SUMMIT JBoss WORLD

PRESENTED BY RED HAT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

redhat.

JBoss Enterprise Middleware & Big Data

Justin Hayes
Senior Architect, Red Hat Consulting
06.28.12

OVERVIEW – context

- **Premise:** Big Data Technologies Becoming Commoditized
    - But not what people are doing with the technologies
    - How you integrate, adopt, and build solutions is key
    - Leverage middleware
- **Goal:** Explore Intersection Between JBoss Enterprise Middleware and Big Data
    - Extensible/customizable reference architecture
    - Solution; not a product
    - Platform to build-your-own solution; not off-the-shelf
    - Avenue to improve JBoss projects

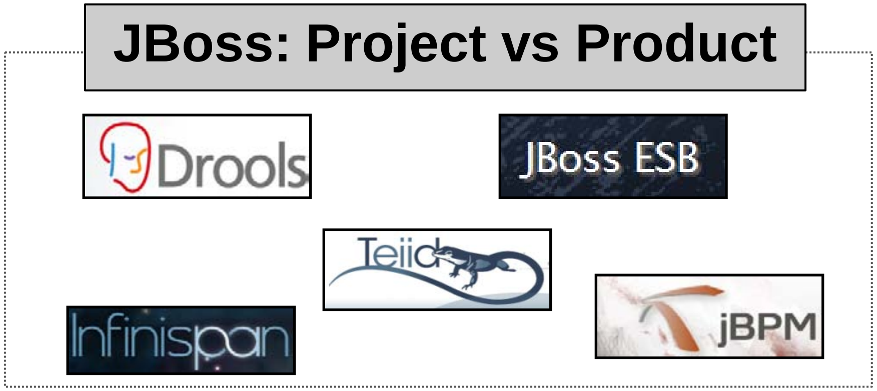
OVERVIEW – context

- Some Middleware-Like Tools in Big Data Ecosystem
  - **Oozie** – workflows for Hadoop jobs; incubator
    - *JBoss equivalent: Java Business Process Manager (jBPM)*
  - **Sqoop** – data transfer between Hadoop and structured data sources
    - *JBoss equivalent : Service Oriented Architecture Platform (SOA-P)*
  - **NoSQL** – key-value, document-oriented DB; not relational; scalable
    - *JBoss equivalent : JBoss Data Grid (JDG)*
  - **PIG** – can be used for data intake, ETL
    - *JBoss equivalent: SOA-P for intake pipeline, with transformation*
- JBoss More Extensive, Mature, and Standards-Based

OVERVIEW – summary

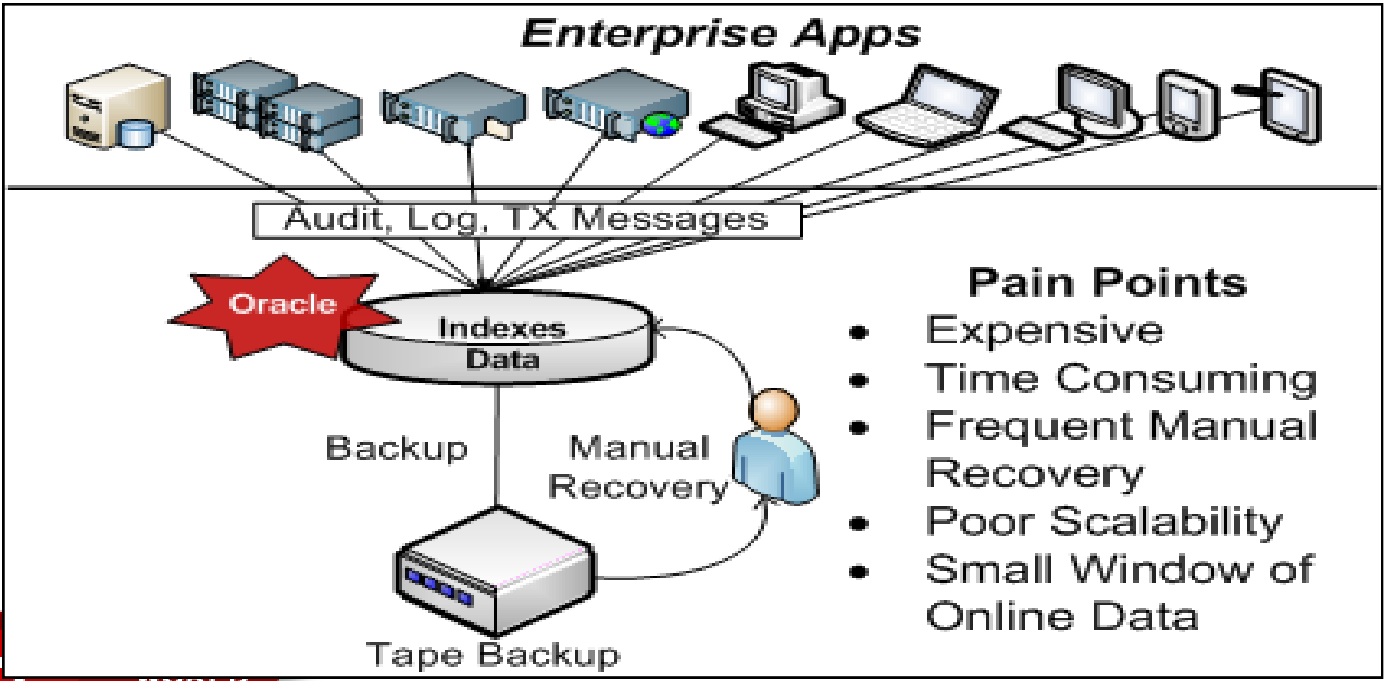
- **JBoss Middleware**: Integrate Technologies and Build Solutions
  - Big Data Just Another Thing to Integrate
  - Standards & Openness Important
- What is Tusk?
  - ***JBoss Reference Architecture Suitable for Addressing Big Data Integration Use Cases***
- What this Means to You:
  - Reference Implementation
  - Fodder for Brainstorming
  - Steal Code

