JUDCon

JBoss Users & Developers Conference

Boston:2011

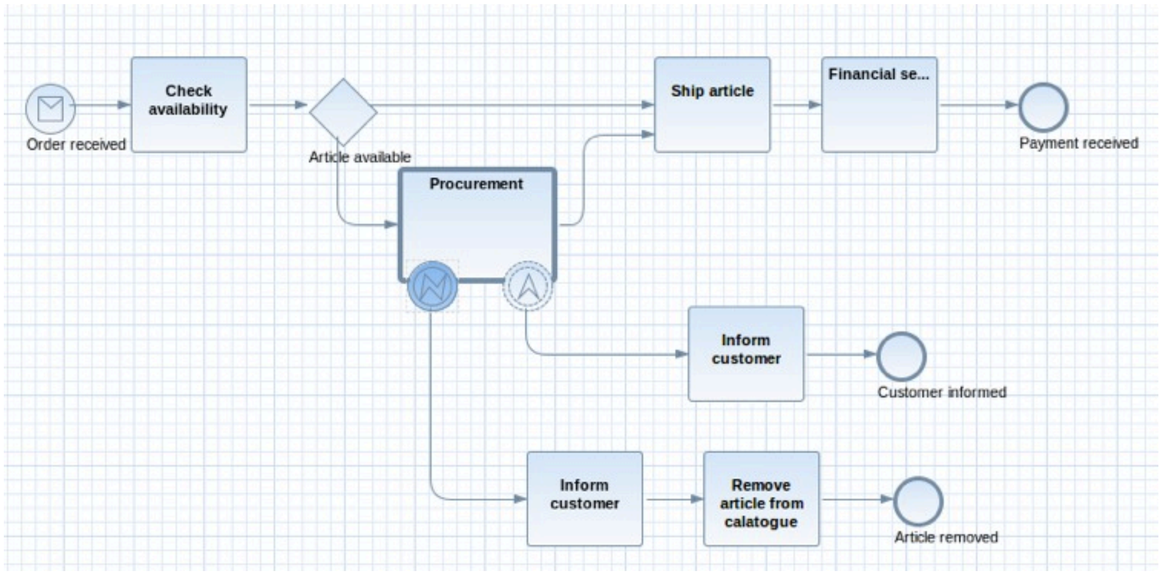
Governing Services, Data, Rules, Processes and more

Randall Hauch
Project Lead, ModeShape
@rhauch @modeshape

Kurt Stam
Project Lead, Guvnor
@guvtalk

Scenario 1

Build business processes using existing services



Scenario 1

Build business processes using existing services

- What services are available?
- What stage are they in?
  dev? test? QA? prod? EOL?
- What services are planned?
- What schemas are used?
  common? standards?
- What do message look like?
- What's already using each service?
- What does each service use?
  other services? databases? external sites?
- What are the performance metrics of each service?
- What's impact of changes?
- Who can approve such changes?
- What SLAs are in place? And for what services?
- Who's the primary contact?

