Mule ESB®

MuleSource®
the open source choice for SOA infrastructure™

Integration Simplified

Rich Remington

rremington@Rich-Software.com

Rich Software Inc

Topics

- Integration, SOA, and ESB
- What Mule ESB is (and isn't)
- Mule Architecture & Components
- Configuration & Deployment
- Enterprise Integration Patterns
- Main Demo
- Wrap-up & questions

1-SEP-2009

Goals

- ➢ Educate 1..* people about Mule ESB & how to simplify integration projects
- ➢ Give back to an awesome community
- ➢ Have a bit of fun
- ➢ Confirm "laziness" is a good thing!
  - – At least some of the time…
- ➢ Convince you it's safe, easy, and fun, to "get on the bus"

1-SEP-2009

Background

- ➤ Who am I?
  - – Mathematician wannabe turned software guy
  - – A bit lazy
- ➤ The bucket of water story
  - – Moral: don't work harder than you have to - it's okay to be lazy ;-)

1-SEP-2009

3 R's - Plan to be Lazy :-)

- ➢ Reduce
  - – Code complexity & bloat
- ➢ Refactor
  - – Move code to its proper place
    - Transformation, Routing, Service Component
- ➢ Reuse
  - – Plan ahead for service/component reuse
  - – Use pre-built components & solutions

1-SEP-2009

Integration Projects

- ➤ Why we integrate
  - – Users don't care where business functions and information reside
  - – We don't always own or control all the pieces
    - • intranet & external partners
  - – Cheaper than building new solutions from scratch
  - – Users want access to information from various devices - browsers, mobile phone, handhelds, ...

 1-SEP-2009

Integration Projects

- ➢ How we integrate
    - – Set up meetings with business domain experts
    - – Identify systems and processes
    - – More meetings…
    - – Map data flows then model
        - • Patterns are extremely helpful
    - – Yet more meetings…
    - – Define services & message formats
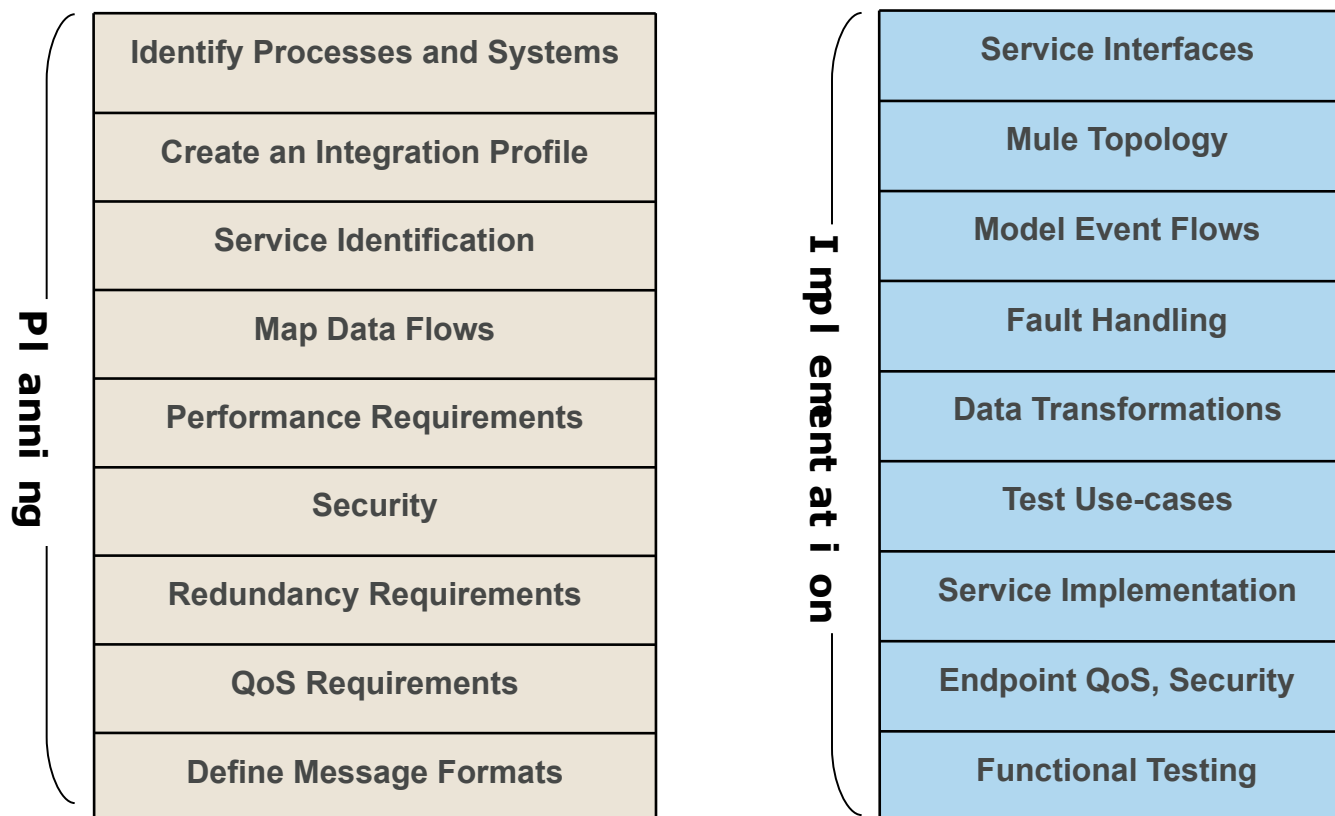   - – Start coding away…
- ➢ But, then what happens?