JBoss: Project vs Product



USE CASE – pain points

- POC for Large Health Insurance Company
  - Enterprise apps log TB of data to Oracle
  - Need to swap out Oracle and expensive/laborious process with more scalable, cost effective one
    - Minimal impact on existing apps – technically or semantically



Pain Points

- Expensive
- Time Consuming
- Frequent Manual Recovery
- Poor Scalability
- Small Window of Online Data

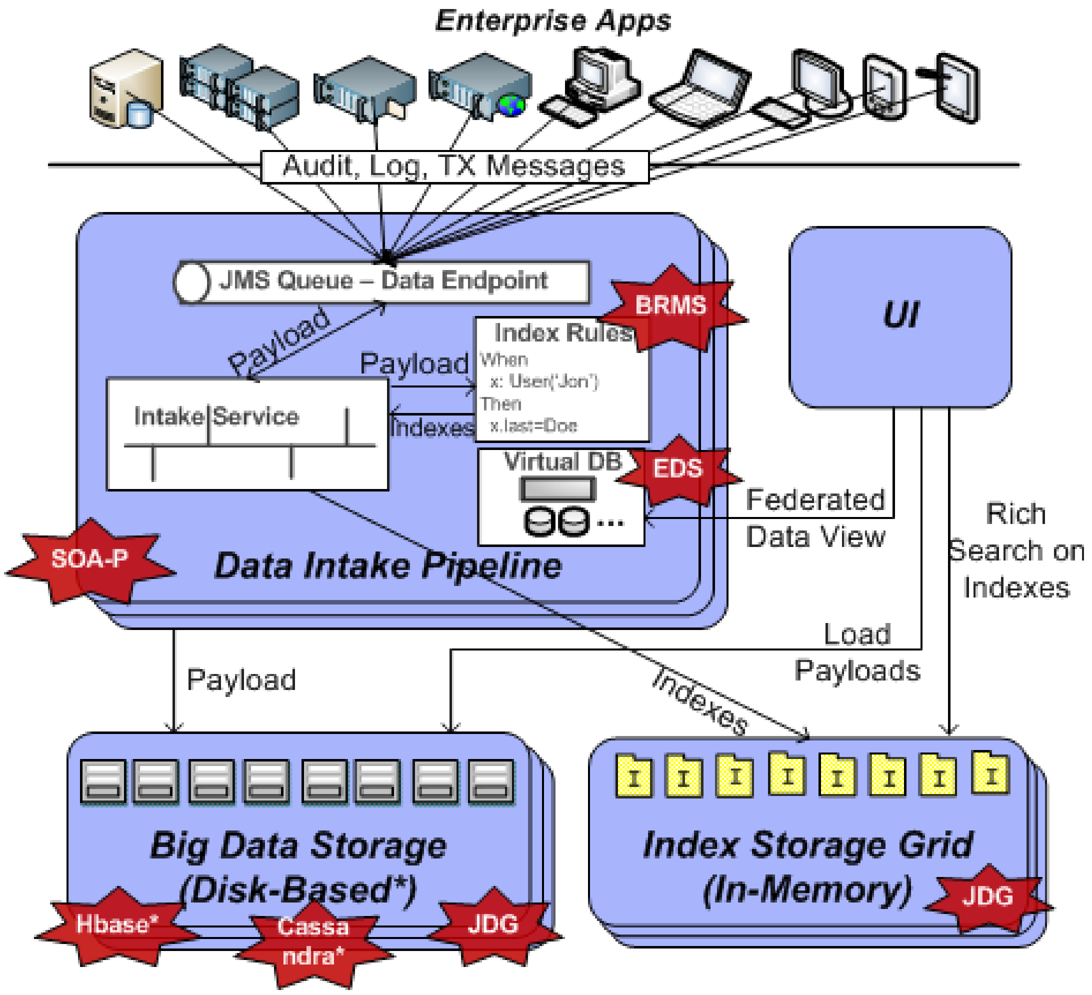
USE CASE – requirements

- Primarily Storage/Search/Retrieval
  - Interested in Hadoop and Cassandra
- Analytics in Future
- Did Not Need a Big Data Product
  - Needed a solution
- Represents Canonical Use Case
  - RH created a solution POC for this…
  - … and is turning it into a reference architecture (**Tusk**)
    - Useful for other use cases as well
    - Customizable, extensible, standards-based, open
    - Platform to build Big Data solutions

USE CASE – business value

- **Cost Savings**
  - More cost effective infrastructure for managing data
  - Reduced operating costs – fewer manual processes
- **Greater Data Visibility**
  - Larger window of online data
  - Enables More Effective Decisions
  - Enables big data analytics
  - Expose big data to the rest of the enterprise architecture