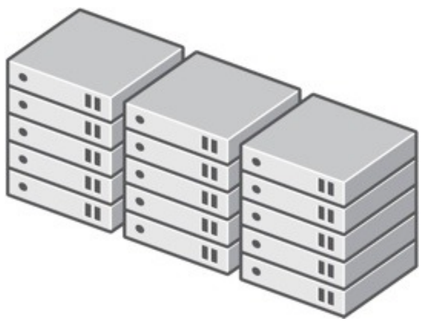
Scenario 2

Integrate DBs acquired thru M&A

multiple ETL systems

app-specific DBs

in the cloud



multiple data centers

regional databases

remote office databases

Scenario 2

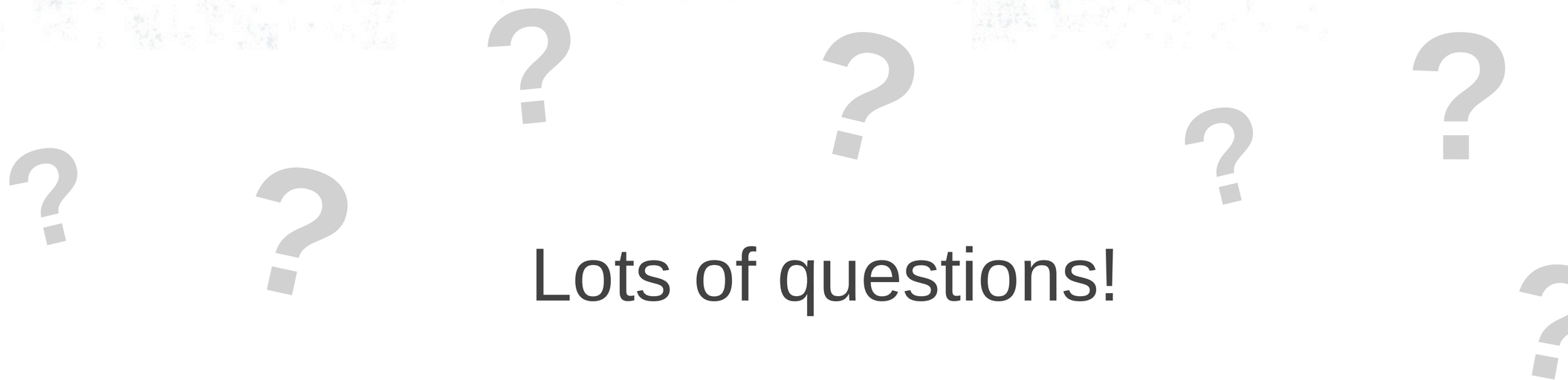
Integrate DBs acquired thru M&A

- What databases are available?
- What stage are they in?
  dev? test? QA? prod? EOL?
- What schemas are deployed? On which DBMSes?
- Where is each DB instance located?
- What applications use each DB?
- What about data virtualization?

- How many entities deal with customer? accounts? records? sales? purchases? products? etc.
- What data transformations are performed?
- What's impact of changes?
- Who can approve such changes?
- What SLAs are in place?
- Who's the primary contact?

A few other scenarios

- Determine why we're not meeting our SLAs
- Determine impact of a bug fix in a critical service
- Determine impact/exposure to security vulnerability
- Determine impact of changes new version of API
- What's our process for rolling out a new deployment

