JBoss WORLD
CHICAGO 2009

FOLLOW US:

TWITTER.COM/REDHATSUMMIT

TWEET ABOUT US:

ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END OF YOUR EVENT-RELATED TWEET

presented by

JBoss WORLD CHICAGO 2009

Boris Lublinsky, Robert Camp
Using JBoss Enterprise Middleware in NAVTEQ
09/04/2009

NAVTEQ

presented by

About the Presenters

Boris Lublinsky

- Lead Architect, NAVTEQ
  - SOA, EA and MDA
  - BPM design and implementation
  - IT Architecture and Strategy
  - 25+ years experience in distributed systems, software and architecture
- Affiliations
  - InfoQ – SOA news editor
- Author
  - Applied SOA: Architecture and Design Strategies, Wiley, June 2008
  - Over 60 articles in Java Development Journal, WebServices Journal, JavaPro, Enterprise Architect, Business Integration, Developerworks, InfoQ, etc.

Robert Camp

- Lead Engineer, NAVTEQ
  - 5+ years of LBS development
  - Mobile Phone client development
  - 20+ years experience in distributed systems, real-time fault-tolerant software

Agenda

- Defining SOA
- JBoss Enterprise Middleware
- NAVTEQ Vertical Markets Solutions
  - The problem
  - Implementation
- NAVTEQ Map Maintenance Automation
  - The problem
  - Implementation
- Lessons learned
- For More Information
- Q/A

JBoss WORLD CHICAGO 2009

Defining SOA

NAVTEQ

presented by

What's in a Name?

- From the point of view of a business executive and business analyst, SOA is a set of services that constitute IT assets (capabilities) and can be used for building solutions and/or exposed to customers and partners.
- From the point of view of an enterprise architect ,SOA is set of architectural principles and patterns addressing the overall characteristics of solutions - modularity, encapsulation, loose coupling, separation of concerns, reuse, composability, etc.
- From the point of view of a project manager, SOA is a development approach supporting massive parallel development.
- From the point of view of a tester and/or quality assurance engineer, SOA represents a way to modularize and consequently simplify overall system testing.
- From the point of view of software developer, SOA is a programming model complete with standards, tools and technologies, for example, Web Services.

SOA Defined

SOA can be defined as an ***architectural style*** promoting the concept of ***business-aligned enterprise service*** as the fundamental unit of design and implementation and ***enterprise business processes*** for composing these services into ***enterprise business solutions***.

Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution (IEEE Standard 1471-2000).

An architectural style is a family of architectures related by common principles and attributes. An architectural style provides a useful set of reasonable alternatives – not all alternatives – and coordinates them to work well together (Richard Hubert. Convergent Architecture).

Today's Enterprise Architecture

- Application-centric
  - Design, development, enhancements and maintenance of software systems revolved around applications.
  - Each application is built for a single purpose with its own data store(s) and for a single set of users.
  - Creates of segregated silos within the enterprise architecture resulting in expensive and inflexible IT systems.
- Islands of data
  - Each application has its own meaning and/or definition of enterprise data
  - Overlapping information between application.
  - No application can provide a complete picture of the enterprise data.
- Islands of automation
  - Each application focuses on a limited set of activities within the enterprise.
  - There is duplication between business processes contained within different applications

Raising the Level of Abstraction

The fundamentals of engineering such as abstraction and separation of concerns never go out of style… there are real opportunities to raise the level of abstraction again (Grady Booch)

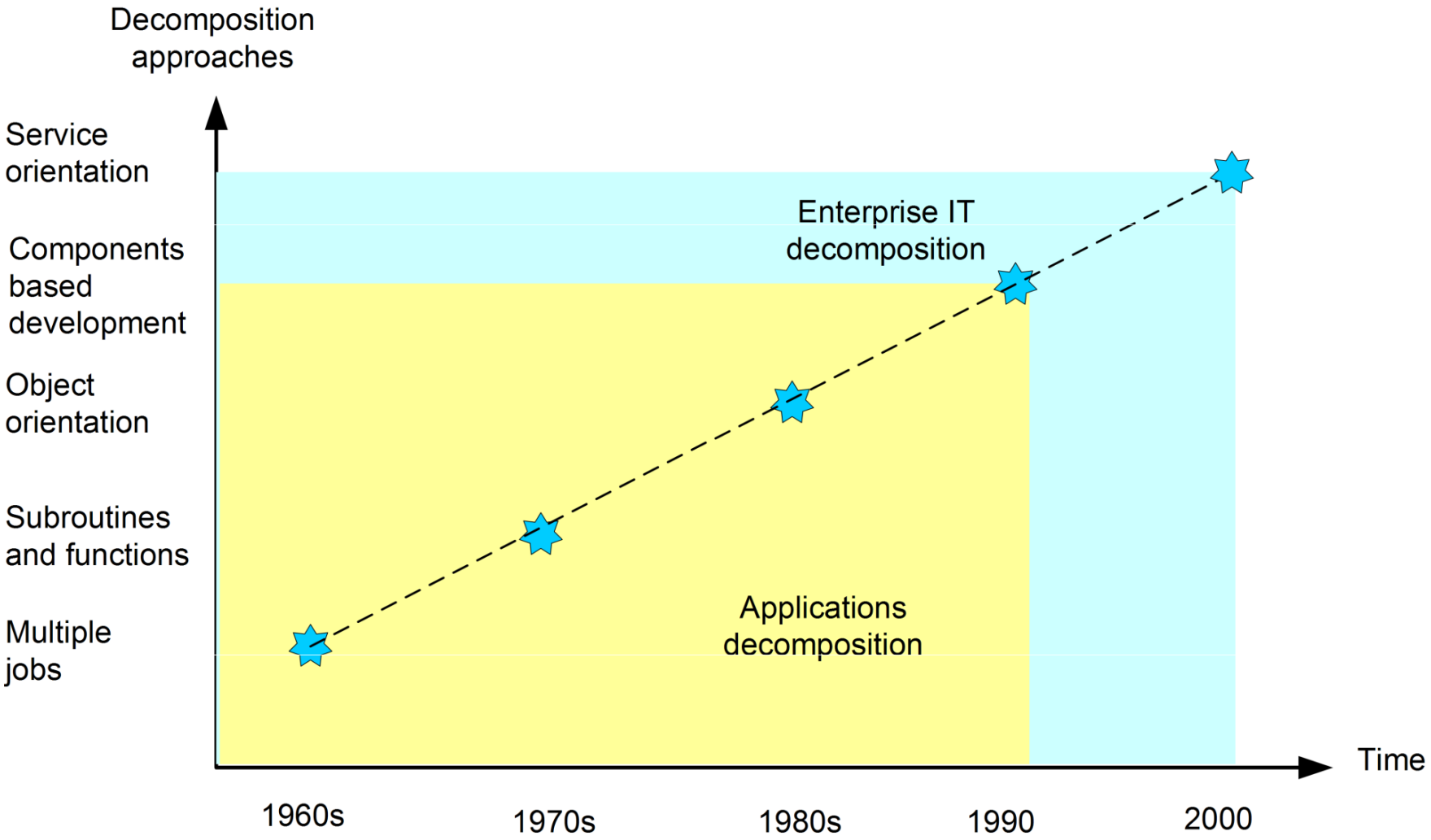
SOA introduces two higher level abstractions - enterprise business services and business processes:

- Enterprise business services represent existing IT capabilities (aligned with the business functionality of the enterprise).
- Business processes, which orchestrate business services, implement the overall functioning of the enterprise.