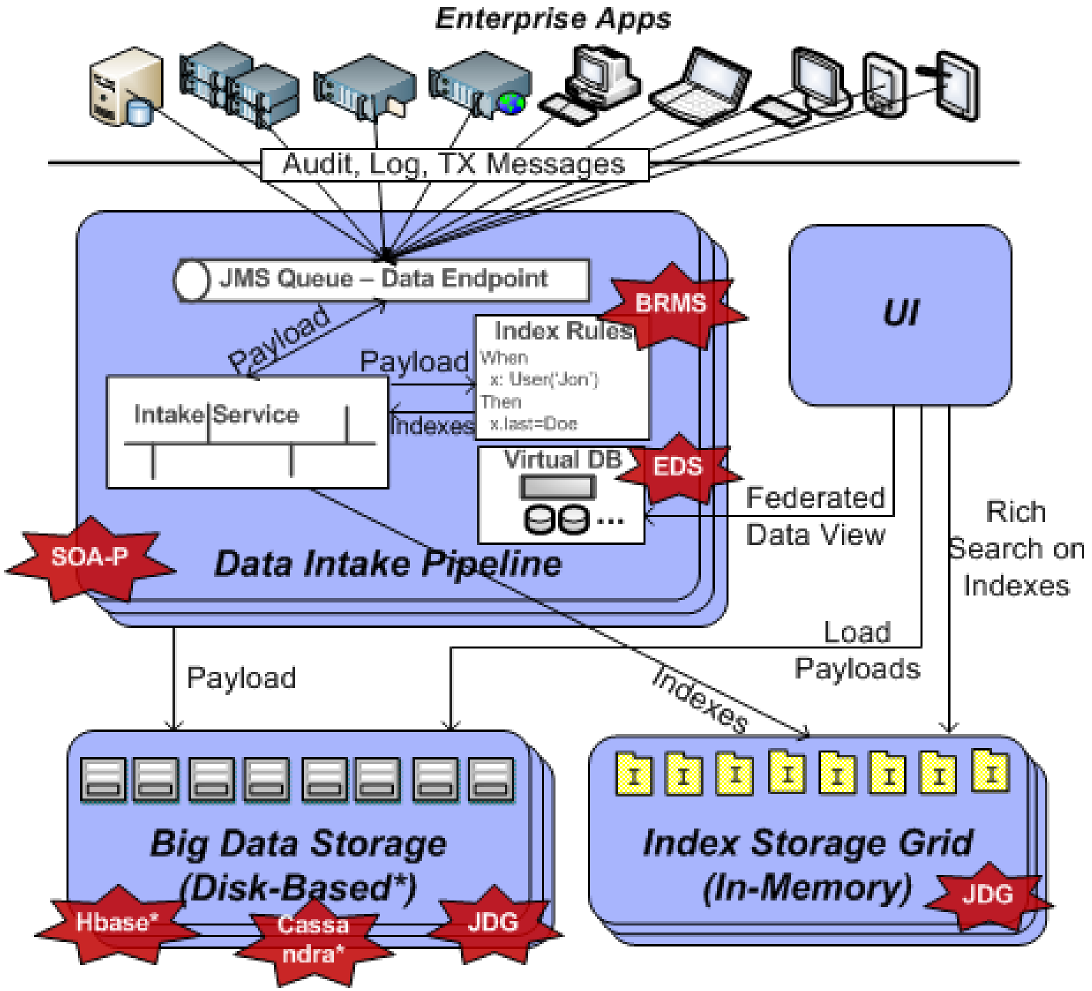
ARCHITECTURE



JBoss

- **Service Oriented Architecture Platform (SOA-P)**
  - Service Orchestration
  - Enterprise Integration Patterns
  - Many Listeners (JMS, FTP, SOAP, …)
  - Service Repository
- **Business Rules Management System (BRMS)**
  - Guided Rule Editor
  - Rule Repository
  - Complex Event Processing
- **JBoss Data Grid (JDG)**
  - Memory-Based NoSQL Data Grid
  - Scalable, Redundant, Fault Tolerant
  - Rich Querying
- **Enterprise Data Services (EDS)**
  - Data Federation
  - Virtual Databases

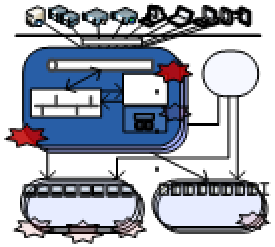