Integration Projects

➢ Three things in life are certain:
- Death
- Taxes
- Requirements will change often!

➢ Plan for change
- Design & build service/object interfaces for reuse
- Take advantage of loose coupling
- Switch implementations or protocols as needed over time, but not the interface

1-SEP-2009

Integration Checklist

Planning

| Identify Processes and Systems |
| Create an Integration Profile |
| Service Identification |
| Map Data Flows |
| Performance Requirements |
| Security |
| Redundancy Requirements |
| QoS Requirements |
| Define Message Formats |

Implementation

| Service Interfaces |
| Mule Topology |
| Model Event Flows |
| Fault Handling |
| Data Transformations |
| Test Use-cases |
| Service Implementation |
| Endpoint QoS, Security |
| Functional Testing |

1-SEP-2009

Integration Approaches/Technologies

➢ Custom, point-to-point solutions

  – Information silos live on

➢ Service Oriented Architecture (SOA)

  – Enables loose coupling of services that are OS & programming language independent

  – Web services (SOAP, REST), RPC, CORBA, etc.

➢ Enterprise Service Bus (ESB)

  – Helps orchestrate & manage the chaos

  – Good place to centralize general services

  – Encourages best practices

The Bus

"There are going to be times when we can't wait for somebody. Now, you're either on the bus or off the bus. If you're on the bus, and you get left behind, then you'll find it again. If you're off the bus in the first place — then it won't make a damn."

-- The Electric Kool-Aid Acid Test (1968)

Daily Double

➢ What are two of the 3 R's?

➢ What was the moral of the "bucket of water" story?

➢ Name the Integration Approach I'm suggesting you use

Audience Poll

➢ How many of you…

- – Need to integrate
- – Are using point-to-point integration solutions
- – Are doing SOA (SOAP, REST, RPC, …)
- – Not doing SOA, but planning to
- – Are using an ESB
- – Not doing ESB, but planning to

1-SEP-2009

Why use an ESB

- ➢ Integration Styles
  - – Batch
  - – File Transfer
  - – Shared Database
  - – Request/Reply
  - – Messaging
    - • Synchronous
    - • Asynchronous

- ➢ ESB Advantages
  - – Supports all styles
  - – Bridges legacy systems
  - – Modular architecture
  - – Simple and flexible
  - – Easy to test and maintain
  - – Scalable
  - – Can be a step towards SOA

1-SEP-2009

Why use an ESB (before)