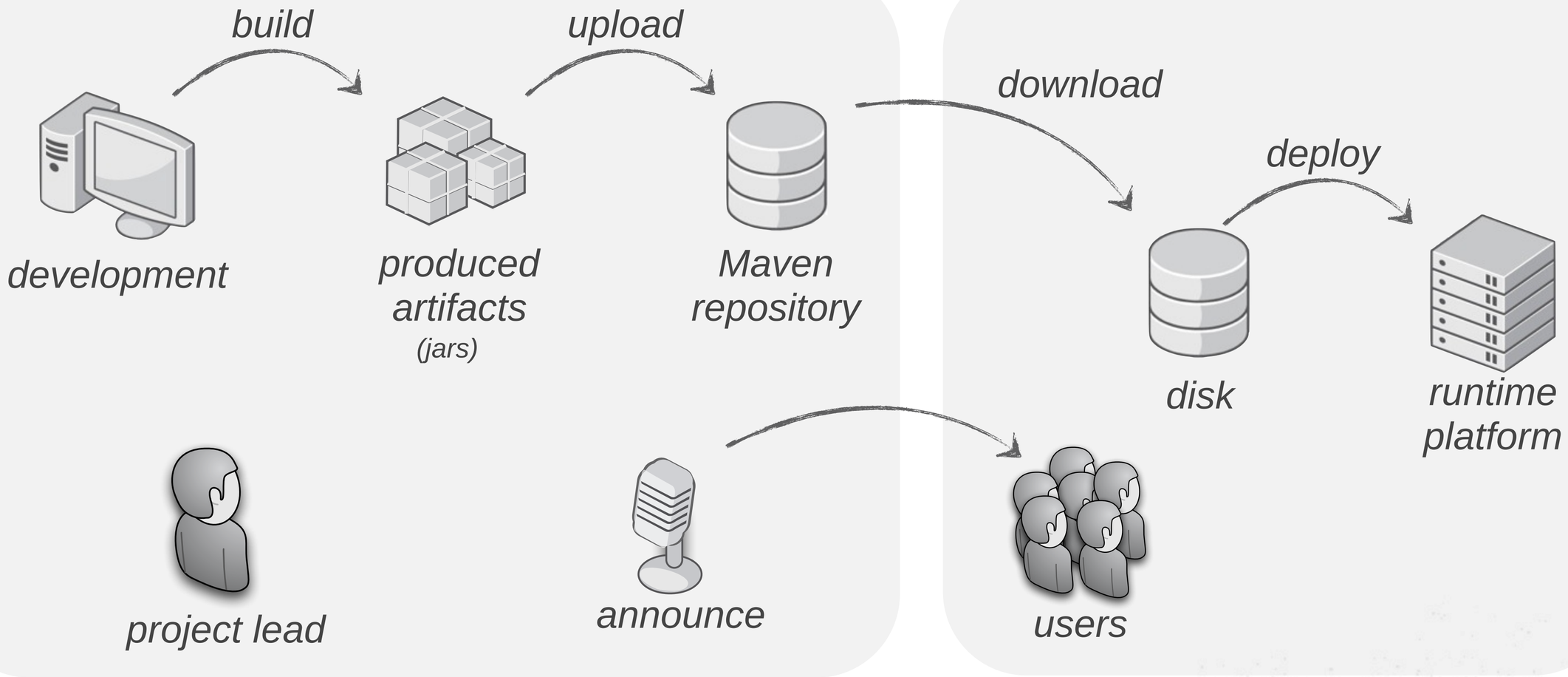


Finding answers to these questions **takes time** and **costs money**

Proper governance helps answer these questions

Open source “production”

a lifecycle of *project* deliverables



More than just JAR files & manifests

- JARs and manifests are required for runtime deployment
- They don't tell us everything we need
  - interchange formats and structures (e.g., XSDs, JSON, XML)
  - data structures (e.g., DDL files, XSDs)
  - interface/API representation formats (e.g., WSDL)
  - semantic models (e.g., BPMN, rules, models)
- Also need to track other things
  - policies (e.g., SLA, security)
  - stages (e.g., planned, in test, QA, ready for production, deployed, retired)
  - acceptance, reviews and approvals

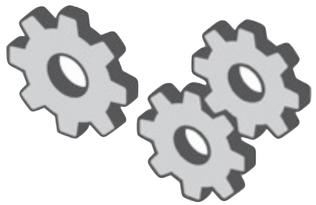
Governance repository

track artifacts and relationships

processes

rules

virtual databases



business services

data services

databases

policies

message structures

Governance repository

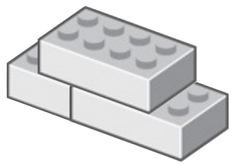
track artifacts and relationships throughout lifecycle

development

testing

support

acceptance



integration

quality

documentation

deployment

change

security

Governance repository provides

- Tracking artifact states
- Transitioning through lifecycle model
- Managing relationships
  - between artifacts and components
  - automatically deriving where possible
- Automation
- Notification
- Visibility and access
- Search and structured query
- Versioning
- Security and audit capabilities

How do we get there?

Drools Guvnor

- Very successful
- Governs rules, from development to deployment
- Starting to also govern processes
- Some extension capability
- Great starting point
- Lots of best practices and lessons

Demo

Drools Guvnor

- Great starting point
- Lots of best practices and lessons
- But limited to rules and processes
  - user interface, semantic concepts, mental model
- Not possible to use for SOA services or Data Services