ARCHITECTURE



Big Data

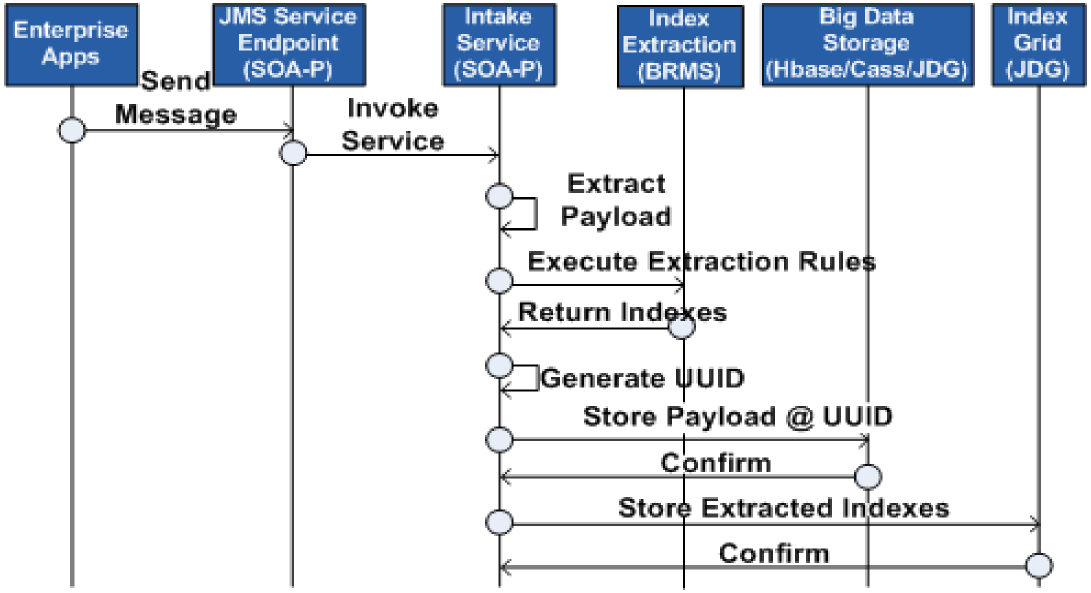
- **Apache HBase**
  - Hadoop Database
  - Random, Real-Time, Read/Write Access to Big Data
  - Distributed, Versioned, Column-Oriented Store
  - Modeled after Google's BigTable
- **Apache Cassandra**
  - Scalable, Highly Available, Fault Tolerant Database
  - Replication Across Data Centers
  - Column Family Data Model for Column Indexes
  - Performance of Log-Structured Updates
  - Support for Materialized Views
- **JBoss Data Grid (JDG)**
  - Can be used for data storage layer too
  - Disk-based cache store

