Server
  - Supports messaging security at various levels
  - Broker clusters (conventional and enhanced clusters)
- Spring Integration Framework
  - Secure message channel (role based access)

Sample Demo Application

- Technologies:
  - GlassFish 3.1 Application Server
  - Open MessageQueue
  - Neo4J (authentication and authorization)
  - MongoDB (persistence and logging)
  - Spring Data Framework
  - Spring Integration
  - Aspect-oriented Programming
- Tools:
  - JDK 1.6
  - Eclipse

Sample Application Architecture



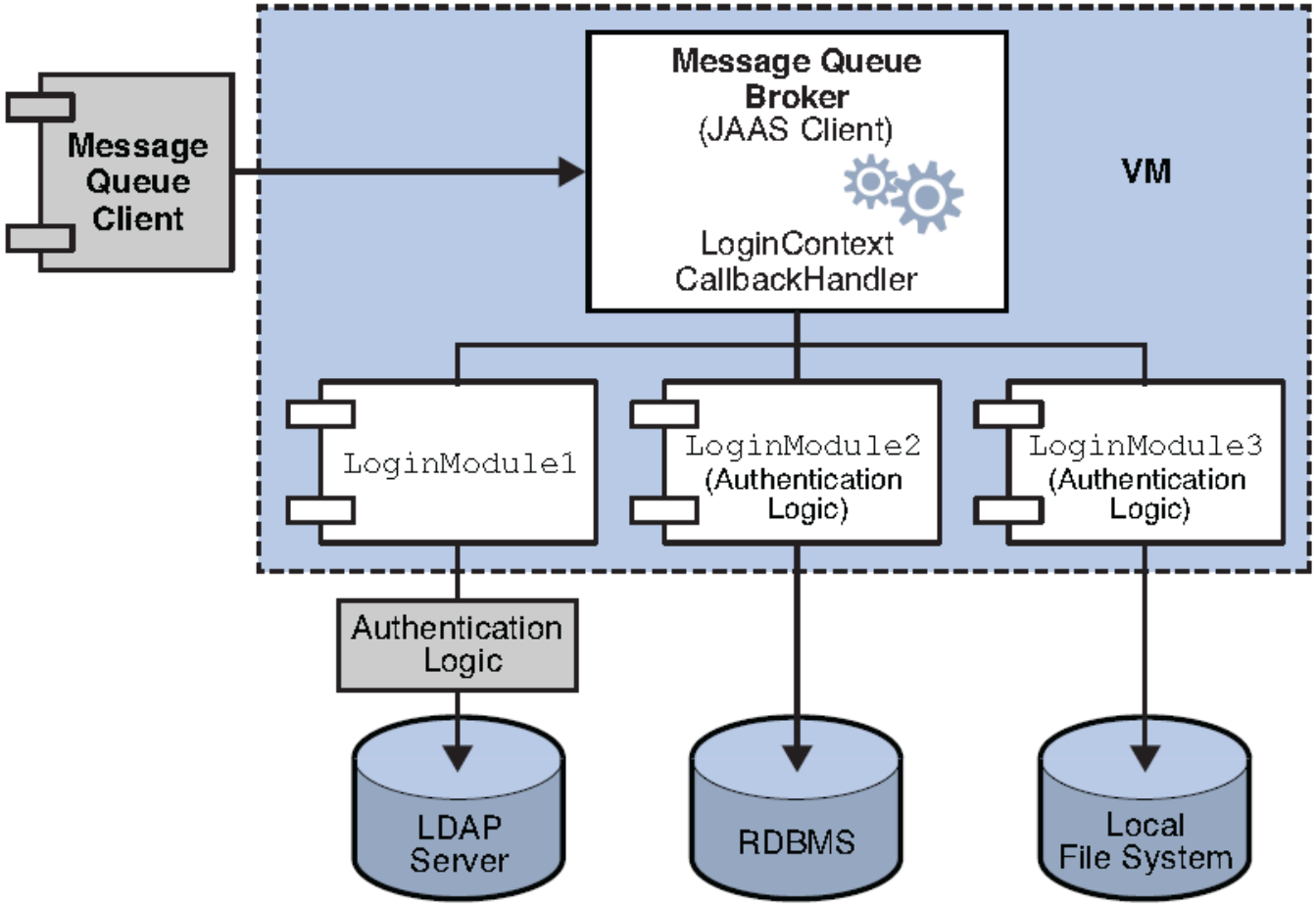
Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Authentication