Current Guvnor

Drools

guvnor-web-app

guvnor-repository

JCR

Guvnor roadmap

Drools

guvnor-web-app

guvnor-repository

JCR

Make more general-purpose

Guvnor roadmap

Drools

guvnor-web-app

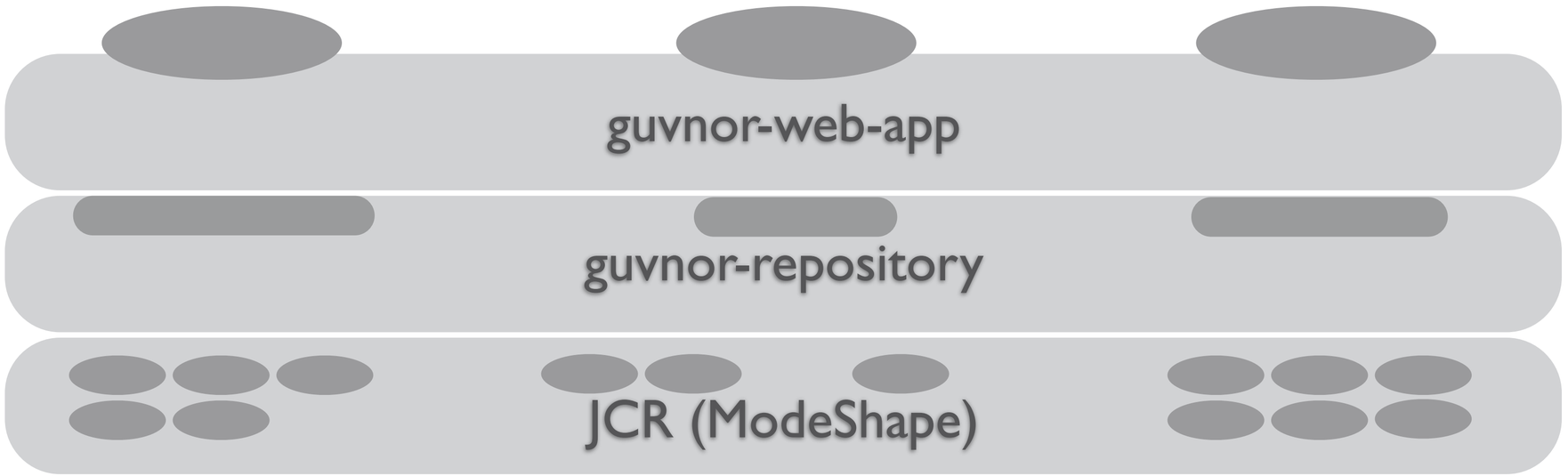
guvnor-repository

JCR (ModeShape)