ARCHITECTURE – data intake

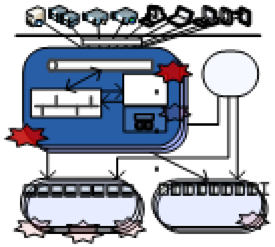


- **Data Intake Pipeline via SOA-P**
  - JMS Endpoint
    - Could be SOAP, FTP, file system, socket, custom via API
  - ESB Intake Service Drives Intake Pipeline
  - Extensible – Can Plug in Other Steps
    - Transformation
    - Audit wiretap
  - Made for Integrating

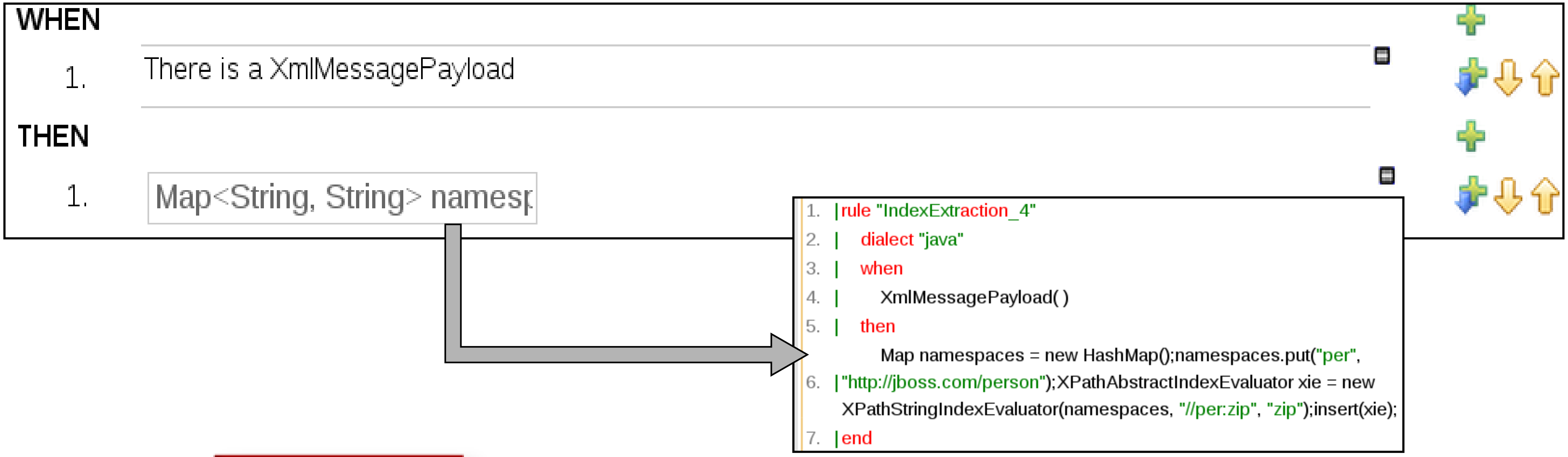
Data Intake Sequence



ARCHITECTURE – data intake



- **Custom Index Extraction via BRMS**
  - Extract Indexes from XML Data Payload
  - Uses Xpath
  - On-the-Fly Rule Editing to Change Index Rules
    - If need to reindex, already have intake pipeline



WHEN

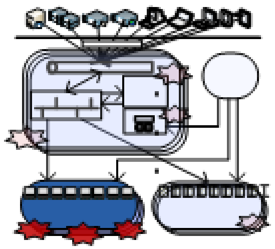
1. There is a XmlMessagePayload

THEN

1. Map<String, String> namesp

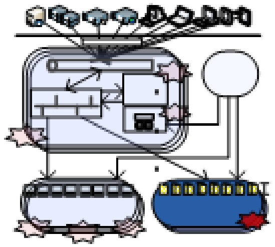
```
rule "IndexExtraction_4"
  dialect "java"
  when
    XmlMessagePayload( )
  then
    Map namespaces = new HashMap();namespaces.put("per",
"http://jboss.com/person");XPathAbstractIndexEvaluator xie = new
XPathStringIndexEvaluator(namespaces, "//per:zip", "zip");insert(xie);
end
```

ARCHITECTURE – data intake



- **Payload Storage via Big Data Technology**
  - Uses Big Data Technologies' APIs
  - ***HBase*** – Custom façade written on top of HBase API
    - *Table:* messages; *Column Family:* data; *Field:* value
  - ***Cassandra*** – Hector API
    - *Keyspace:* TuskData; *Column Family:* Messages; *Columns:* body, timestamp
  - ***JDG*** – Infinispan API
    - java.util.Map → put/get
    - Arbitrary schema (NoSQL)
- Hibernate Object/Grid Mapper (OGM)
- Which data store to use?