1-SEP-2009
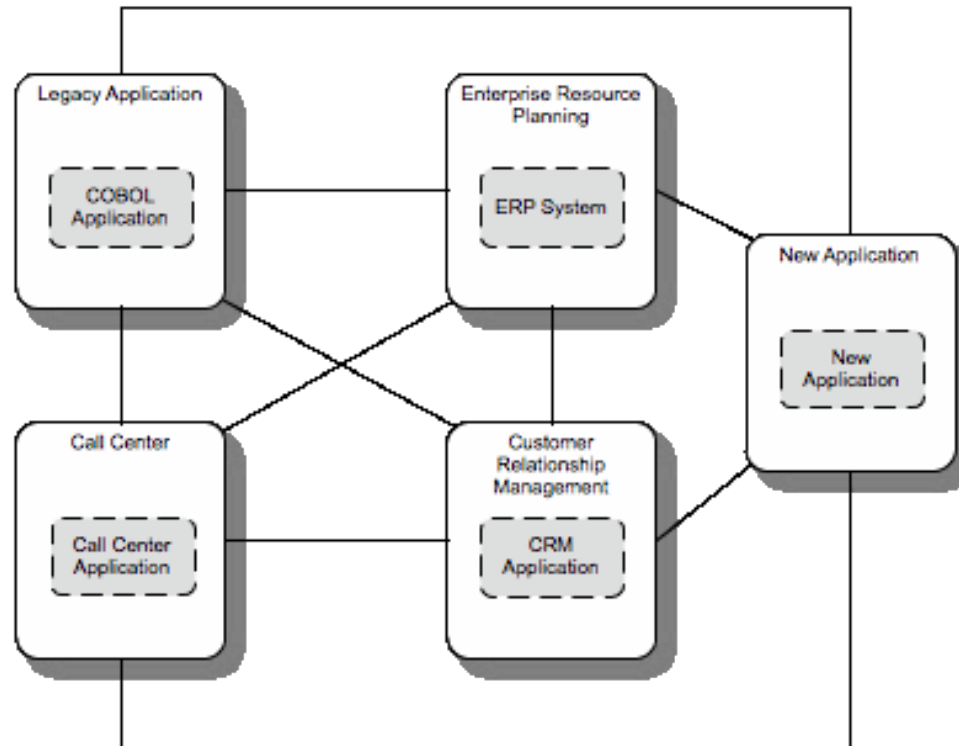
Why use an ESB (after)

What is Mule ESB?

"Mule is a lightweight integration platform and service container that allows you to quickly and easily connect your applications together. Mule provides a robust, secure and scalable platform to build enterprise applications offering an array of transports such as JMS, HTTP, Email, FTP, JDBC and many more. It also offers a rich set of features for web services, message routing, mediation, transformation and transaction management. Designed around the ESB (Enterprise Service Bus) concept."

-- www.mulesource.org

What is Mule ESB?

"Mule is a lightweight integration platform and service container that allows you to quickly and easily connect your applications together. Mule provides a robust, secure and scalable platform to build enterprise applications offering an array of transports such as JMS, HTTP, Email, FTP, JDBC and many more. It also offers a rich set of features for web services, message routing, mediation, transformation and transaction management. *Designed around the ESB (Enterprise Service Bus) concept.*"

-- www.mulesource.org

What is Mule ESB?

Service creation and hosting	• Expose transport end-points, EJBs, Spring beans, and POJOs as modular and reusable services • Host services as a lightweight service container
Service mediation	• Separate business logic from messaging • Shield service components from message formats and protocols • Enable location-independent service calls
Message routing	• Route messages based on content and complex rules • Filter, aggregate and re-sequence in-bound messages
Data transformation	• Exchange data across applications with varying data formats • Transform data as needed, based on need • Format messages across heterogeneous transport protocols

1-SEP-2009

Mule: Lightweight and Adaptive

Adaptive integration	• Standards-based and vendor-neutral • Seamless integration with existing infrastructure • Plug-in architecture allows for building block approach • Use with any application server or standalone
Development simplicity	• Simplif ed programming model • Ease of use through conf guration and control • Rapid development and deployment through code reuse
Open source advantage	• No expensive up-front license commitments • Access to source code allows customization and lowers risk • Vibrant developer community delivering best practices and modules

