MuleSource

the open source choice for SOA infrastructure

SOA Governance: 5 common mistakes and how to avoid them

Introduction

- Dan Diephouse, MuleSource Software Architect

Agenda

- Governance, Registry, and Repository overview
- 5 Common Mistakes of SOA Governance
- Questions

What is SOA Governance?

MuleSource

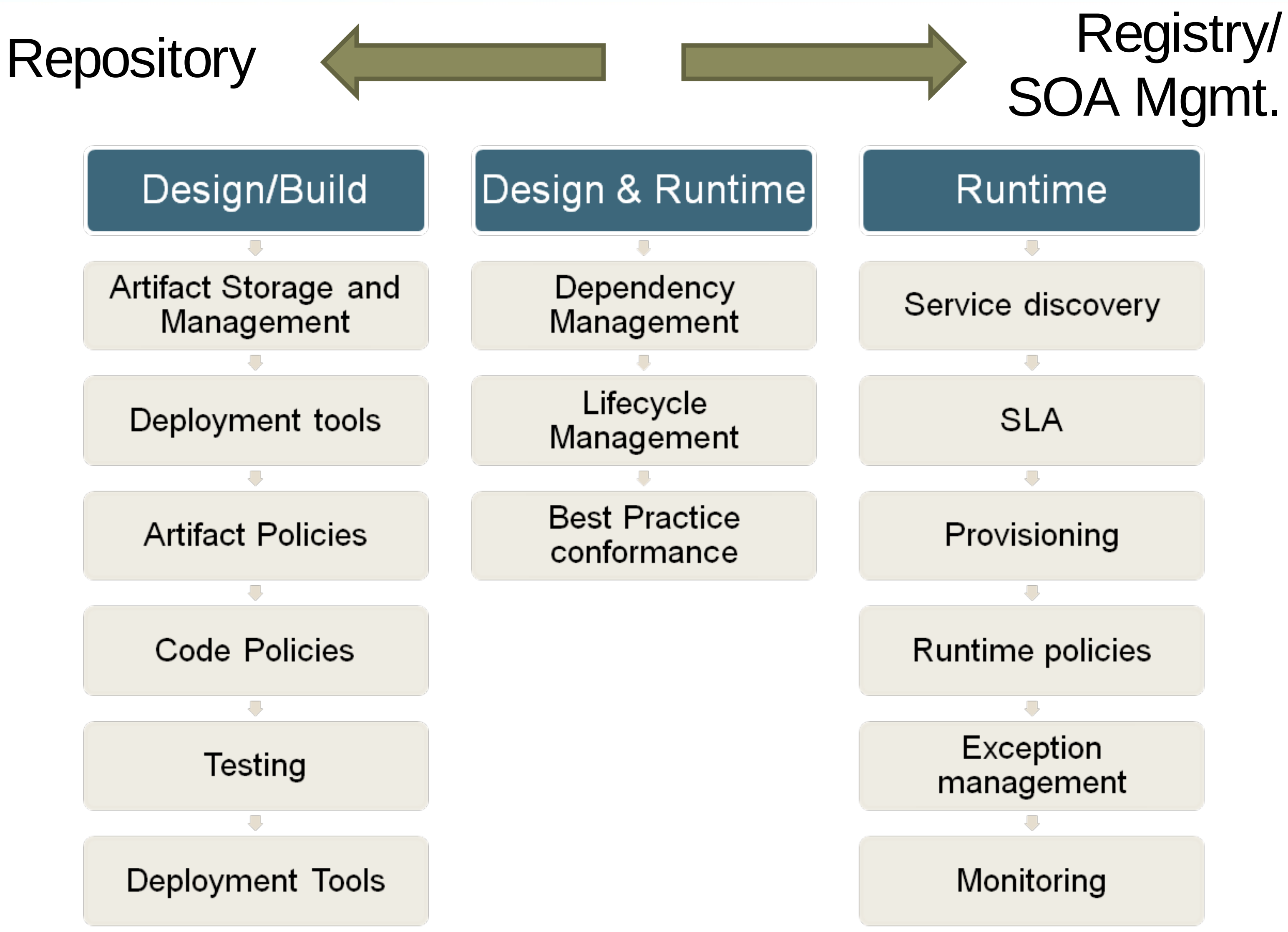
- SOA Governance refers to processes that ensure that services and applications are developed so that they are aligned with:
  - Best practices
  - Business Requirements
  - Laws
  - SLAs
- Governance spans people, policies, processes and tools