Additional features when using ModeShape

Guvnor roadmap

Services ***Rules & Processes*** ***Data***

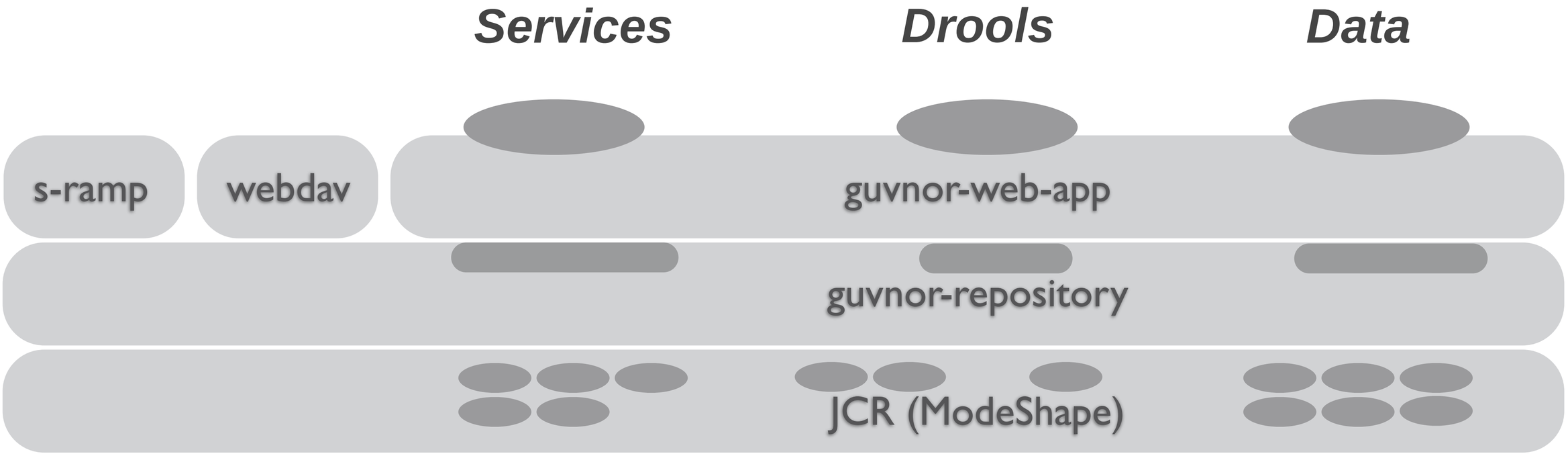


*Add **services**-specific functionality via extensions*

*Add **rules**-specific functionality via extensions*

*Add **data**-specific functionality via extensions*

Guvnor roadmap



Support client access over REST & WebDAV

S-RAMP

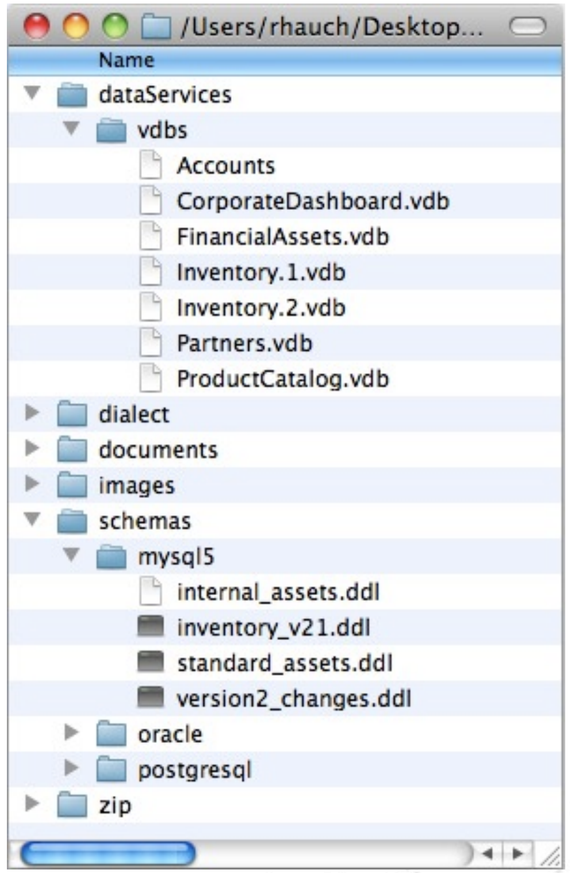
SOA Repository Artifact Model & Protocol

- Currently in work by OASIS Technical Committee
  - IBM, HP, SOA Software, TIBCO, Red Hat, WSO2 & others
- Common data model for SOA repositories
  - artifacts, derived information, taxonomies, query model
- Interaction protocol using AtomPub
  - CRUD, batch ops, query, notifications

The goal is interoperability for SOA tooling

WebDAV

- Remotely access repository
  - mount repository as network share
  - or access via HTTP
- Navigate, create and delete folders
- Download, upload or delete files
- Use in scripts for automation
  - particularly useful for deployments



Why ModeShape

- Content Repository for Java (JCR) API
  - JSR-170 (1.0)
  - JSR-283 (2.0)
- Easily clusterable *(JGroups FTW!)*
- Plays nice with JBoss technologies
  - uses Hibernate, JGroups, Infinispan, RESTEasy, PicketBox, etc.
  - deploy as a service within JBoss AS
  - monitor with Embedded Console and JON
- Rich query capabilities (beyond spec)
- JDBC driver for querying content
- Connectors for storing & accessing content
  - JDBC (via JPA/Hibernate), Infinispan, SVN, file system, JDBC metadata
- Sequencers ...