1-SEP-2009
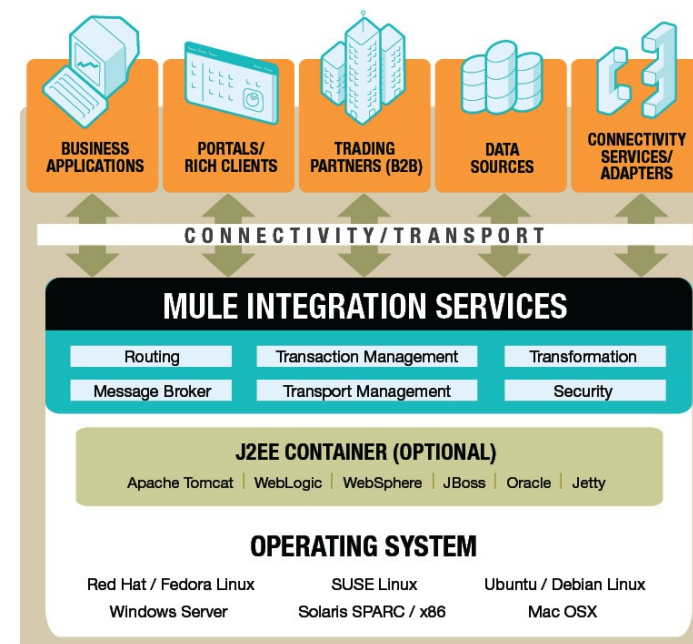
What Mule ESB isn't

- ➤ It does not place restrictions on your environment
- ➤ It is not message queuing, it uses other (JMS capable) products
  - – And, it does not require message queues
- ➤ It is not complex
- ➤ It does not need to be "the grand master of all integration endpoints across the enterprise"
  - – Think big, start small
  - – It plays nicely with other ESBs

Why use Mule ESB

JAVA	1.4 / 5 / 6 / 7		
APPSERVER	Apache Tomcat, WebLogic, WebSphere	Geronimo, JBoss, Oracle, Resin	Jetty, JRun
TRANSPORT	JMS, MQ Series, File, FTP, HTTP, HTTP Servlets, HTTPS, IMAP, In-Memory	JBI, JDBC, SOAP, SSL, Multicast, Oracle AQ, POP3, Remote EJB, RMI, SMTP	System I/O, TCP, TIBCO, TLS, VFS, UDP, XMPP, AS400 Data Queues, File system
INTEGRATION	Spring, EJB, GigaSpaces, JavaSpaces	JBI, JCA, JNDI, JOTM	JTA, PicoContainer, Plexus, HiveMind
WEB SERVICES	XFire, Axis	SOAP, REST	Glue
SECURITY	Acegi	JAAS	PGP
OTHER	BPEL, jBPM	JSR-223 (Scripting)	OGNL Filters, Quartz

- ➢ Decouples Business Logic
- ➢ Location Transparency
- ➢ Transport Protocol Conversion
- ➢ Message Transformation
- ➢ Message Routing
- ➢ Message Enhancement
- ➢ Reliability (Transactions)
- ➢ Security
- ➢ Scalability

1-SEP-2009

Why choose Mule

- No prescribed message format
  - XML, CSV, Binary, Streams, Record, Java Objects
  - Mix and match
- Zero code intrusion
  - Mule does not impose an API on service objects
  - Objects are fully portable
- Existing objects can be managed
  - POJOs, IoC Objects, EJB Session Beans, Remote Objects
  - REST & Web Services
- Easy to test
  - Mule can be run easily from a JUnit test case
  - Framework provides a Test compatibility kit
  - Scales down as well as up

1-SEP-2009

Mule ESB Architecture

➢ Architecture is based on concepts discussed in Enterprise Integration Patterns

1-SEP-2009
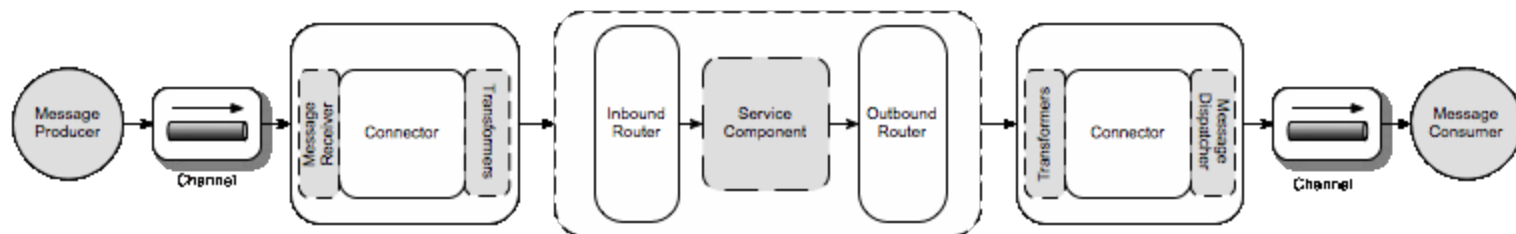
Mule Components

- Endpoint
  - Channel, MessageReceiver, Connector

- Transformer
  - Message transformation / enhancement

- Router
  - Message flow control (inbound/outbound)