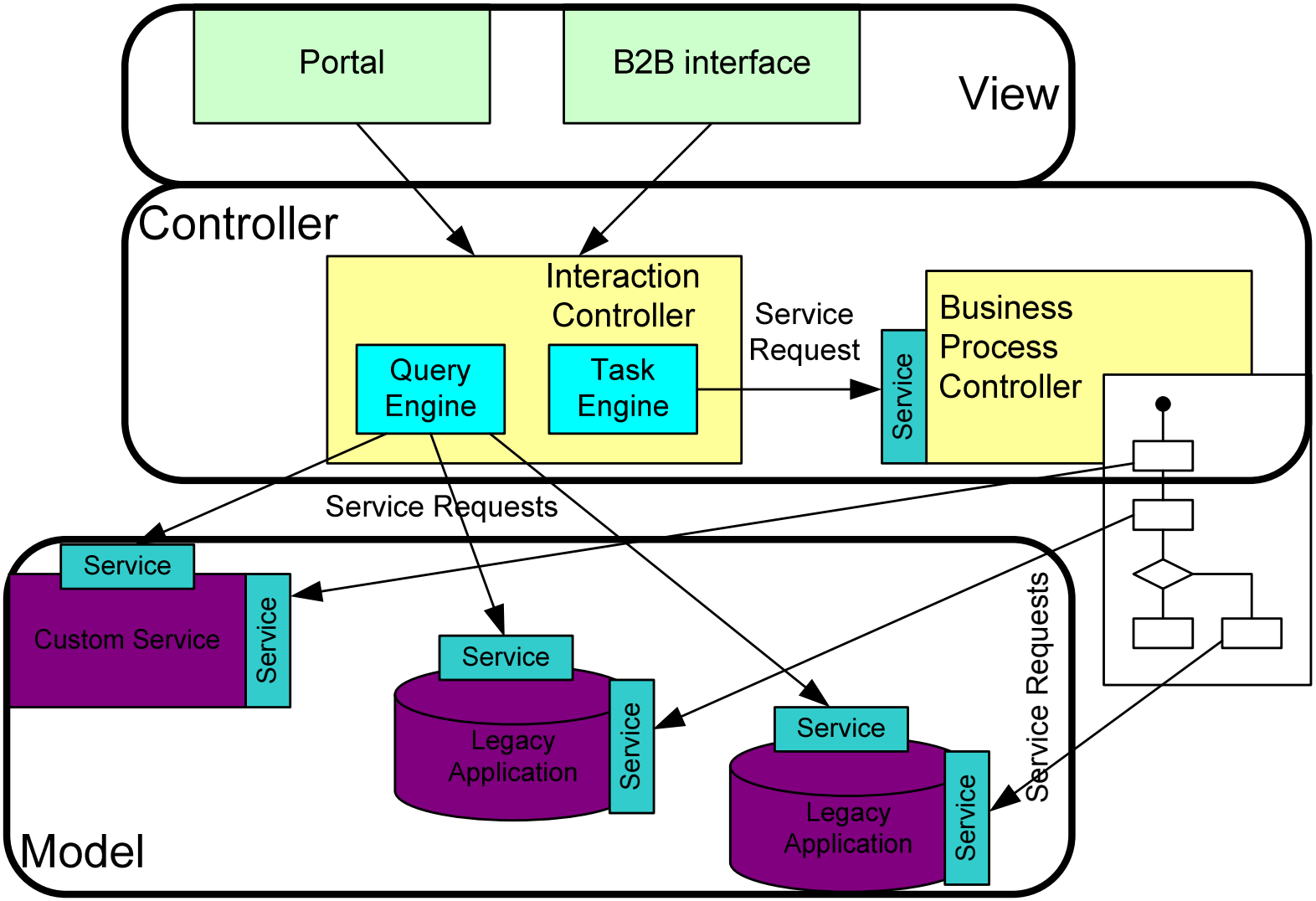
Architectural Impact

Characteristic	Application-centric architecture	SOA
Design and implementation	• Function-oriented • Build to last • Long development cycles	• Coordination oriented • Build to change • Build and deployed incrementally
Resulting system	• Application silos • Tightly coupled • Object oriented interactions	• Enterprise solutions • Loosely coupled • Semantic message oriented interactions

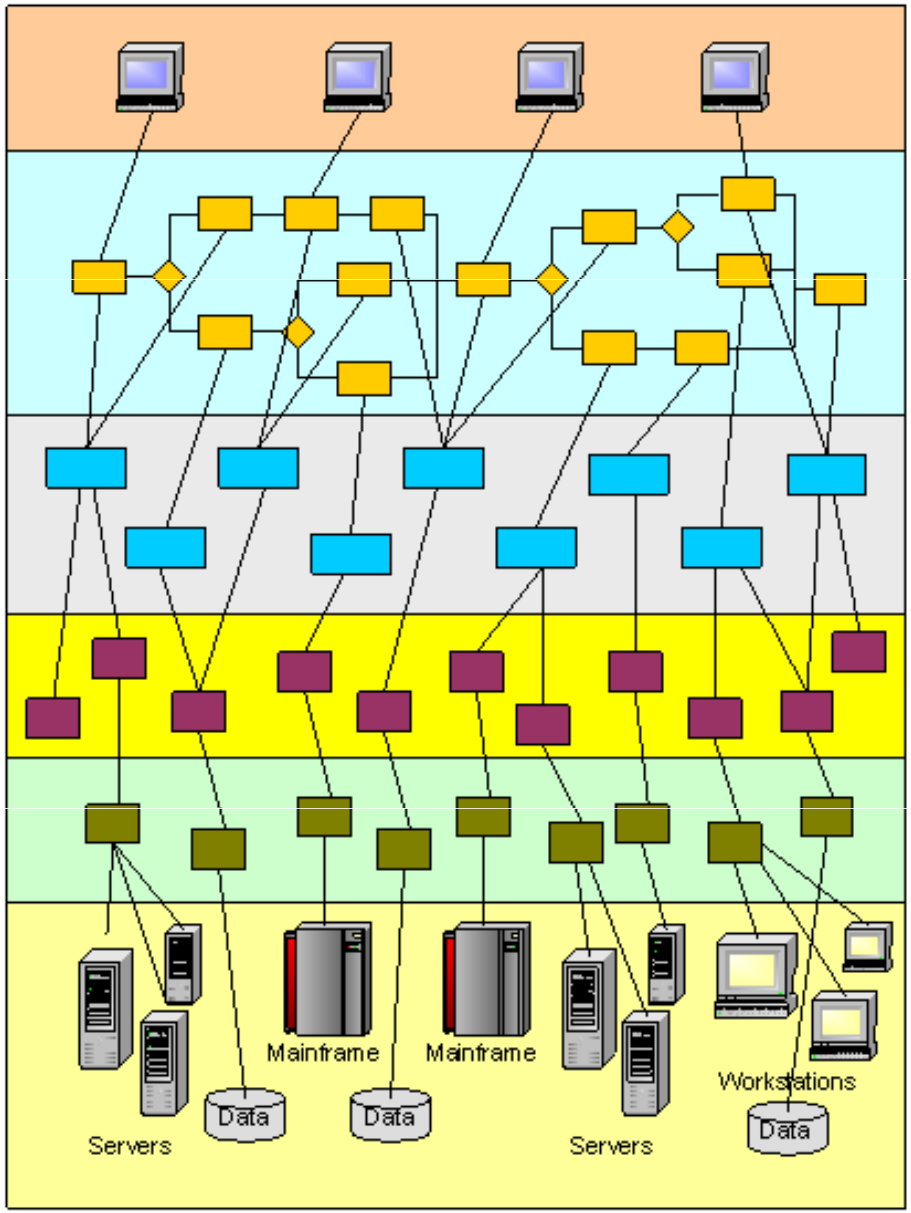
Decomposition is the Foundation



Conceptual Solutions Architecture



Layered Enterprise Architecture



Customer facing applications- view into enterprise processes

Business Process Driven Enterprise

Services - Rationalization of Enterprise Systems Against Business Meaningful Services

Business Components- Encapsulation of integration and additional functionality

Integration Access. Exposing Existing Enterprise Systems

Existing IT Hardware/Sofware Topology

JBoss WORLD CHICAGO 2009

JBoss Enterprise Middleware

NAVTEQ

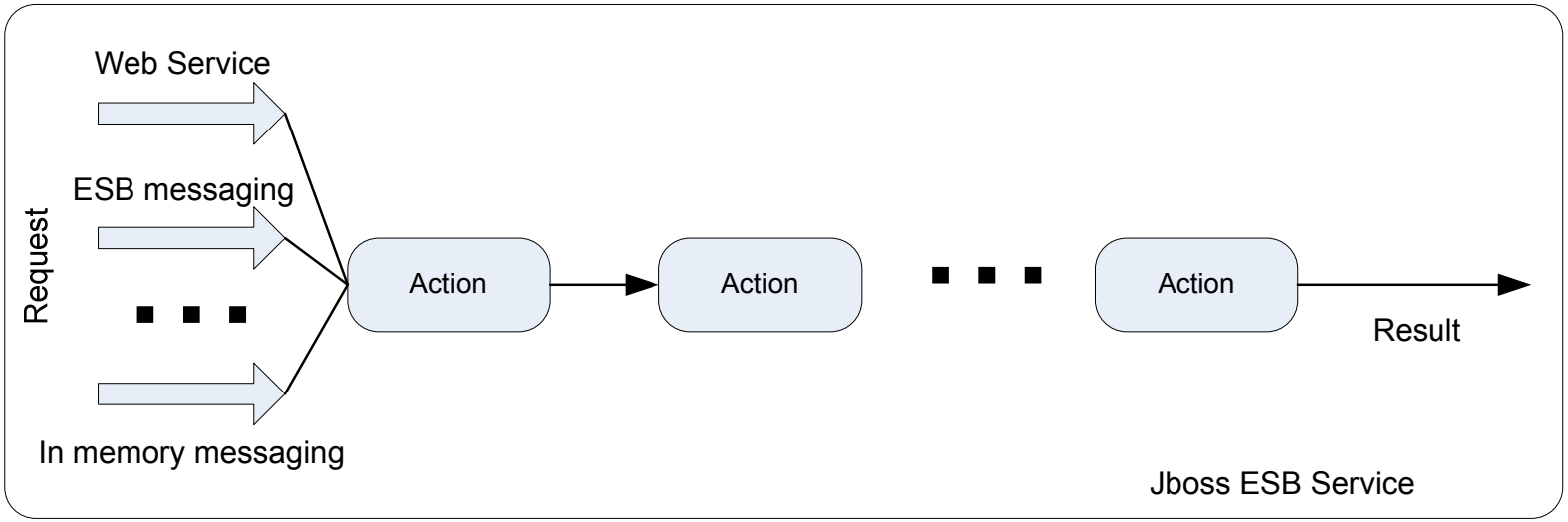
presented by

JBoss ESB

- Everything is a service
- ESB service != Web Service. No WSDL, no formal interface definition
- Single service – multiple endpoints - listeners
- Single method (doWork) services with a “standardized interface”

  Message output = service.doWork(Message input)
- Service invocation paradigm - messaging
- Message – similar to SOAP message - header, body, fault, attachments, etc. Build in WS-Addressing
- Body – map of serializable Java objects – not strongly typed
- Web services support – SOAPProcessor and SOAPClient