Benefits of SOA Governance

MuleSource

- Improved adherence to best practices, requirements, and laws
- Increased service and application reuse
- Lower application maintenance costs
- Decrease time to market
- Ability to make informed decisions about what services/applications/artifacts can be reused

Governance tools from design to runtime



What is a Repository?

MuleSource

- Storage of artifacts
  - Mule configurations, Applications, WSDL, WS-Policy, etc
- Dependency Management
- Lifecycle management
- Enforce artifact policies
- Value:
  - Gain visibility into other applications & services which can be reused
  - Collaborate on services or applications
  - Make applications, schemas, WSDLs, etc available for easy reuse
  - Ensure that artifacts meet governance requirements

What is a Registry?

MuleSource

- System of Record
- View runtime information such as
  - Which services are running
  - SLA information
- Track dependencies from an artifact all the way down to a particular node. i.e. Machine X is actually using this WSDL
- Track service lifecycles:
  - Which services can I use?
- Enforce runtime policies:
  - All endpoints must be SSL encrypted
  - Services must be WS-I compliant

Not…

- Excel is not a Registry
- A folder on your desktop is not a Repository
- SVN is not a Repository either

What is SOA Management

- Provisioning
  - Add new nodes or clusters
  - Roll out new versions of applications
  - Push from development to production
- Business activity monitoring

What are the biggest challenges you have faced in deploying governance solution?

1. Too expensive
2. I find existing tools to be too heavy-weight or difficult to use
3. Existing tools require too many changes to my organization or processes
4. I require significant customization, and I don’t want to use proprietary solutions
5. Don’t have time to implement a “big-bang” approach to governance
6. I haven’t looked into SOA governance

MuleSource

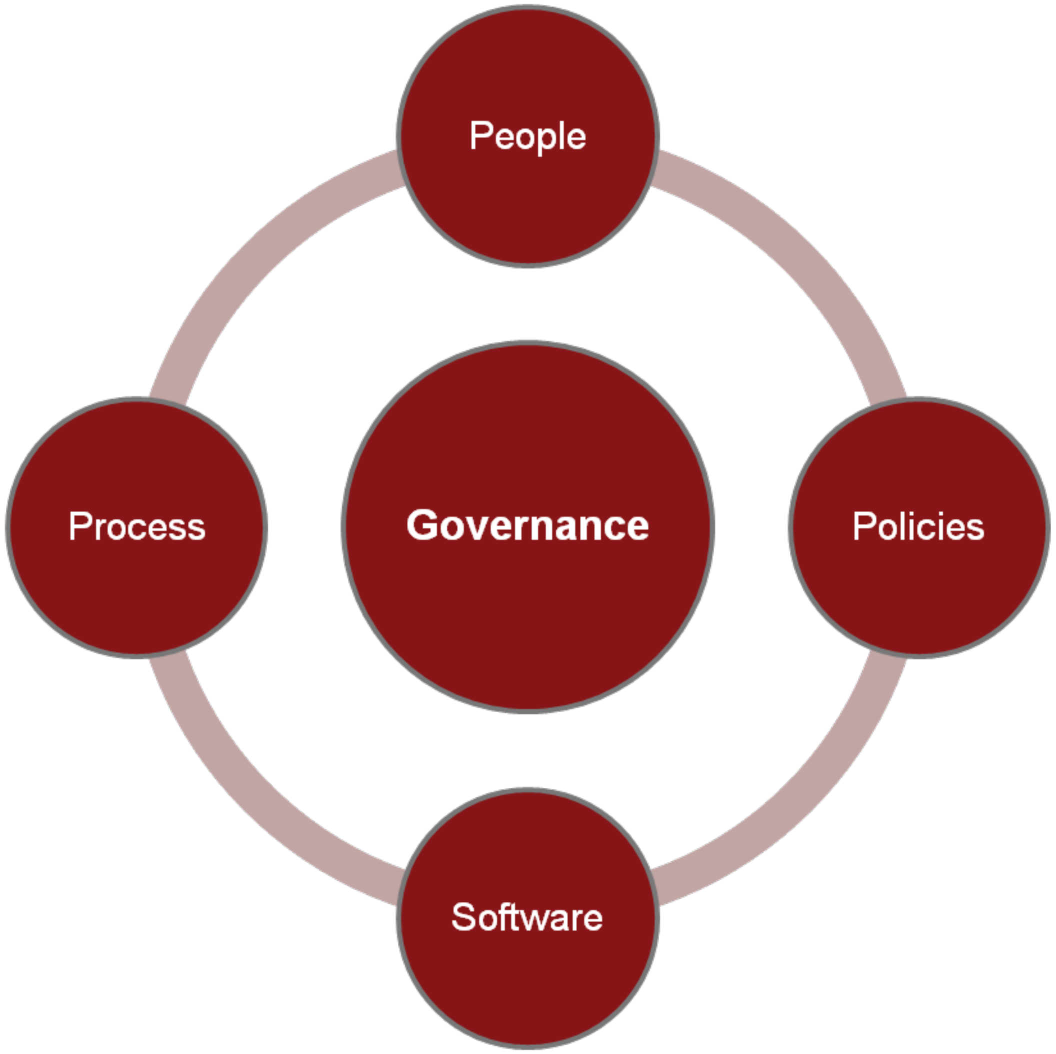
the open source choice for SOA infrastructure