ARCHITECTURE – data intake



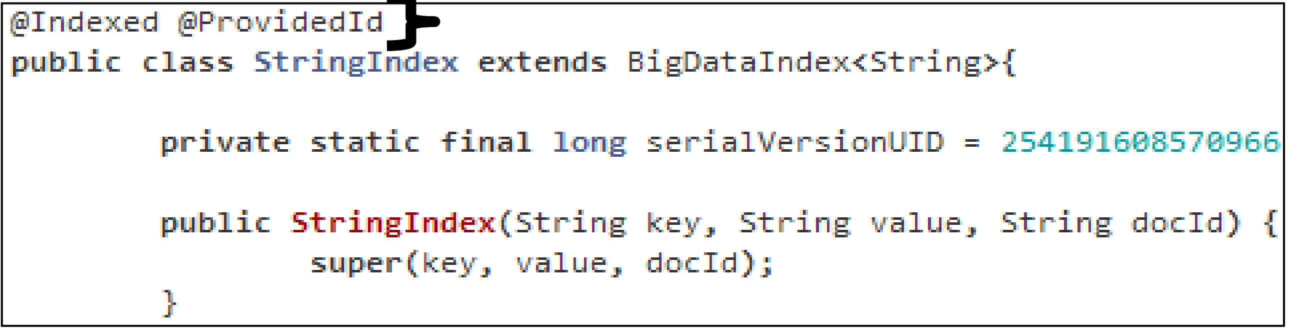
- Index Storage via JDG
    - Scales with Big Data Storage
        - Backed by the same data store
    - Store Indexes for Querying
    - Generic Index POJO: StringIndex

Example StringIndex

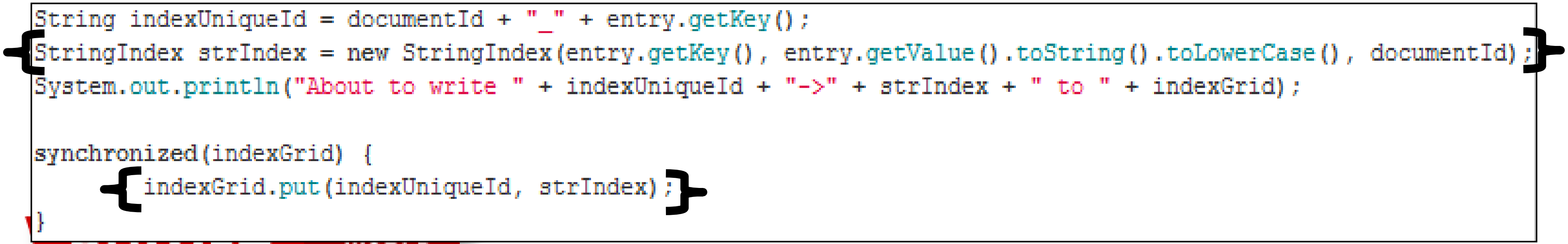
Key: zip

Value: 20009

DocId: 9483-2BA2-AE17…

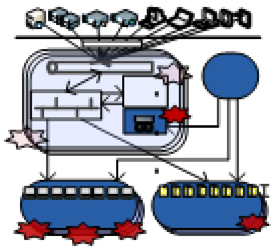


```
@Indexed @ProvidedId
public class StringIndex extends BigDataIndex<String>{

        private static final long serialVersionUID = 254191608570966

        public StringIndex(String key, String value, String docId) {
                super(key, value, docId);
        }
```