JBoss ESB - continued

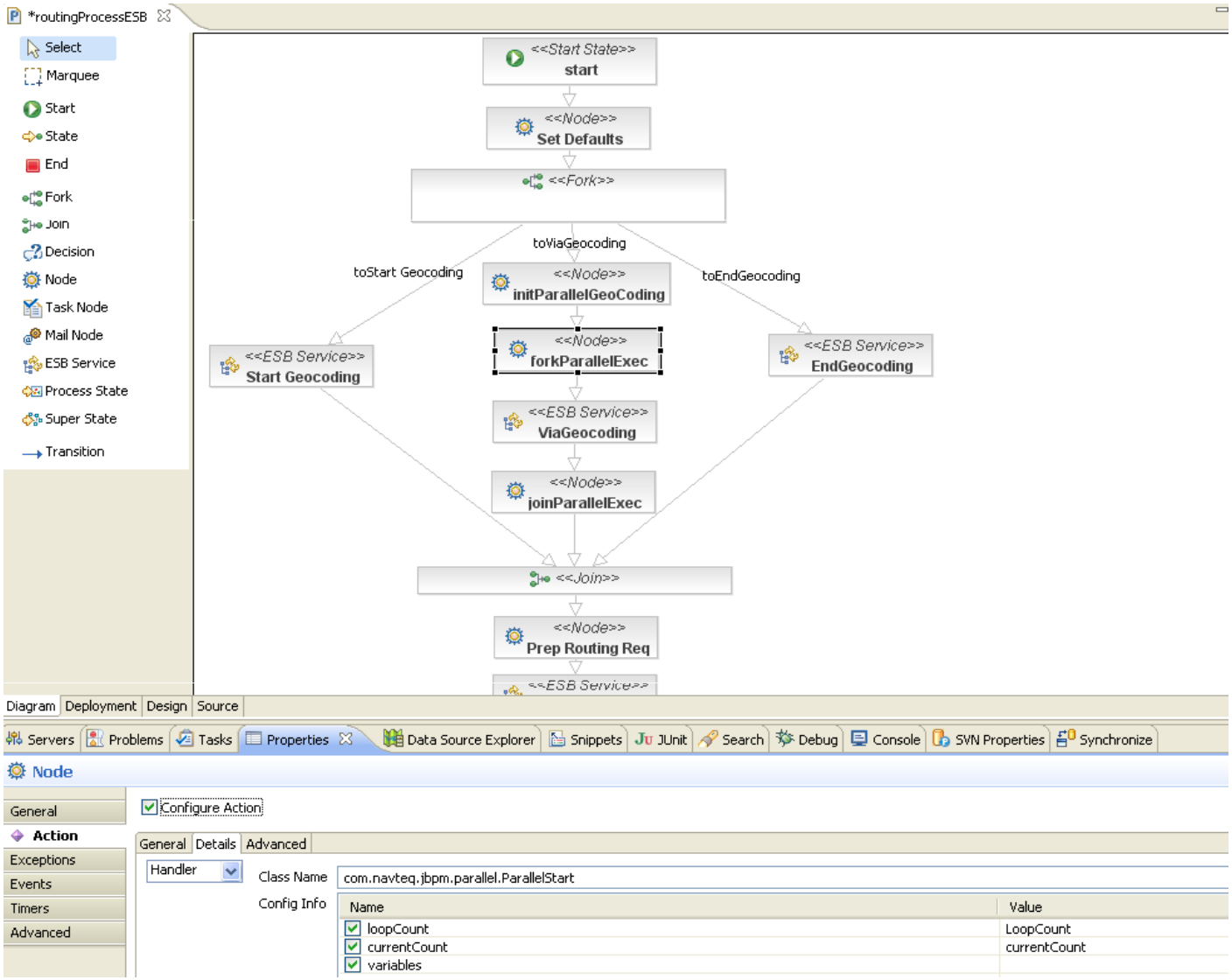


- Service is a component execution pipeline
- Service invocation is based on a message, delivered by transport.
- Transports local and remote
- MEPs (Message Exchange patterns) – asynchronous/synchronous
- Service invocation is based on build in registry
- ***ServiceInvoker*** class hides invocation complexity

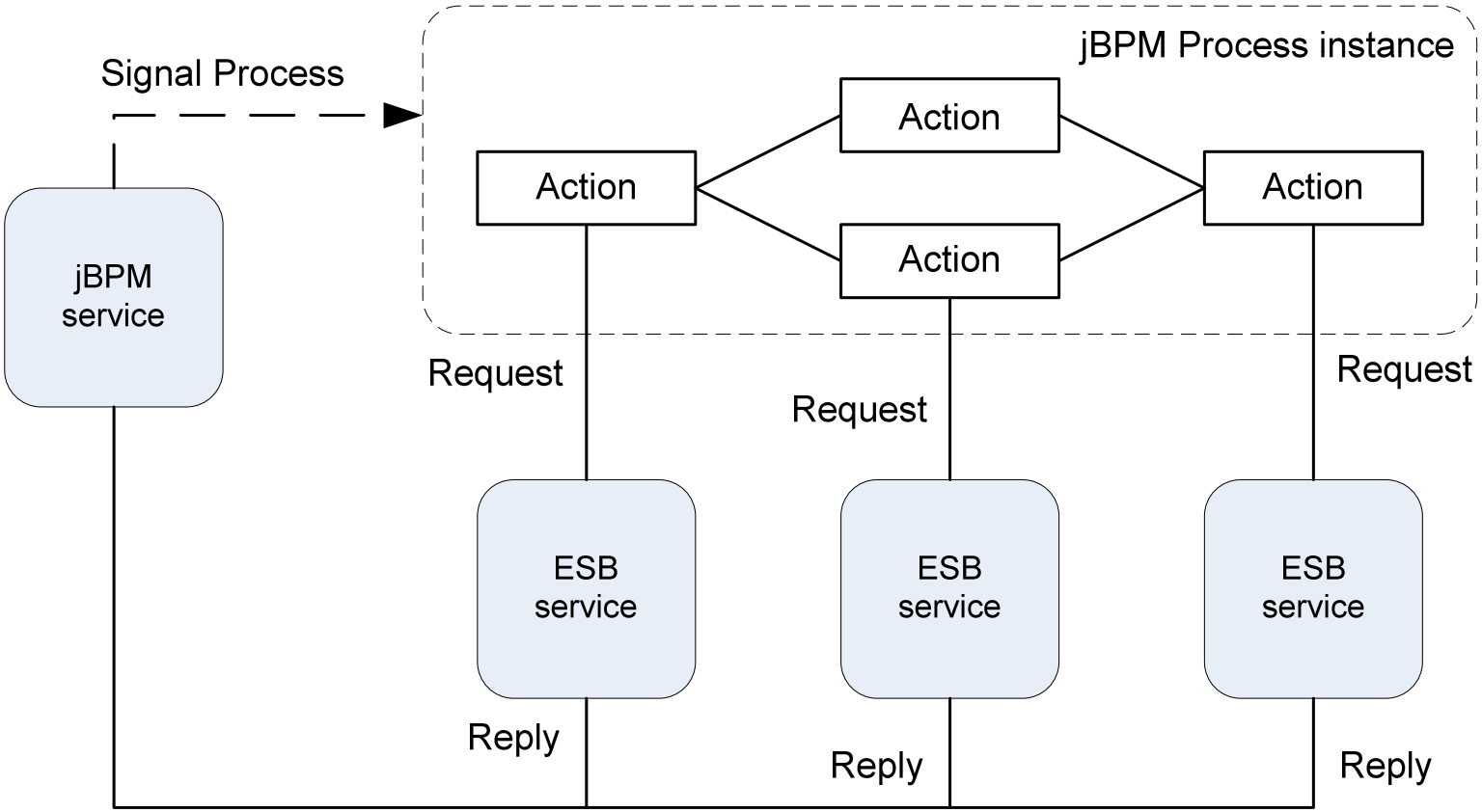
Smooks

- Generate an Event Stream from any data source (XML, Java, EDI, etc) and applies the Visitor pattern to process it
- Does not require any if-then logic, if an element does not exist in the source, the event is not produced and mapping will not occur.
- Naturally supports componentization of mapping.
- Improves maintenance through externalization of transformation definitions.
- Minimize transformation overhead
- JBoss/Smooks provides (graphical) tooling for designing/debugging of transformations

jBPM


*routingProcessESB
Select
Marquee
Start
State
End
Fork
Join
Decision
Node
Task Node
Mail Node
ESB Service
Process State
Super State
Transition
<<Start State>>
start
<<Node>>
Set Defaults
<<Fork>>
toViaGeocoding
toStart Geocoding
toEndGeocoding
<<Node>>
initParallelGeoCoding
<<ESB Service>>
Start Geocoding
<<Node>>
forkParallelExec
<<ESB Service>>
EndGeocoding
<<ESB Service>>
ViaGeocoding
<<Node>>
joinParallelExec
<<Join>>
<<Node>>
Prep Routing Req
Diagram
Deployment
Design
Source
Servers
Problems
Tasks
Properties
Data Source Explorer
Snippets
JUnit
Search
Debug
Console
SVN Properties
Synchronize
Node
General
Action
Exceptions
Events
Timers
Advanced
Configure Action
General
Details
Advanced
Handler
Class Name
com.navteq.jbpm.parallel.ParallelStart
Config Info
Name
Value
loopCount
LoopCount
currentCount
currentCount
variables