- Service Component
  - Your integration logic lives here

1-SEP-2009

Mule Configuration

- ➢ Spring based
- ➢ mule-config.xml (default)
- ➢ Multiple config files using import

Mule Configuration
- Schema Import Statements
- Spring Beans Definition
- Mule Connectors
- Mule Filters
- Mule Transformers
- Mule Global Endpoints

Mule Model
Mule Service
- Inbound Router
- Component
- Outbound Router

1-SEP-2009
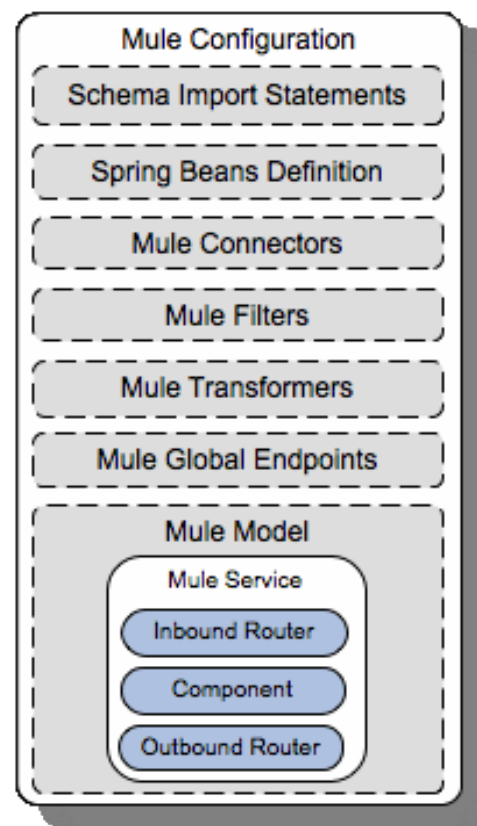
Daily Double

- ➢ Name two reasons to use Mule ESB
- ➢ Name one thing Mule ESB isn't
- ➢ Name two Mule Components

1-SEP-2009

Mule Examples (mini-demo)

- ➢ Download Mule
  - – http://www.mulesource.org/display/MULE/Download
- ➢ Configure the environment
  - – MULE_HOME, JAVA_HOME
- ➢ echo example
  - – options 1 & 3, show config

1-SEP-2009

Mule Deployment

- ➢ Standalone
- ➢ Embed in servlet container
- ➢ JCA 1.5 resource adapter

1-SEP-2009

Mule Deployment Topologies

➤ Enterprise Service Bus

1-SEP-2009
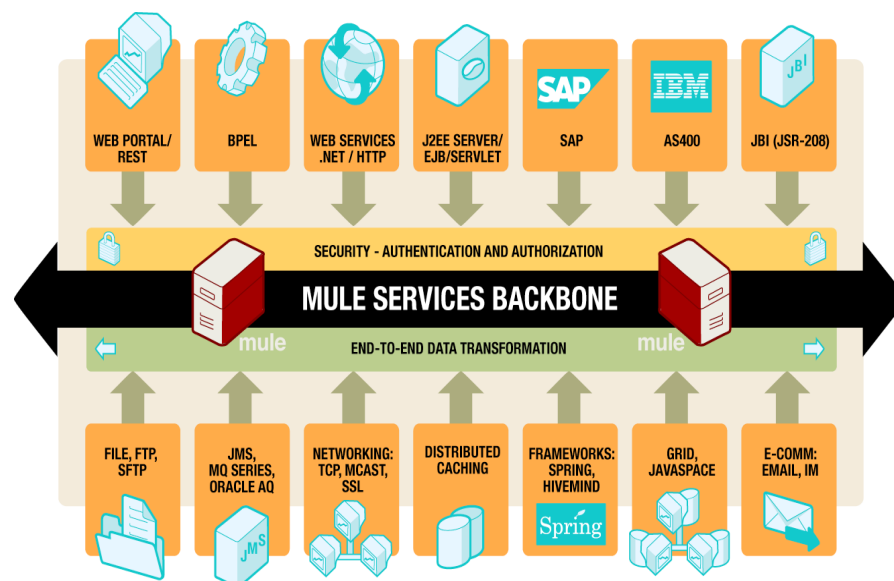
Other Deployment Topologies

Mule also supports:

➢ pipeline

➢ peer network

➢ client/server

➢ hub-and-spoke, etc.