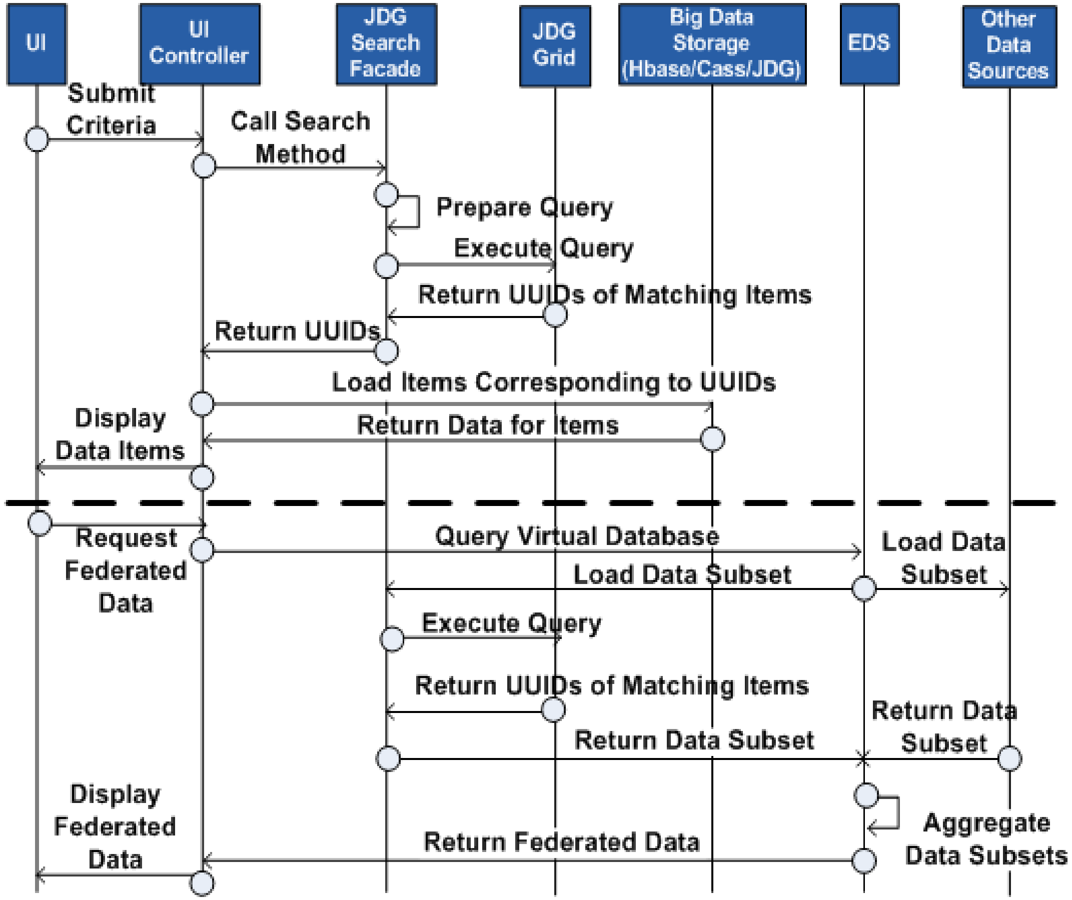
```
String indexUniqueId = documentId + "_" + entry.getKey();
StringIndex strIndex = new StringIndex(entry.getKey(), entry.getValue().toString().toLowerCase(), documentId);
System.out.println("About to write " + indexUniqueId + "->" + strIndex + " to " + indexGrid);

synchronized(indexGrid) {
    indexGrid.put(indexUniqueId, strIndex);
}
```

ARCHITECTURE – search and retrieval

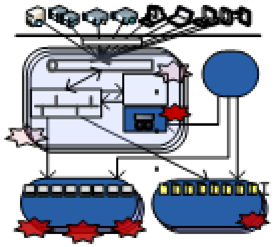


- Spring MVC WebApp
- Search Fields Match Index Extraction Rules
- Search Returns UUID of Matching Payloads
- Load Payloads w/ UUIDs
- Uses EDS for Combined View of Big Data Assets and Conventional Data Sources