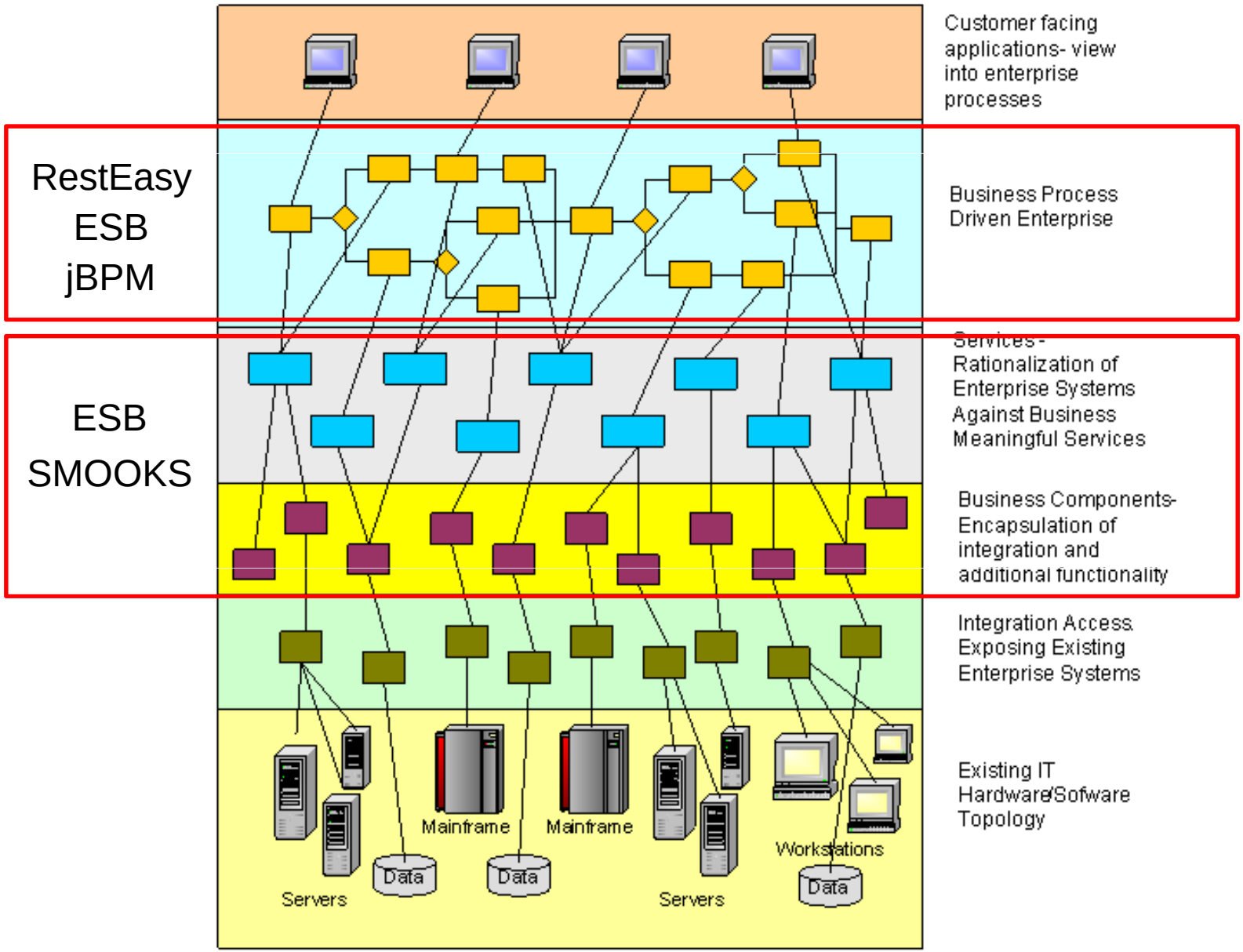
JBoss ESB/jBPM integration



RestEasy

- Fully certified JAX-RS implementation
- Portable to any Web container
- JAXB marshalling support for XML, JSON, Jackson, Fastinfoset, and Atom as well as wrappers for maps, arrays, lists, and sets of JAXB Objects.
- Support for Query and Path parameters
- Client library, simplifying client implementation
- The ability to use the same interface for both client and server implementation.
- Asynchronous Job Service

Layered Enterprise Architecture with JBoss Enterprise Middleware



JBoss WORLD CHICAGO 2009

NAVTEQ Vertical Markets Solutions

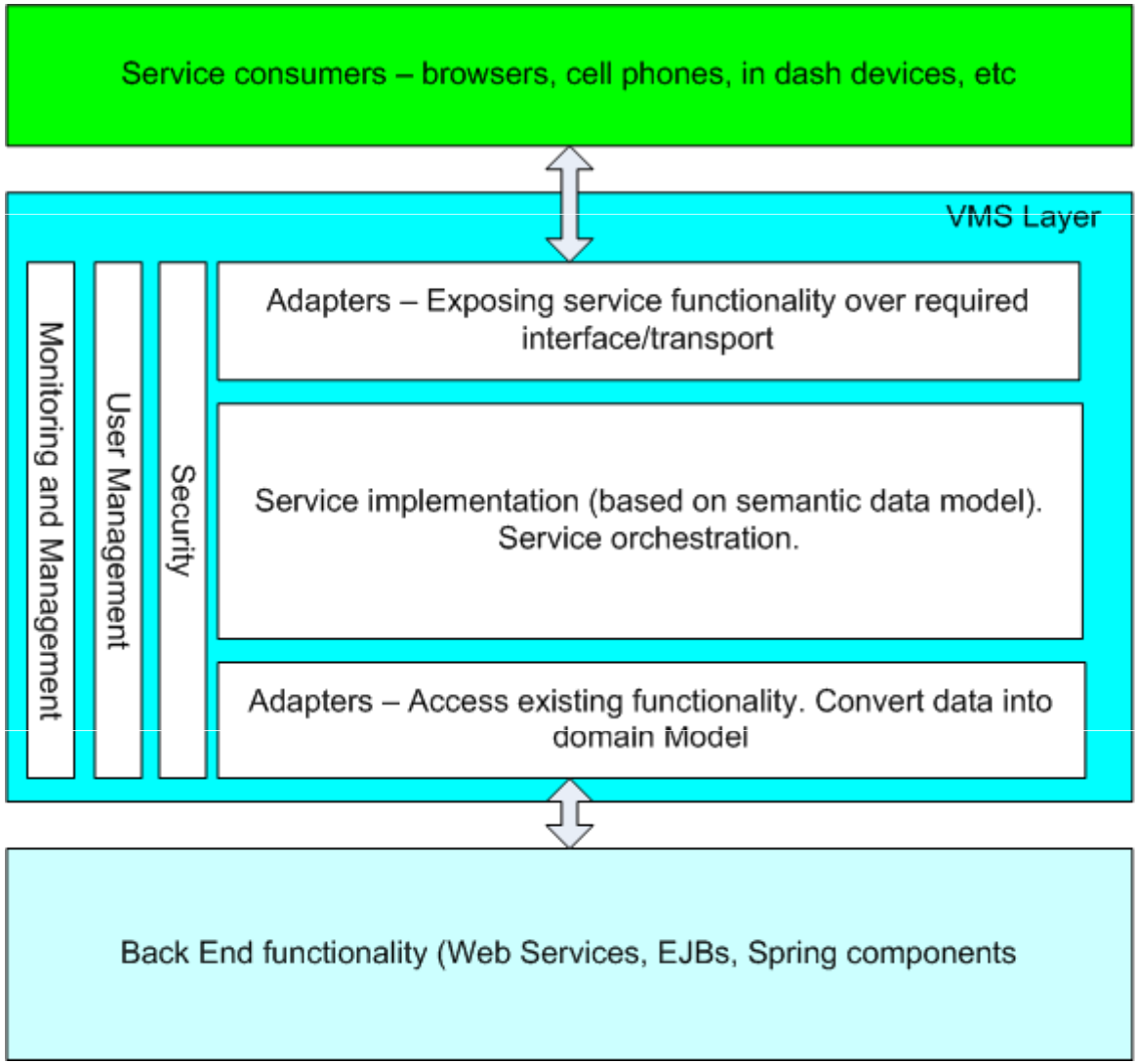
NAVTEQ

presented by

Business problem

- Create meaningful location-based business functions
- Integrate various disparate services
  - Assets obtained by numerous acquisitions
  - Services developed for specific opportunities
  - Inconsistent data models between services
- Provide a uniform endpoint pattern
- Employ uniform security mechanism
- Support of wide array of client platforms
- Simplify partner integration
- Meet customers' demand for quick solutions

Overall Solution Architecture



JBossWS, RestEasy

JBossESB, jBPM, Smooks, Hibernate

JBossWS, RestEasy, Smooks

Overall solution Architecture

- Back end adapters layer responsibilities
  - Access to existing back-end (or/and 3rd party) functionality
  - Return results of execution (in the form of standard domain model) into a platform (RestEasy,JBossWS, Smooks)
- Service implementation layer responsibilities
  - Data transformation between multiple existing services and conform to semanitc data model (Smooks)
  - Orchestration of existing services (accessible through back-end adaptors) into composite services required by customers. (JBossESB, jBPM)
  - Custom functionality that does not currently exist can be implemented in this layer. (JBossESB, Hibernate, etc.)
- Consumer adapters layer responsibilities
  - Deliver results of service execution to different types of clients over different protocols/transports. (RestEasy,JBossWS)