5 Common Mistakes

Mistake #1

THINKING GOVERNANCE IS A PRODUCT

Products are rarely solutions to the whole problem



* Some of these were taken from *Service Oriented Architecture*, by Thomas Erl

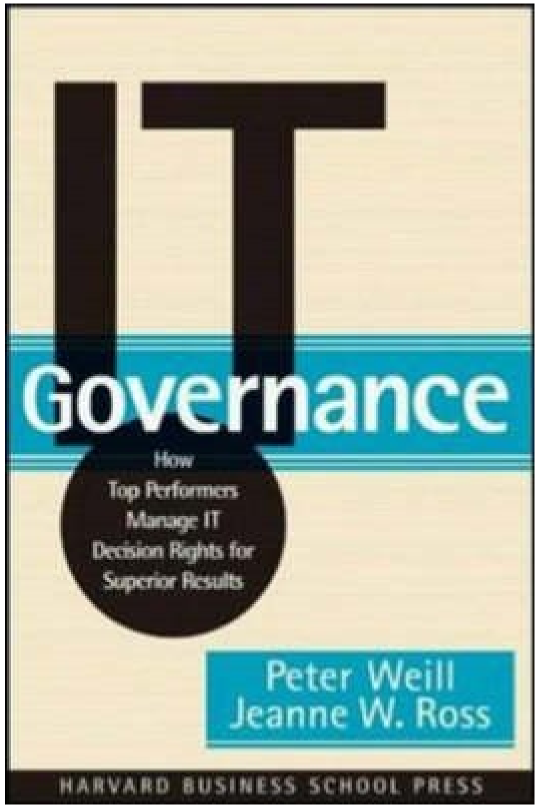
Governance

- Goal – encourage desirable use of IT
- All stakeholders have necessary input in decision process (executives, IT staff, customers, etc)
- Coordination of projects
- Metrics
- Change Management
- Spending, Incentive systems
- Exception Management

IT Governance

- IT Governance

  by Peter Weill, Jeanne Ross


IT
Governance
How Top Performers Manage IT Decision Rights for Superior Results
Peter Weill
Jeanne W. Ross
HARVARD BUSINESS SCHOOL PRESS

MuleSource

Mistake #2

REINVENTING THE WHEEL

Decentralizing common artifacts

- Scattered artifacts translate to:
  - Time wasted searching for service interfaces and schemas
  - No central authority for the published service interface/schema
  - Discourages the discovery and reuse of services/schemas/applications