Search & Retrieval Sequence



ARCHITECTURE – search and retrieval



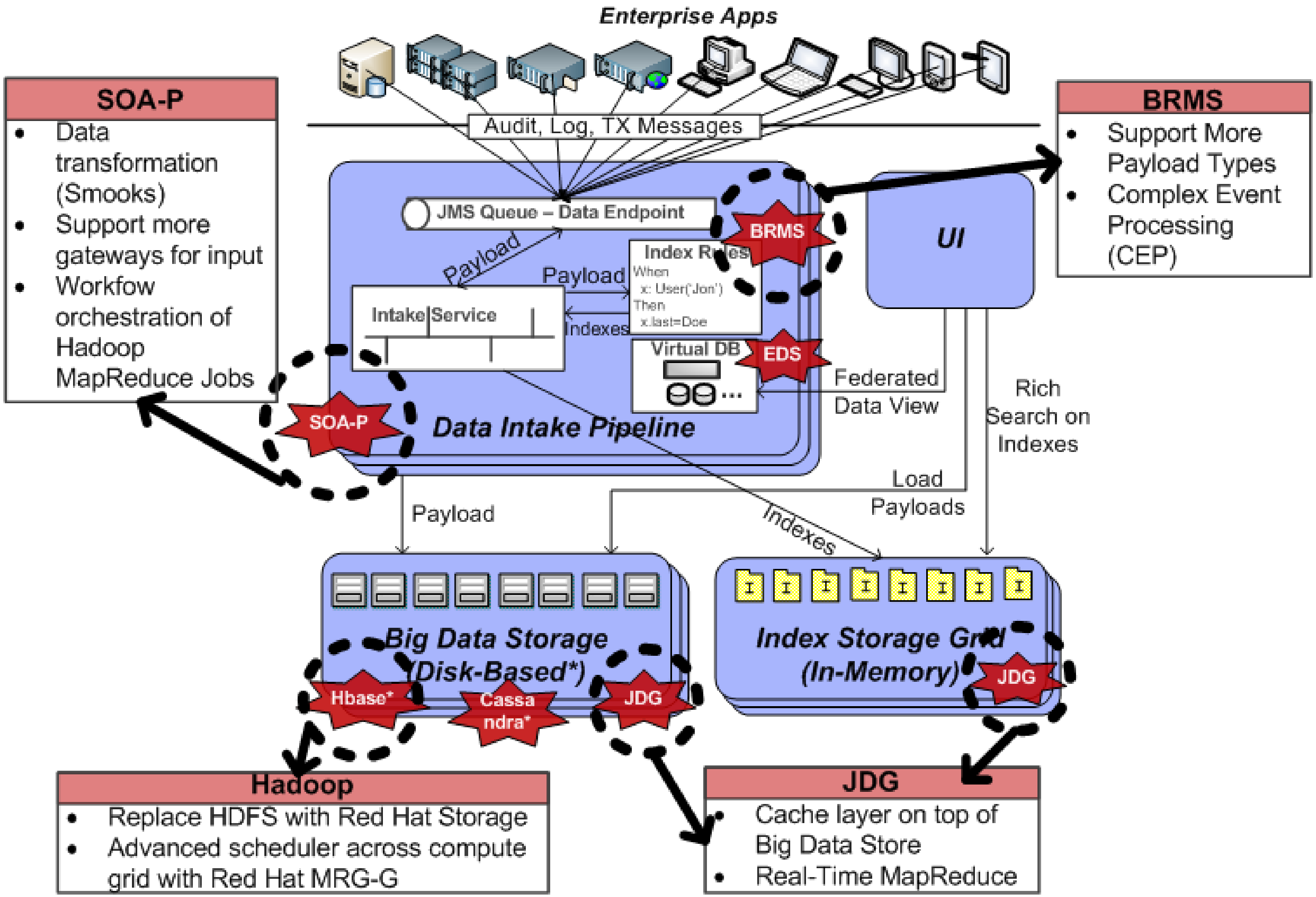
- Infinispan includes Scalable, Distributed Apache Lucene Directory Implementation
  - Fast in-memory search < slow disk-based MapReduce
  - Stores indexes in cluster-wide shared memory
- Hibernate Search with Criteria Style Queries
  - Each criterion → BooleanJunction
    - *Key* (index field name)
    - *Value* (target value)
  - 'AND' all criteria junctions together to get main query

Query Example

((**key**=zip and **value**=20009*)
and
(**key**=name and **value**=just*))

- ***Caveats** – not cluster friendly yet; still performance testing this feature; not in initial JDG release (Infinispan only)*

TUSK'S FUTURE – customize & extend



SOA-P

- Data transformation (Smooks)
- Support more gateways for input
- Workfow orchestration of Hadoop MapReduce Jobs

BRMS

- Support More Payload Types
- Complex Event Processing (CEP)

Hadoop

- Replace HDFS with Red Hat Storage
- Advanced scheduler across compute grid with Red Hat MRG-G

JDG

- Cache layer on top of Big Data Store
- Real-Time MapReduce

And be Sure to Check These Out…

- **NoSQL & Big Data at Red Hat**
  - Thu @ 4:50, Room 207

- **Large Scale / Big Data Federation & Virtualization: A Case Study**
  - Fri @ 11:00, Room 208

REFERENCE

- **Tusk Lead**
  - Justin Hayes – jhayes@redhat.com
- **Tusk Code**
  - https://github.com/jboss-tusk/tusk
- **JBoss Products**
  - http://www.redhat.com/products/jbossenterprisemiddleware/soa
  - http://www.redhat.com/products/jbossenterprisemiddleware/business-rules
  - http://www.redhat.com/promo/dg6beta
  - http://www.jboss.org/infinispan.html
  - http://www.redhat.com/products/jbossenterprisemiddleware/data-services

- QUESTIONS -

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey

SUMMIT JBoss WORLD

PRESENTED BY RED HAT