Core Services and Data Models

Services

- Get Map
- Geo Code Text Address
- Proximity Search
- Calculate Route
- Get Route by ID
- Get Friends
- Send SNS Message
- Get Ads
- Get User Profile
- CDS Fuel Search
- CDS Movie Search
- Get Purchased Map Products
- Get Latest Product for Location

Data

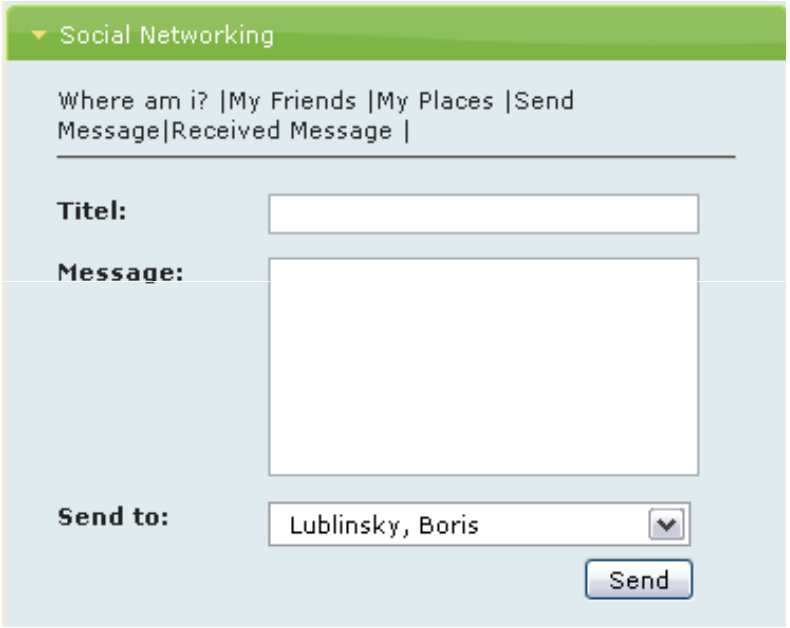
- User
- Location
- Route
- Map Product

Examples of Orchestrated LBS Services

- Send a map of my location to friends within 5 miles of my current location
- Display ads for businesses along route
- Find cheapest gas along route at intervals corresponding to fuel range of vehicle
- Find route to movie theater playing “Half-Blood Prince” within driving distance of next start time.
- Check for map updates along a route.

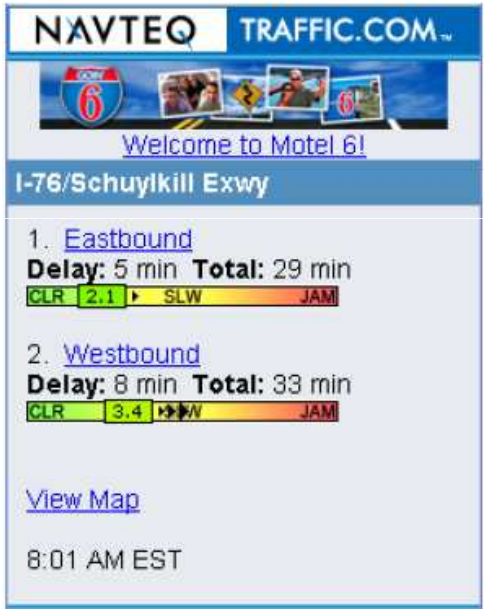
Social Networking

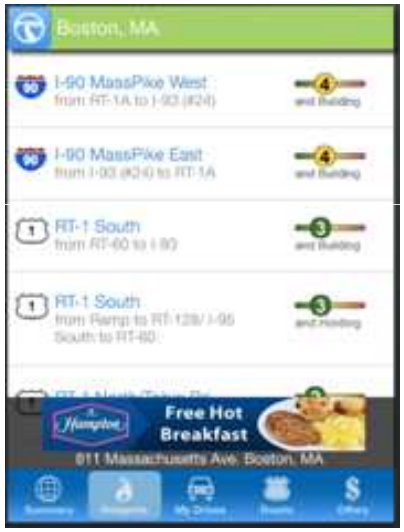
- Send map to friends
- Map friends
- Location Based Messaging
- Update My Location

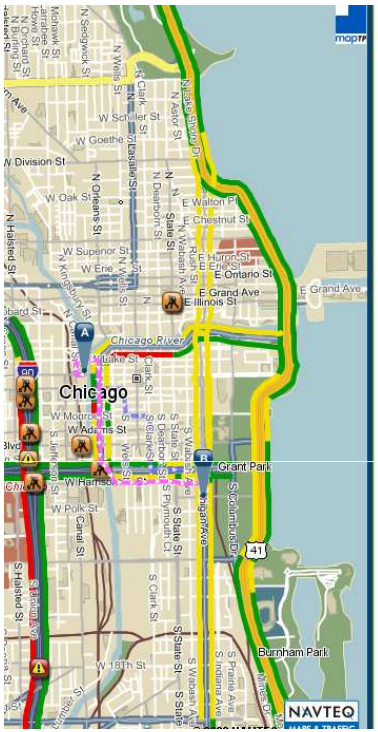


Advertising

- Ads along route
- Banner Ads
- Location Based Ads
- Click-to-call
- User Action Reporting







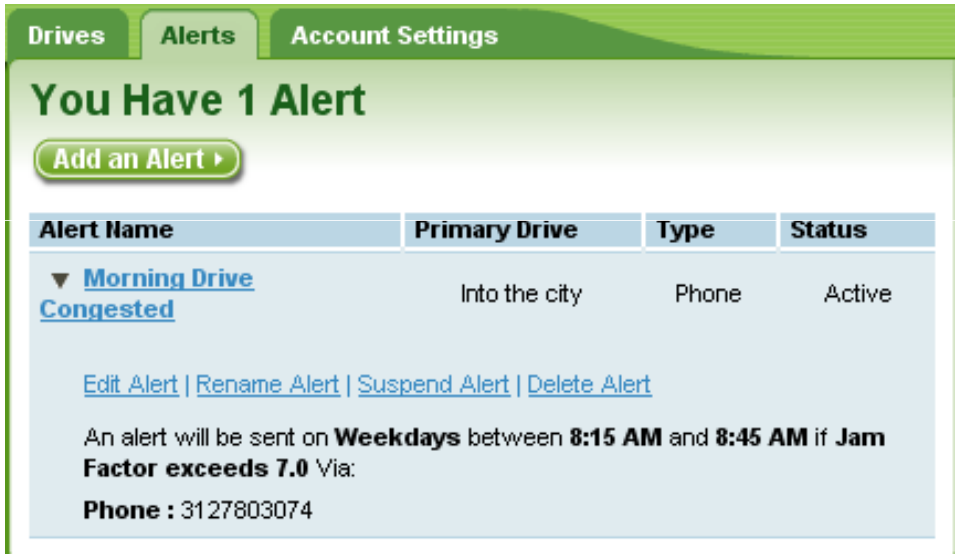
3	Turn right onto W Lake St.	0.1 mi
4	Turn right onto N Upper Wacker Dr, N Wacker Dr.	0.3 mi
5	Head straight on S Upper Wacker Dr, S Wacker Dr.	0.4 mi
6	Head straight on S Wacker Dr.	0.1 mi
7	Turn left onto W Harrison St.	0.4 mi
8	Turn right onto S State St.	0.0 mi
9	Turn left onto E Harrison St.	0.2 mi
10	Turn right onto S Michigan Ave.	0.1 mi
11	You have arrived at your destination S Michigan Ave.	0.0 mi

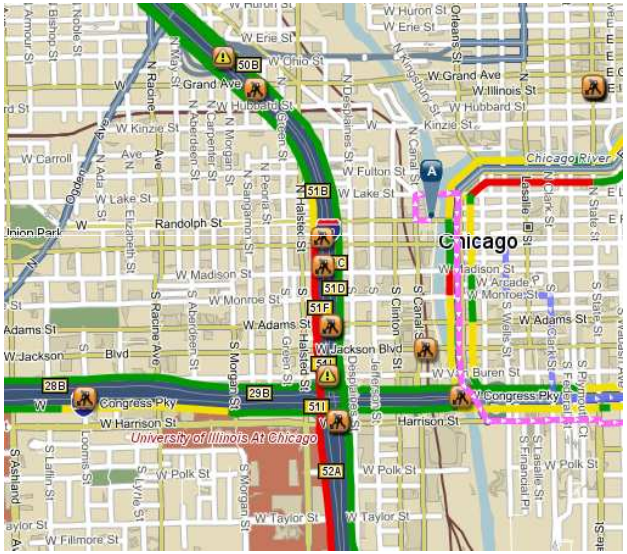
B 720 S Michigan, Chicago, IL

JBoss WORLD CHICAGO 2009

Commute Management

- Fuel prices
- Drive Time Comparison
- Incident Reports
- Flow visualization
- Traffic Alerts