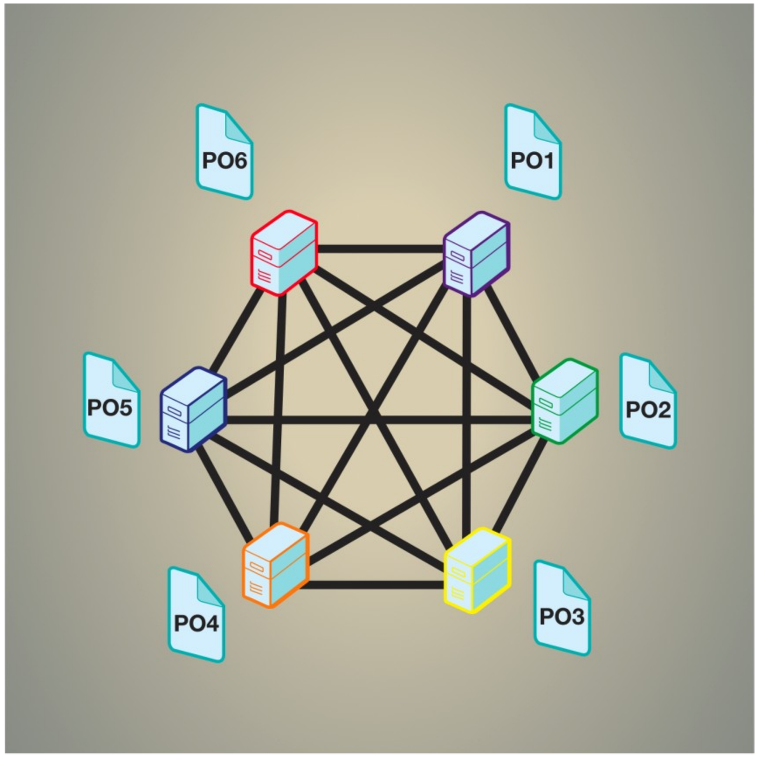
* Some of these were taken from *Service Oriented Architecture*, by Thomas Erl

A centralized authority

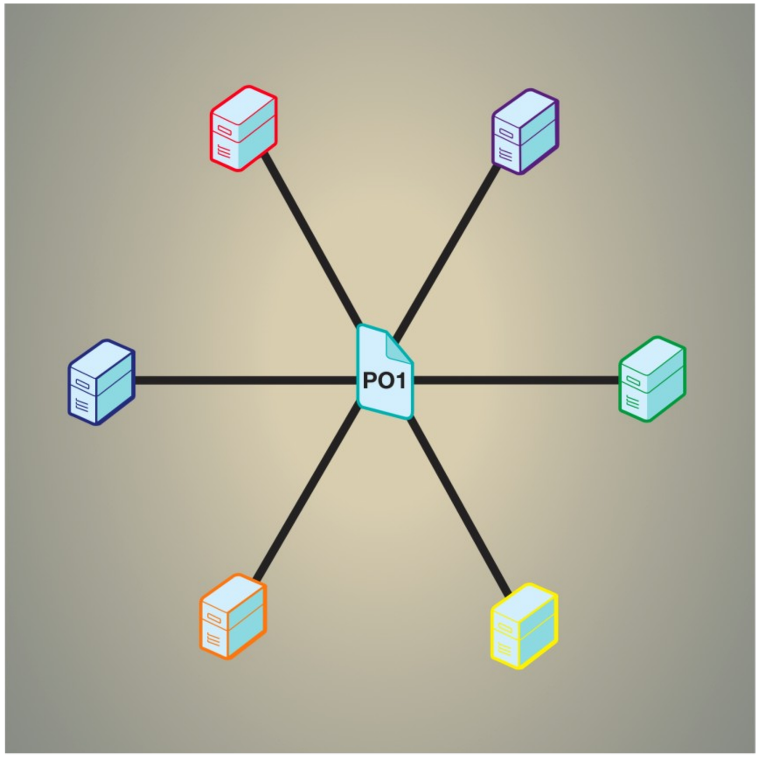
MuleSource

- Creating a centralized repository/registry allows others to
  - Reuse your artifacts
  - Utilize an authority for what services, schemas, and applications are available
- Allows tracking of not just what is available, but the state of its availability
  - i.e. What services are available which are in production?
- Often, the usage of a registry/repository must be *required* or it may not be used and no benefits will be realized