JCR stores files

LoanService.wsdl

- Files are content
  - stored as binary data
  - metadata (MIME type, who, when,...)
- Optionally version some/all files
- Extract text for searching

but nothing else

ModeShape understands files

LoanService.wsdl

embedded XSD types

creditInformationMessage
- name (xsd:string)
- amount (xsd:integer)

approvalMessage
- accept (xsd:string)

loanServicePortType
- input (creditInformationMessage)
- output (approvalMessage)

loadServiceBinding
- operation ("request")
  with input, output, fault

...

- Files are content
  - stored as binary data
  - metadata (MIME type, who, when,...)
- Optionally version some/all files
- Extract text for searching

but can also "sequence" these files:

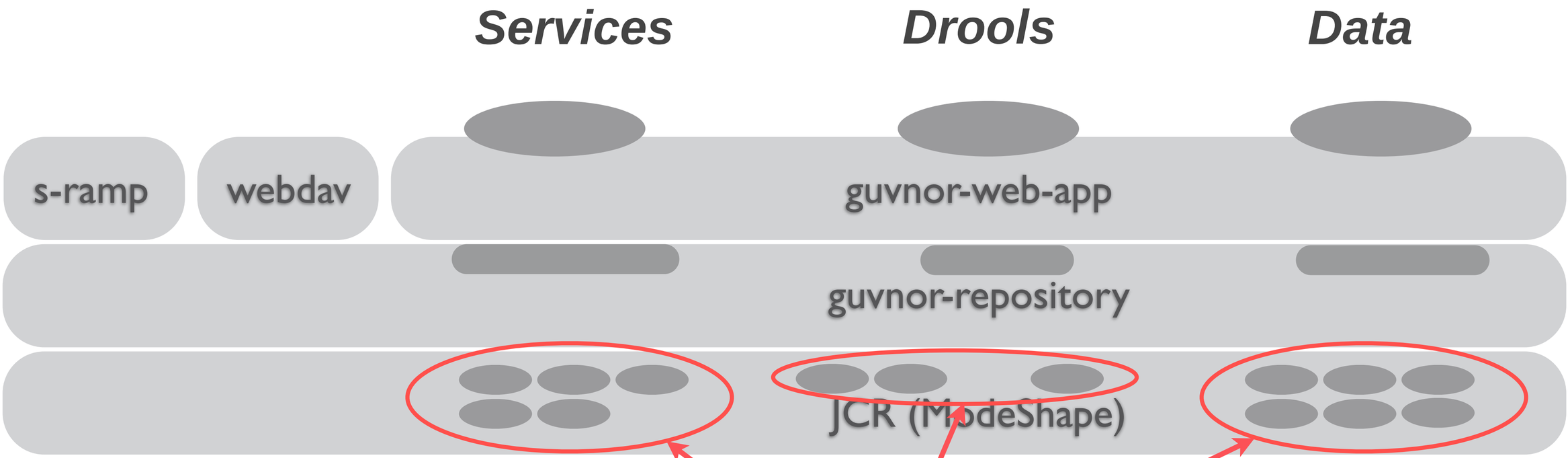
- Determine MIME type
- Parse files
- Extract structure
- Store that back in the repository
  - where it can be queried & accessed

ModeShape understands files

- ZIP, JAR, EAR, and WAR files
- DDL files
- Java source and class files
- Teiid Relational models and VDBs
- Text files (CSV and delimited)
- JCR Compact Node Definition
- XML documents

- Plans for more
  - WSDL documents
  - XSD documents
  - BPEL processes
  - Drools rules

Guvnor roadmap



ModeShape sequencers for domain-specific file types

Summary

- Track the artifacts we produce, throughout their lifecycle
- Provide access to the governed artifacts (via REST/WebDAV)
- Automate workflow
- Notify people when actions are required
- Provide visibility (navigation & search & query)

Status

- Collaborating with multiple JBoss.org projects
  - most work under Guvnor
  - communities will eventually own their own extensions
- Working with the Drools community
  - enabling use of ModeShape in place of Jackrabbit
  - extract the general parts of Guvnor
  - keep Guvnor working for Drools
- S-RAMP
  - Working with OASIS S-RAMP TC
  - Working on an implementation
- WebDAV client and server libraries

Thanks for listening!

Questions?