- Authentication
  - Flat file repository
  - LDAP authentication
  - JAAS-based authentication
- Per broker repository
- NoSQL database for storing user profiles
  - Graph databases (Neo4J)

JAAS and Message Queue



Source: OpenMessage Queue 4.5 Administration Guide

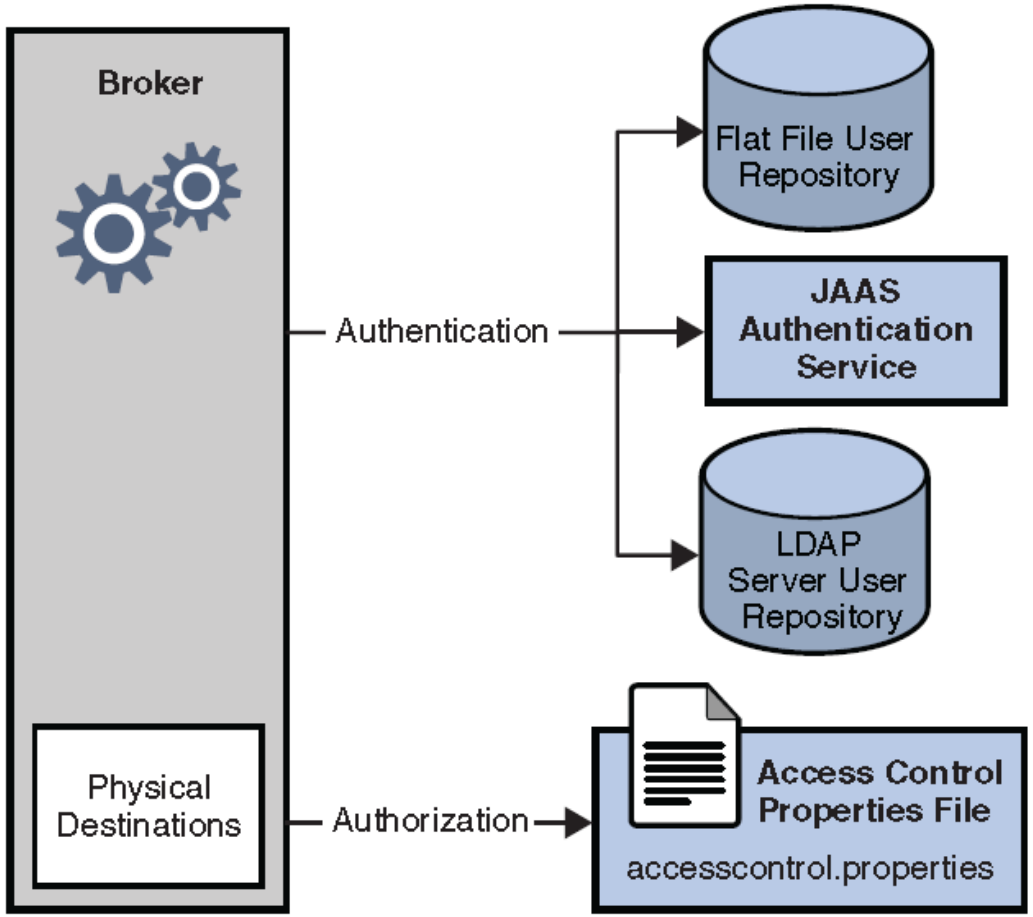
Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Authorization

- Users and groups
- Pre-defined groups
  - `admin`
  - `user`
  - `anonymous`
- Access control events
  - Connecting to a broker
  - Creating a message producer or consumer
  - Auto-creating or browsing a queue destination
- Configuration based
  - `accesscontrol.properties`

Authentication and Authorization



Source: OpenMessage Queue 4.5 Administration Guide

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Messaging Security

- Broker Level (Connection)
- Destination Level
- Message Level

Encryption

- Encrypted client-broker communication
- SSL based connection service
- Connection service types
  - `ssljmsservice` (TCP/IP)
  - `httpsjms` (HTTPS Tunnel Servlet with HTTP protocol)
  - `ssladmin` (TCP/IP)
  - Cluster connection service
  - JMX connector (RMI over TCP)