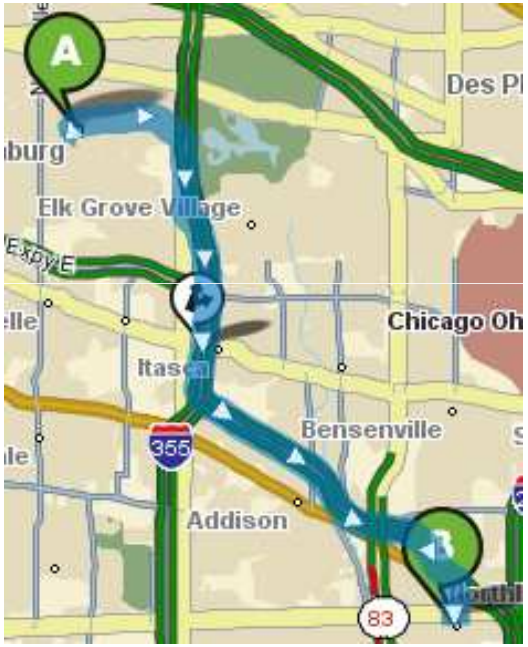




Trip Planning

- Event Searches
- Routing
- Directions
- POI Searches
- Junction Views

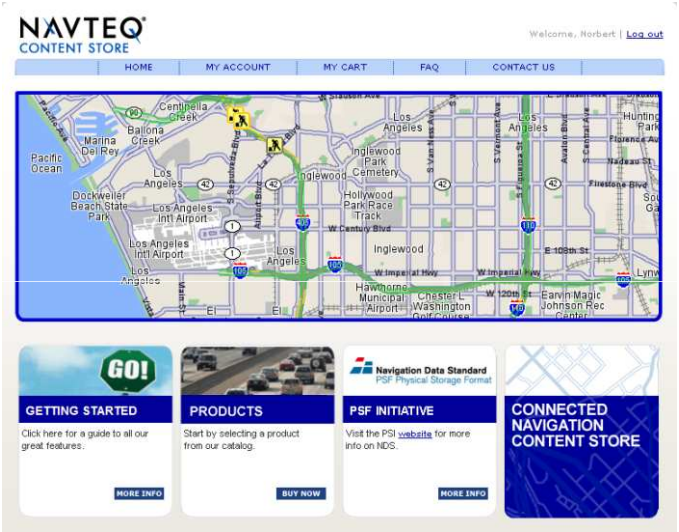
1.	Start on **Schaumburg Ct** and head towards Summit Dr.	623 ft
2.	Turn right onto **Summit Dr**.	558 ft
3.	Turn right onto **E Schaumburg Rd**.	2.01 miles
4.	Turn left onto N **Martingale Rd**.	1411 ft
5.	Head straight on **Martingale Rd**.	4757 ft
6.	Head straight on **Rohlwing Rd**.	1870 ft
7.	Turn left onto **Biesterfield Rd (Il-53)**.	886 ft
8.	Exit Biesterfield Rd (Il-53) and enter **I-290 East** .	8.78 miles
See Junction Views and Sign-as-Real		
9.	Exit I-290 East and enter **York Rd,** N **York St**.	3904 ft
10.	Head straight on N **York St**.	1411 ft
11.	You have arrived at your destination N **York St**.	0 ft

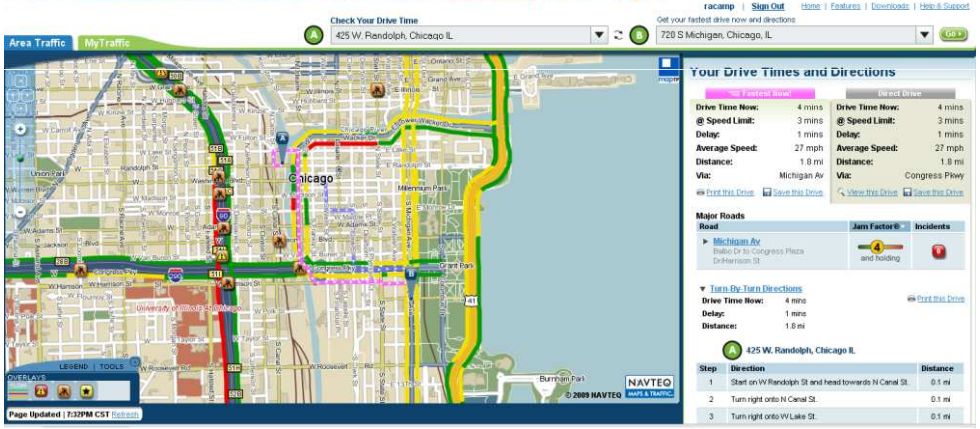


JBoss WORLD CHICAGO 2009

Map Products

- Portals
- PNDs
- Mobile Phones
- Vehicle Head Units





JBoss WORLD CHICAGO 2009

NAVTEQ Map Maintenance Automation

NAVTEQ

presented by

Business problem

- Multiple existing Map Maintenance scripts that have to be executed together in order to achieve business goal, for example, download map context, transform map data, find matches with other map sources, run validation, etc.
- Sequence of scripts execution is important.
- Common data required for scripts execution is currently entered multiple times, thus creating a potential for data errors
- Error recovery can be fairly complex and requires knowledge of the scripts executed to date.
- Poor communications structure existing for cooperation between script execution participants leads to extended execution time.
- Poor visibility into the state of execution makes it difficult to understand where exactly the problems are.

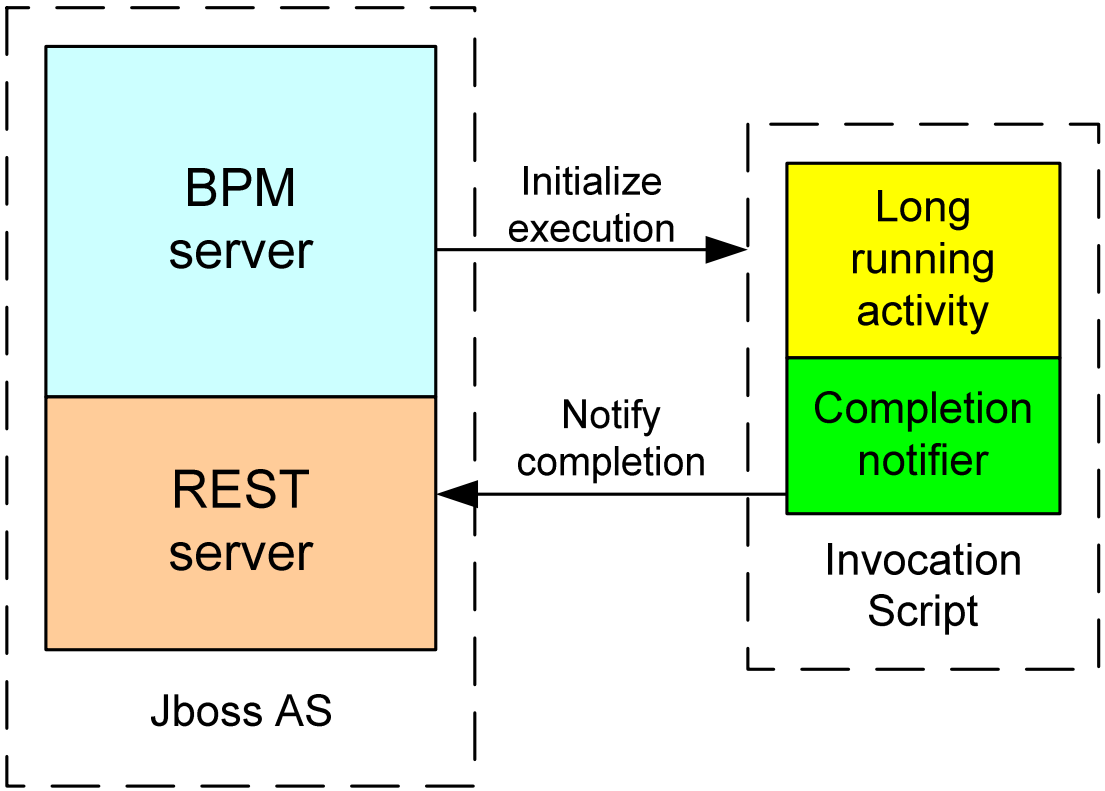
Overall solution approach

- Implement business processes, orchestrating map maintenance scripts execution for different use cases.
- Business processes execute existing scripts with minimal/no changes to them.
- Common data, required for individual script execution is stored in the process execution context
- Human interactions are used extensively for entering required data and decision making.
- Email is used extensively for notifying process participants about abnormal execution of scripts and tasks ready for execution and management about
- Multiple UIs are required for viewing script execution results, entering data, decision making, process state/variables viewing, etc.

Why JBoss jBPM

- Simple-to-use, but quite powerful, extensible Java framework.
- Eclipse-based plug-in, allowing for visual design of business processes and simplifying interaction between process designers and implementers.
- Support for creation on reusable library of nodes for a specific sets of business problems.
- A rich set of Java APIs for programmatic manipulation of processes, users' tasks, etc.
- Fairly active user community, based on forum and mail list activity, which simplifies issues resolution and provides a lot of resources on specific solutions and best practices.
- JUnit - based testing options, which does not require application server, and allows for test driven development of processes.
- Simple web management tool – jBPM console
- Availability of training and paid support for installation and maintenance from JBoss.