Combine them to make an ESN

1-SEP-2009
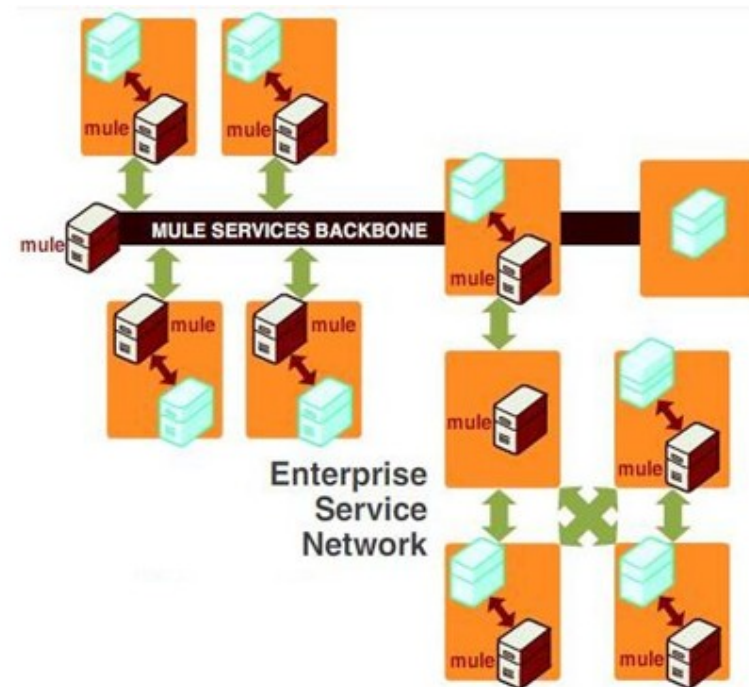
Enterprise Integration Patterns

- ➢ **Message Filter**
  - – Service component receives specific message types
- ➢ **Message Translator**
  - – Transformer: change format, tag names, etc.
- ➢ **Content Enricher (Message Enhancement)**
  - – add additional data to message
- ➢ **Content-based Router**
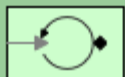 - – Route message based on contained data
- ➢ **Messaging Bridge**
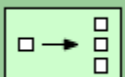 - – Send message between different protocols or messaging systems
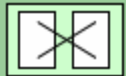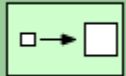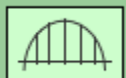
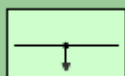1-SEP-2009