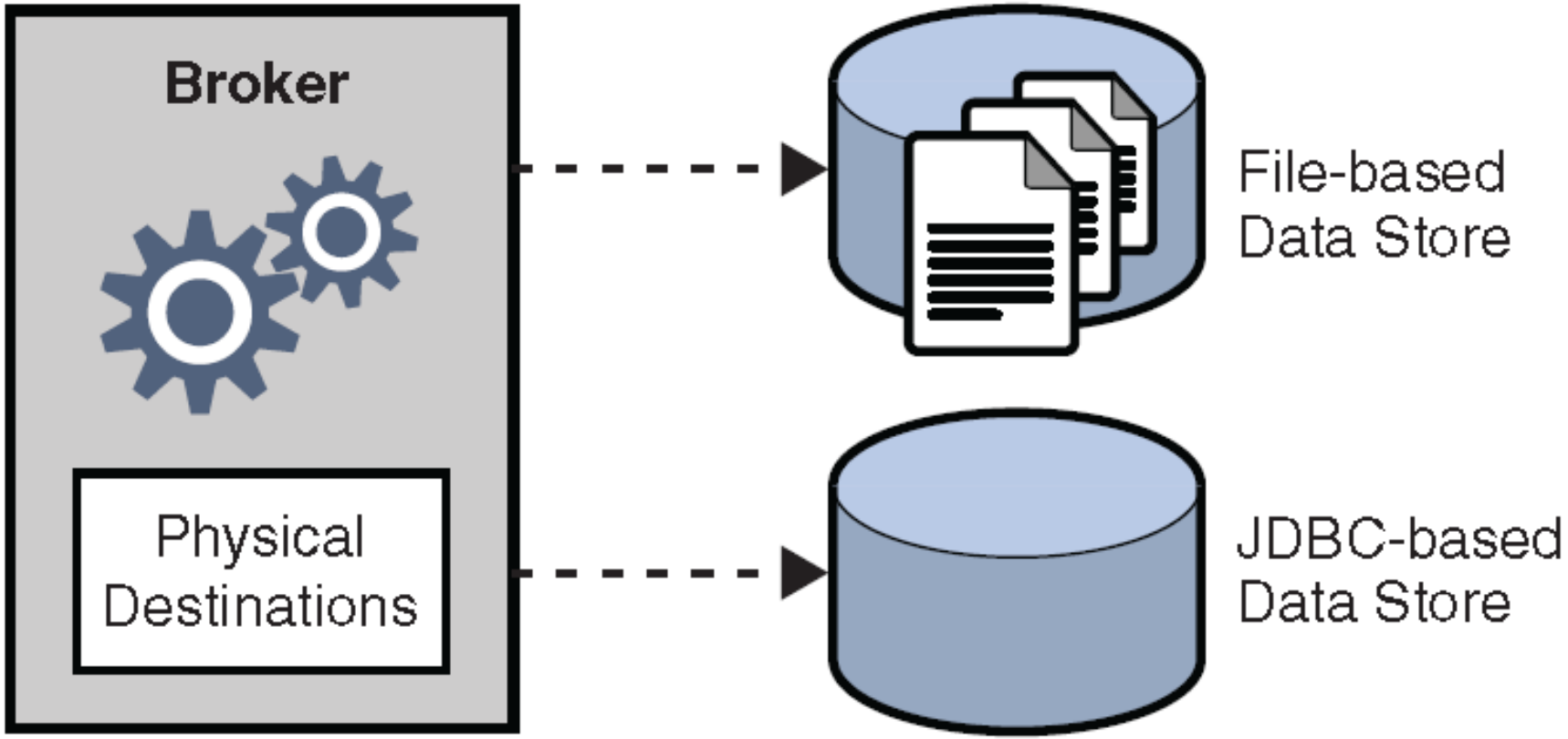
Encryption Best Practices

- Symmetric Key Algorithms:
  - AES with minimum 128 bit key length
- Hash Functions:
  - SHA-256
  - Always use a salt value (salted SHA, SSHA) especially for passwords to defend against rainbow table attacks
- Asymmetric or Public Key Algorithms:
  - rDSA with 1024 bit minimum key length
- Data Integrity:
  - Data Integrity/Data Signature or Message Authentication Code – HMAC (hash function-based message authentication code)
  - Use any underlying hashing algorithm since HMACs are substantially less affected by the potential for collisions that the related hashing functions alone
- Secure Network Communication:
  - SSLv3 or TLS to ensure the encrypted transmission of data between systems
- Security Standards Java API:
  - OWASP's ESAPI libraries