Invoking long running activities

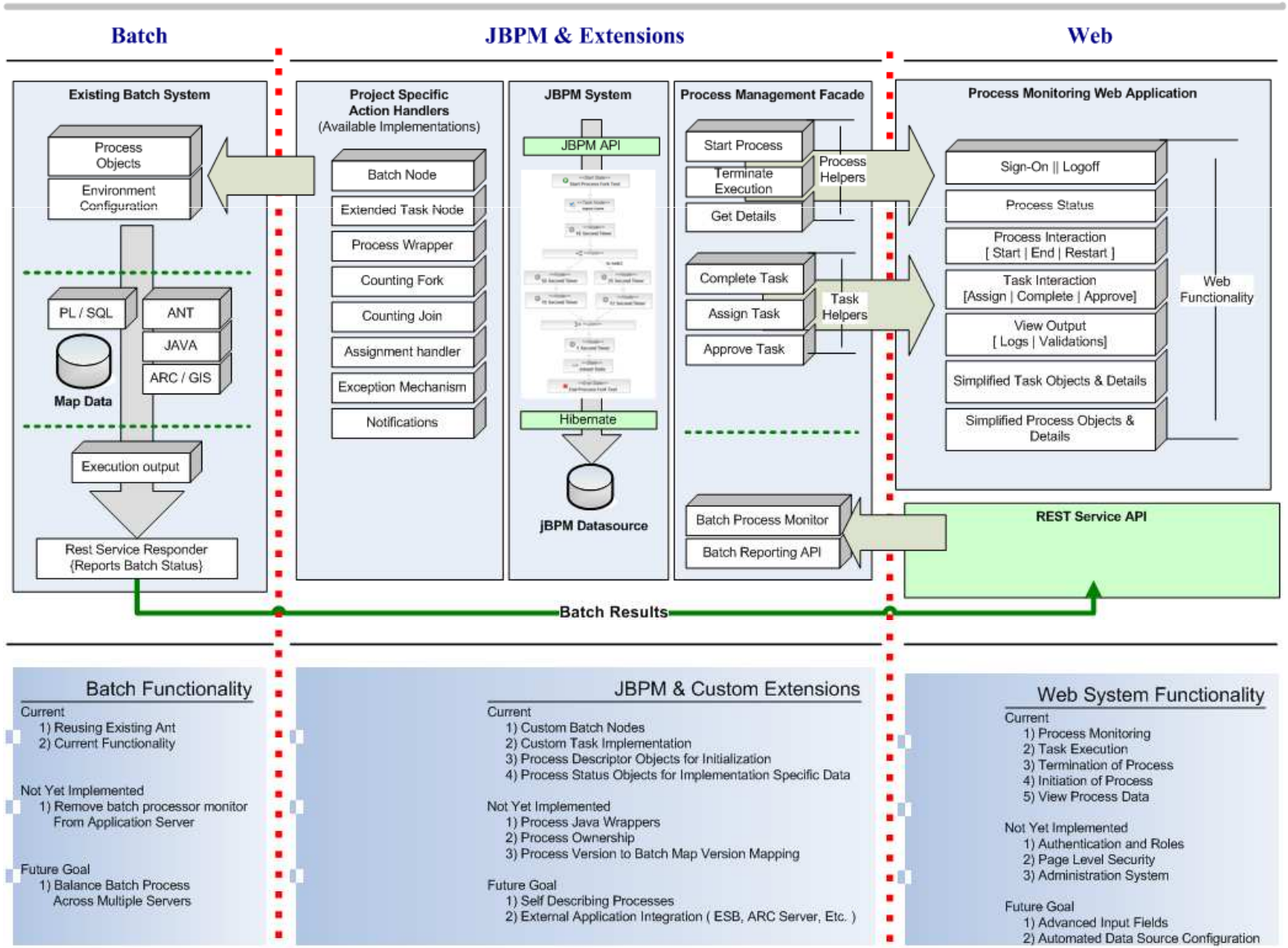


- Activity invoker node starts long running activity and return
- jBPM server orchestrates activity invokers
- REST server (RestEasy) exposes jBPM APIs (subset) for remote invocation
- Completion notifier uses Rest server to set execution results and signal process to continue

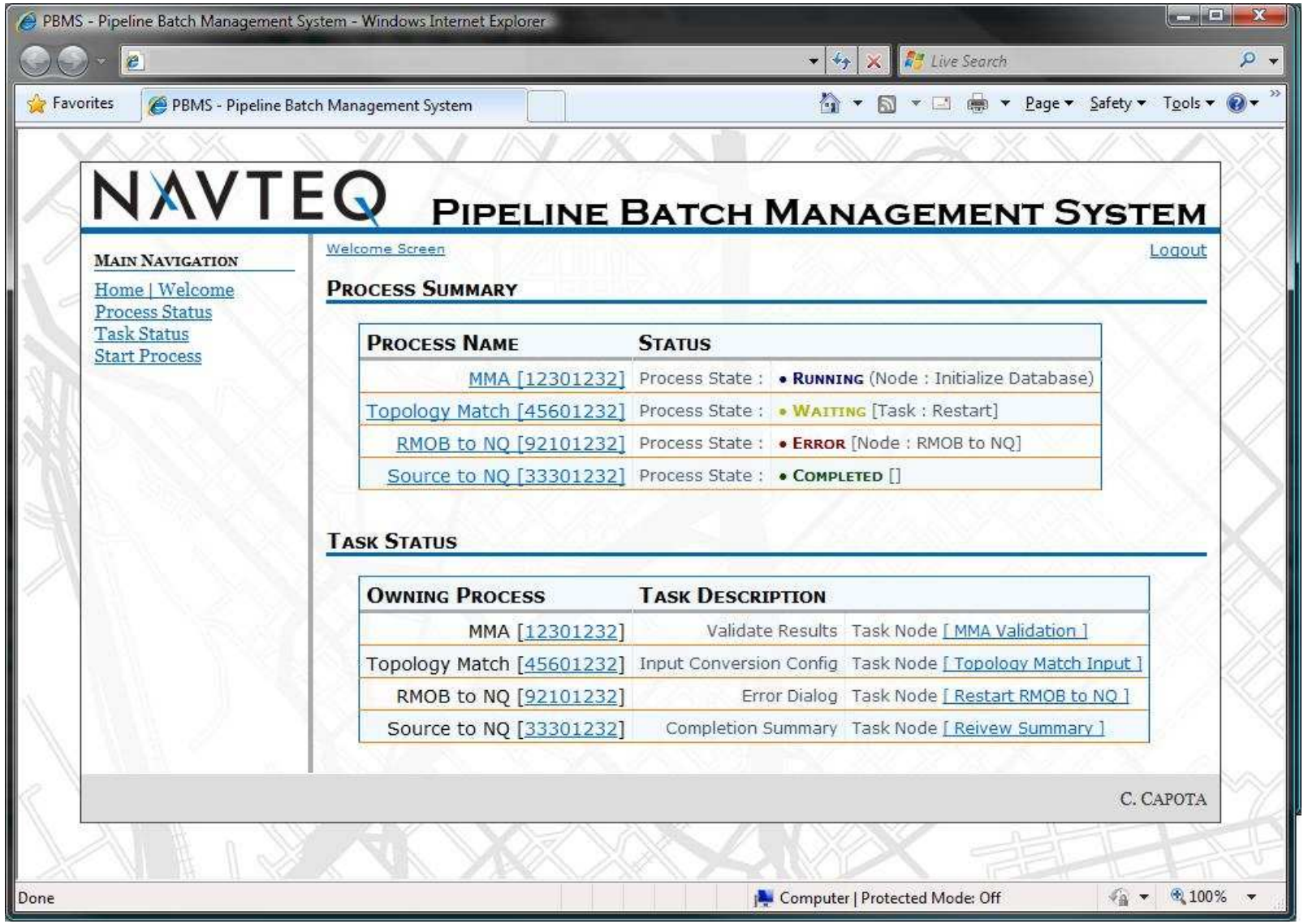
Implementing advanced user interactions patterns.

- **The 4-eyes principle**, often referred to as “separation of duties,” is a common scenario when a decision is made by two or more people independently of one another. In many cases, it is simply enough to obtain a second opinion/signature.
- **Nomination** is a situation when a supervisor manually assigns tasks to his team members based on their schedule or workload constraints or expertise.
- Tasks are often modeled to express the expectation that they will be completed within a certain time frame. If a task is not progressing as expected, an **escalation** mechanism is required. Two typical escalation implementations are - reassignment of the tasks, often with a notification, that an escalation has occurred and notification (typically to a manager) that a task has not been completed in time.
- **Chained execution** is a process (fragment) where a sequence of steps is executed by the same person.