N^2 Problem



Disparate Purchase Order Representations



Universal Representation of Purchase Order

Problems with reinventing the wheel

- Increased application development time
- Services aren't as robust
- Increased maintenance due to service and application proliferation

How to stop the reinvention cycle

MuleSource

- Ensure you're sharing requirements and collaborating on services and requirements
- Watch for new services and applications across an organization
- Reuse applications, services, and artifacts whenever possible
- Use a centralized repository/registry to store information

Mistake #3

HOPING FOR BEST PRACTICES

Hoping for best practices

- Developers will not automatically align themselves
- Developers may not know best practices
- Net negative results
  - Badly written and/or buggy code
  - Inconsistent security, or even worse, no security
  - Non interoperable services

What are some best practices?

- Best practices can be enforced anywhere from the build (Maven/Ant) to the Registry (Galaxy) to the process/people level.
- Build time:
  - Checkstyle
  - PMD
- Design Time:
  - Check for backward compatibility
  - WS-I Basic Profile Compliance
  - Require Documentation
- Run Time
  - Security Requirements
  - WS-I Basic Profile Compliance

Mistake #4

FORGETTING ABOUT YOUR CONSUMERS

Consequences of consumption

MuleSource

- What changes can I make to my service or schema without affecting others?
  - Will you break somebody else’s service?
  - Is anyone even using your service?
  - Has your service, unbeknownst to anyone, become a critical part of the company’s infrastructure?
- What features are others using?
- What features do people want?

Tracking Consumers

MuleSource

- Dependency Management
  - What other artifacts directly depend on the ones I’ve published?
  - What artifacts do my artifacts apply on?
- Galaxy can track which artifacts important other ones in WSDLs, Schemas, and WS-Policy files
- Use Maven and a Maven repository tool to track build time dependencies

Mistake #5

INCONSISTENT APPLICATION DEPLOYMENT STRATEGY

Common Application Deployment Problems

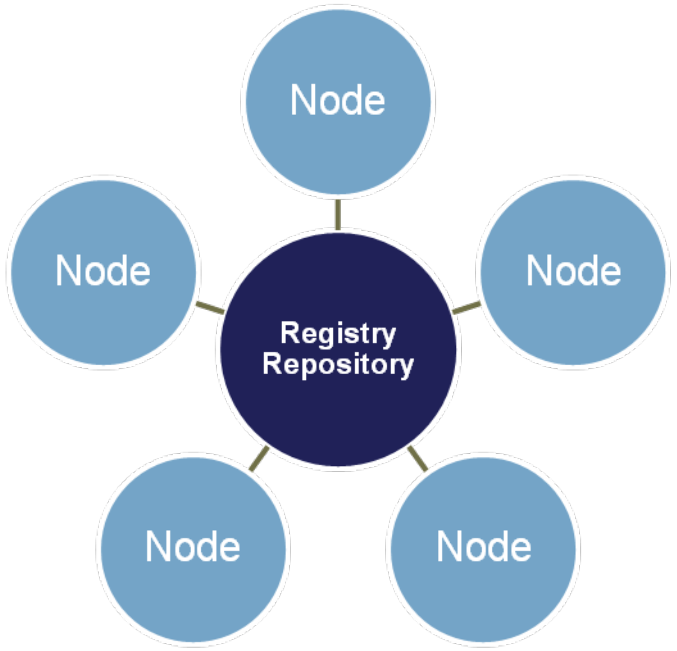
MuleSource

- Many application deployment strategies are ad-hoc, not well documented, and only understood by one person
- They involve error prone manual processes
  - Developer actions should be automated as much as possible
- Rolling back to a previous version is often is impossible
- Often confusion about exactly what version of libraries/configurations is in production