Message Persistence

- Configurable persistence
  - File or JDBC-based data store
- Securing a JDBC persistence store
- NoSQL DB for persistence (MongoDB)

Persistence Options



Source: OpenMessage Queue 4.5 Administration Guide

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Security Logging and Auditing

- Event management (SIEM) and forensics
- Critical for regulatory compliance (SOX, HIPAA, PCI, FISMA)
- Log events
  - Startup, shutdown, restart, and removal of broker
  - User authentication and authorization
  - Reset of a persistent store
  - Creation, purge, and destruction of destinations
  - Administrative destruction of durable subscribers
- Message Queue audit logging
  - JAAS based audit logging
- Performance:
  - MongoDB Logger

Security Logging Considerations

- What data needs to be logged for security analytics purposes?
- What should be the log format for business v. security logs?
- Do we need to store the security logs in a different file (a new log4j appender) so only authorized users (admin) will have access to it?
- How would the logs work with SIEM tool (if applicable)?

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

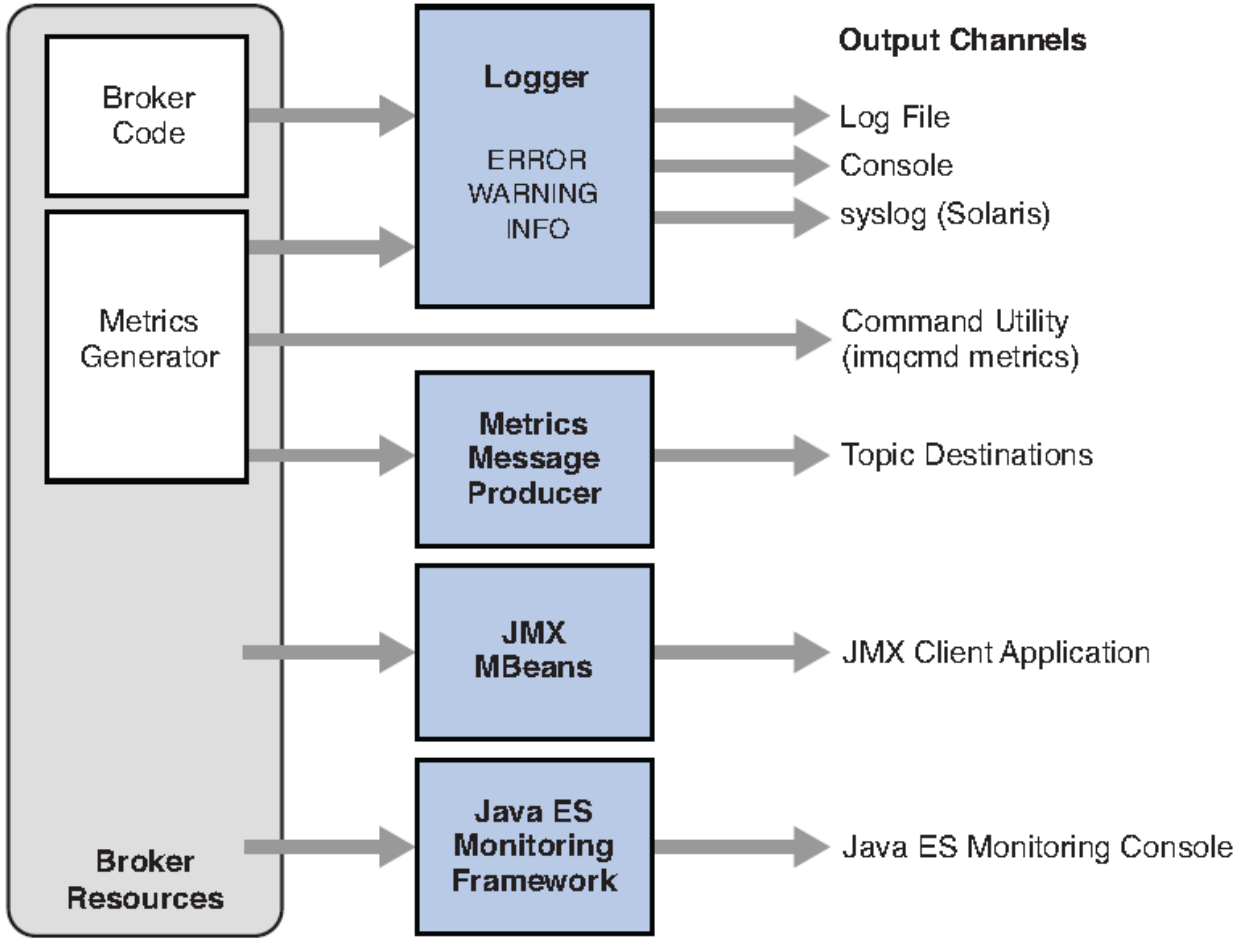
Monitoring

- Standards:
  - JMX
  - Remote JMX
- Secure JMX
- Tools:
  - JConsole/VisualVM

Monitoring Messaging Components

- Messaging components
  - Broker
  - Connections
  - Destinations
  - Producers
  - Consumers
  - Messages
- Monitoring Tools:
  - VisualVM
  - Open MQ Administration Console
  - Any standards based monitoring tool (e.g. JMX for Java)

Monitoring Services



Source: OpenMessage Queue 4.5 Administration Guide

Demo

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Best Practices

- Separation of messaging logic from application & business logic