Enterprise Integration Patterns (cont.)
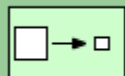
- ➢ Content Filter
  - – Shrink message to fewer data items
- ➢ Polling Consumer
  - – File directory, email, FTP
- ➢ Scatter-Gather
  - – Send to multiple endpoints / aggregate replies
- ➢ Splitter
  - – Batches, Lists, Collections, etc.
- ➢ Wire Tap
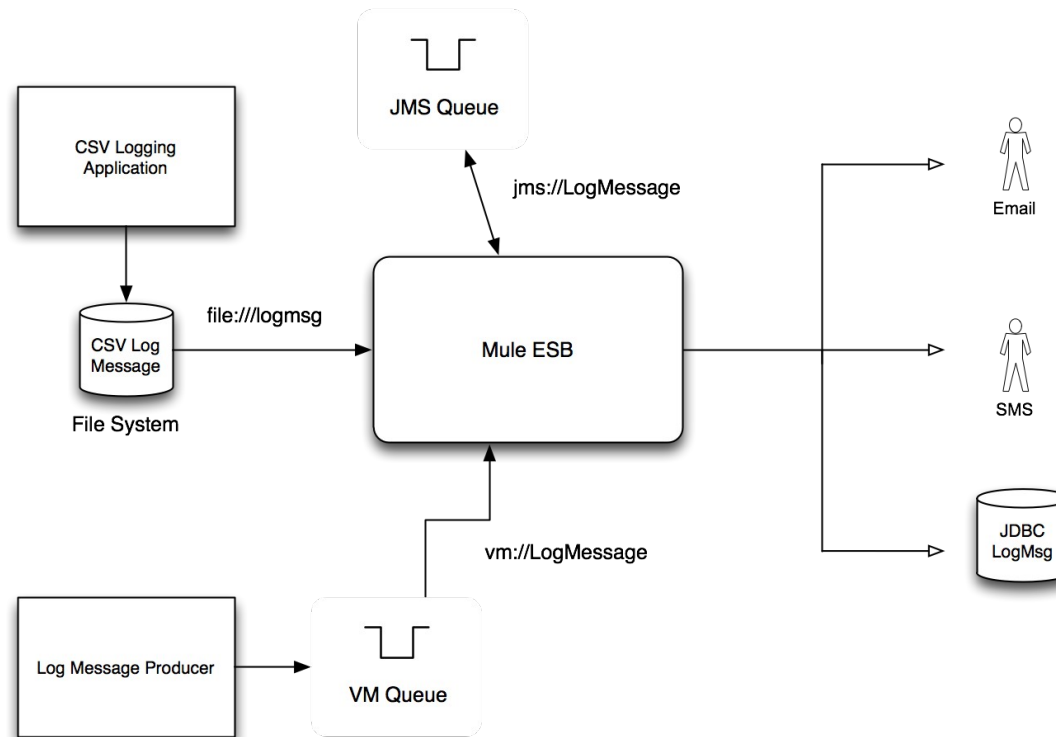 - – Inspect messages during flight (logging, monitoring, etc.)

1-SEP-2009

Main Demo

- Set the stage [see diagrams]
  - Describe scenario
  - Configuration
- Show demo
- Show code
  - Transformer(s)
  - Service Component(s)

1-SEP-2009

Main Demo - Scenario

1-SEP-2009

Other MuleSource Goodies

- ➤ MuleForge
  - – Mule extensions, mostly provided by community
    - Mule IDE, modules, tools, transports, etc.

- ➤ Mule Galaxy™
  - – SOA governance platform with an integrated registry/repository

- ➤ Mule ESB Enterprise
  - – Subscription based support that includes
    - Mule HQ™: systems monitoring / management
    - Premium connectors (e.g. IBM Websphere MQ)
    - Online knowledgebase & support links

1-SEP-2009

The Bus

"The bus came by and I got on, that's when it all began..."

-- "The Other One" by the Grateful Dead

So, get on the bus!

1-SEP-2009

Audience Poll

➢ How many of you…

- – learned something about Mule ESB?
- – learned something you didn't know about integration patterns?
- – had a bit of fun?
- – will think more about how to be lazy?

1-SEP-2009

Summary

- Integration Projects
- What Mule ESB is (and isn't)
- Mule Architecture, Components, Deployment
- Enterprise Integration Patterns
- Easy to implement integration scenarios
- Plan to be lazy :-)

Double Jeopardy

- ➢ Send me a use-case for Mule ESB within your organization using the contact form on the RSI website (www.rich-software.com/main/contact)
- ➢ Put "Mule ESB use-case" in the subject
- ➢ The use-case must be received within 48 hours of this talk to be eligible for a prize
- ➢ *I will buy two lucky people a free lunch that includes up to two hours of free business analysis on your use-case and/or mentoring on Mule ESB. And, you will receive one of the Manning books!*
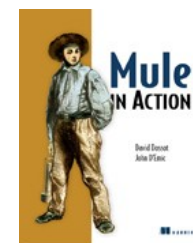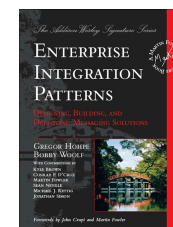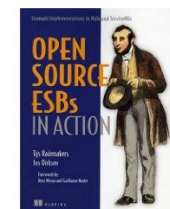
Rich Software Inc

Resources

➢ Mule ESB:
– http://www.mulesource.org (Community)
– http://www.mulesource.com (Enterprise)

➢ Open Source ESBs In Action
– http://www.esbinaction.com

➢ Enterprise Integration Patterns
– http://www.enterpriseintegrationpatterns.com

➢ Mule In Action (July 2009)
– http://www.manning.com/dossot

1-SEP-2009

Thank you!

Questions?

1-SEP-2009