Application management with a Reg/Rep

MuleSource

- Solution: Use a centralized place for software/server updates
- Pull applications, configurations, etc directly from the repository
- Results in
  - Improved manageability
  - Smoother deployments
  - Easier upgrades/rollback
  - Ability to track which versions of an application are in use



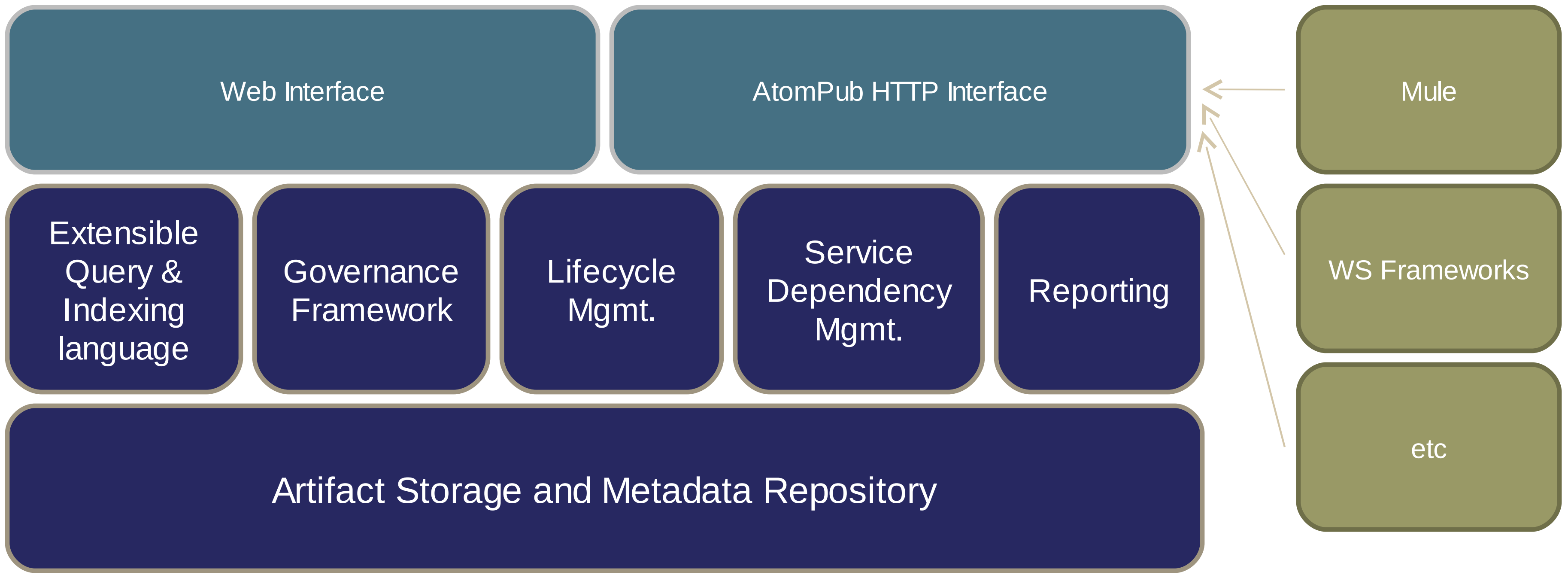
MuleSource

the open source choice for SOA infrastructure

Mule Galaxy

Open Source SOA Governance

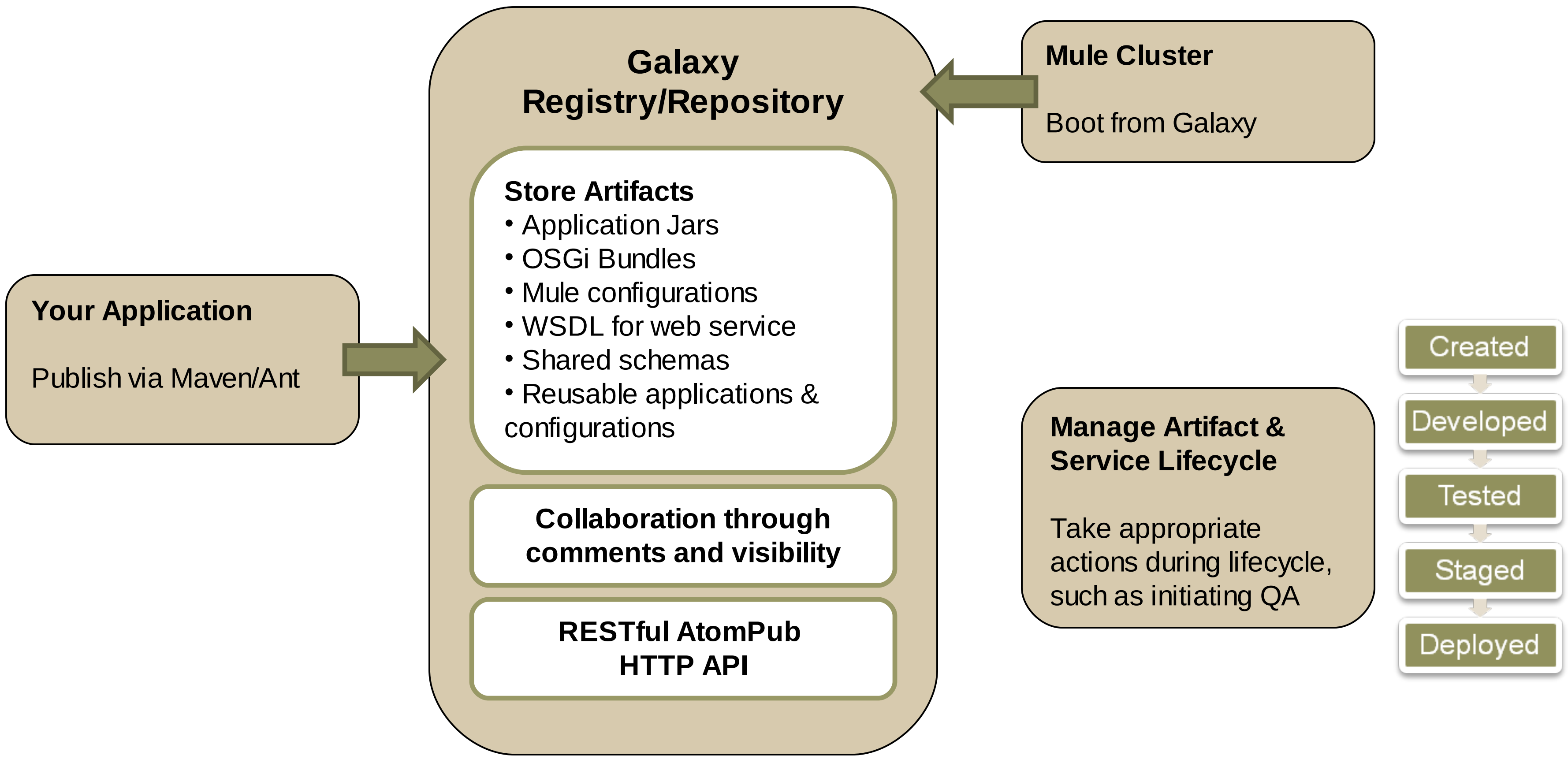
Galaxy Architecture



Why Galaxy?

- Open Source
  - Download, get started now!
  - Extend it to support your own custom needs
  - Your feedback will be incorporated into the product
- Works with web services *and* non web services
  - Store any artifact inside the repository
  - Extend the repository to support your own artifacts
- Extensible query language
- Dependency and Lifecycle management
- Record of changes inside repository
- Tight integration with Mule

Where does Galaxy Fit?



MuleSource

the open source choice for SOA infrastructure

Galaxy Demo

Open Source SOA Governance

Questions?

- Galaxy: http://mulesource.com/products/galaxy.php
- Dan’s Blog: http://netzooid.com/blog
- Questions: dan@mulesource.com