- Messaging code should be agnostic to MQ container
- Declarative v. Programmatic message security
- Emerging trends in messaging architecture
  - Light-weight and Agile message brokers
  - Embedded brokers (for unit testing)
  - Built-in security, monitoring and clustering
- "Build in" security logging and monitoring capabilities into the product as a feature
- Involve Dev & Ops from early phases of project lifecycle

Agenda

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions

Conclusions

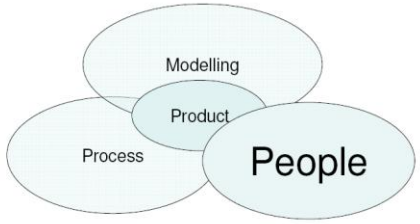
- Messaging security is critical
  - Message at rest, in transit and in use
- Messaging security considerations
  - Authentication, Role Based Access, Encryption
- Logging and Monitoring
- Messaging security support in Open MQ container
- Role of NoSQL DBs in messaging security use cases
- “One Size Fits All” fits nothing

Resources

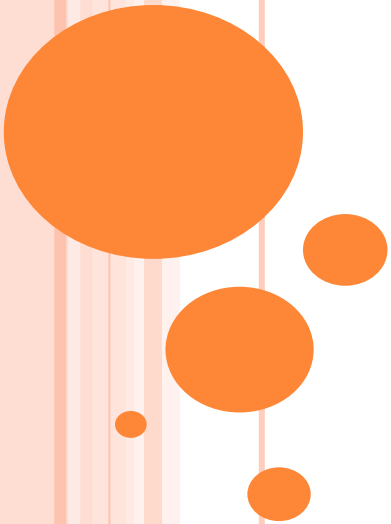
- Messaging Patterns
- Java Message Service (JMS)
- Open Message Queue
- Java Management Extensions (JMX) Technology
- Open Message Queue 4.5 Administration Guide
- Glassfish Samples
- GlassFish Server Open Source Edition 3.1 Security Guide
- GlassFish Server Open Source Edition 3.1 Administration Guide

Thank You

- Thank you
- Contact Information
  - http://www.infoq.com/author/Srini-Penchikala
  - srinipenchikala@gmail.com
  - @srinip
  - http://srinip2007.blogspot.com
- Questions?



Non-Linear First-Order Components
Alistair Cockburn



Bonus Slides

Security Implementation and Enforcement Using AOP

- Implement messaging security using Aspects
- Architecture:
  - Separate security event logic from application and business logic
- Tools & Technologies:
  - ActiveMQ
  - MongoDB
  - Esper
  - AspectJ and SpringAOP
- Demo

Integrating SOAP and MessageQueue