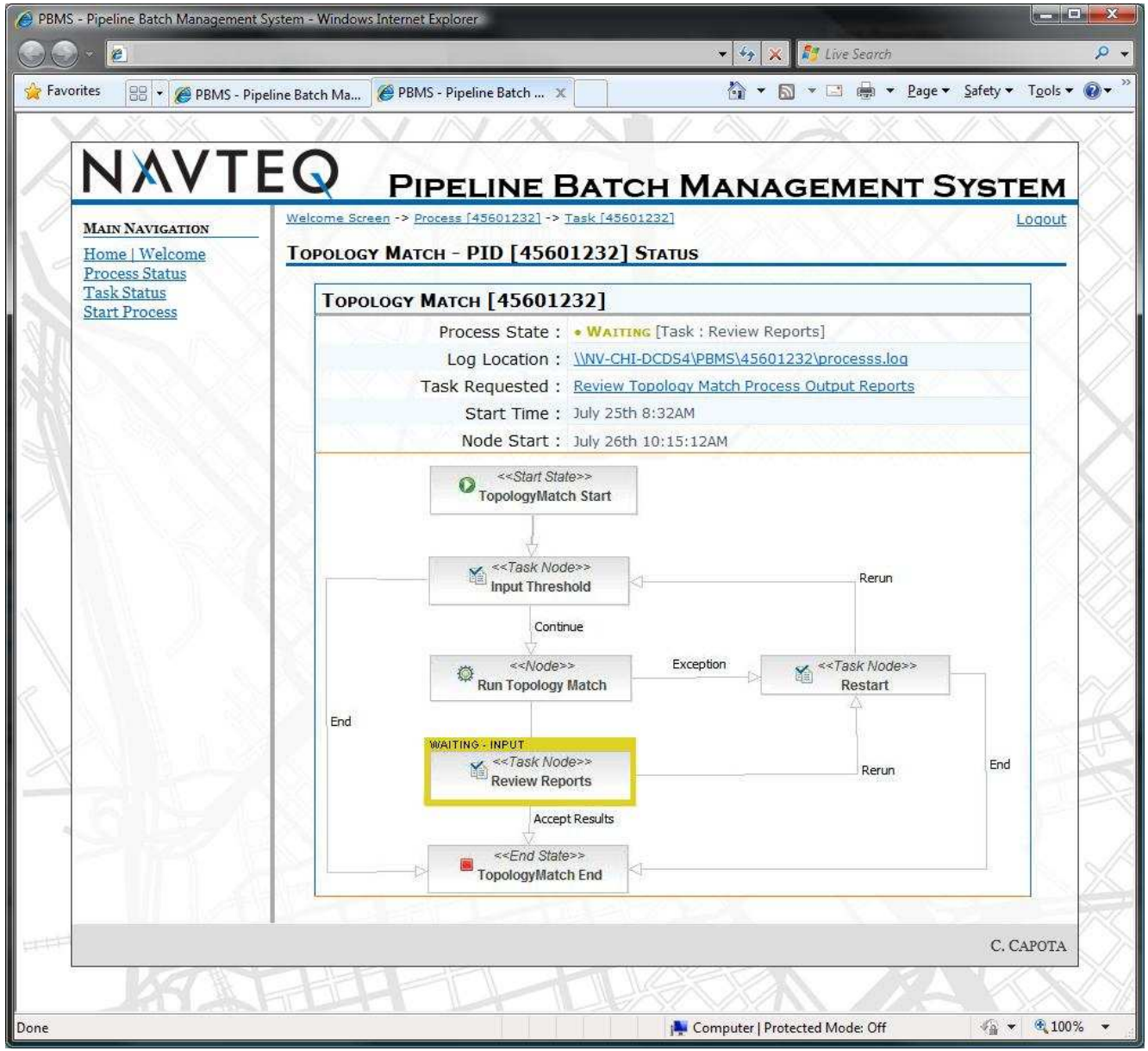
Map maintenance Architecture



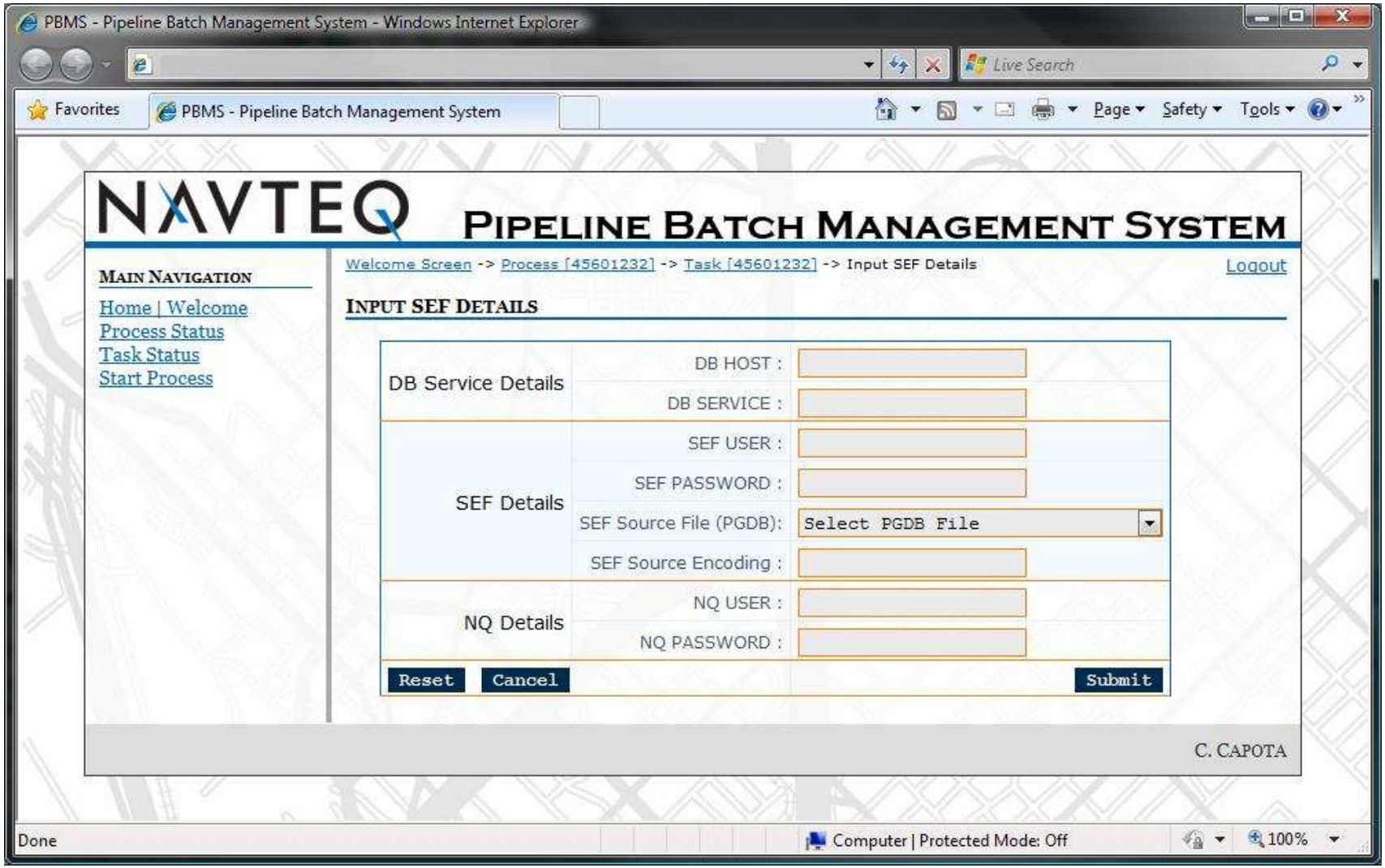
User summary



Process details



Task Input


PBMS - Pipeline Batch Management System - Windows Internet Explorer
Favorites
PBMS - Pipeline Batch Management System
Page
Safety
Tools
NAVTEQ
PIPELINE BATCH MANAGEMENT SYSTEM
MAIN NAVIGATION
Home | Welcome
Process Status
Task Status
Start Process
Welcome Screen -> Process [45601232] -> Task [45601232] -> Input SEF Details
Logout
INPUT SEF DETAILS
DB Service Details
DB HOST :
DB SERVICE :
SEF Details
SEF USER :
SEF PASSWORD :
SEF Source File (PGDB):
Select PGDB File
SEF Source Encoding :
NQ Details
NQ USER :
NQ PASSWORD :
Reset
Cancel
Submit
C. CAPOTA
Done
Computer | Protected Mode: Off
100%

Conclusion

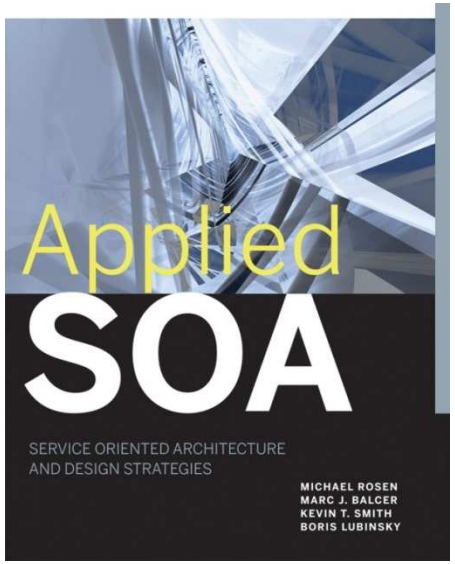
Lessons Learned - Approach

- Get senior management buy-in up front
  - Difficult to find success stories
  - Difficult to find benchmarks
  - Hard to dispel irrational fears
- Good middleware enables architecture to survive implementation

Lessons Learned - JBoss

- JBoss middleware is very powerful but very demanding.
  - Steep learning curve
  - Plan for training (formal and self)
  - Technical Support not a substitute for training
  - Practice makes it perfect
  - Product innovations and changes require ongoing learning
- Documentation varies in completeness by product.
- Finding implementation examples is often like finding a needle in a haystack.
- Availability of source code is a blessing and a curse.
  - Allows for understanding of how things work to compensate for lack of architecture/design documentation.
  - Provides a way to enhance/modify/customize behavior
  - Warning: Source reading may cause heavy drinking and sleepless nights

For More information



Michael Rosen , Boris Lublinsky, Kevin T. Smith , Marc J. Balcer. Applied SOA: Service-Oriented Architecture and Design Strategies.

http://www.amazon.com/Applied-SOA-Service-Oriented-Architecture-Strategies/dp/0470223650

Boris Lublinsky. Using JBoss ESB and JBPM for Implementing VMS Solutions
http://www.infoq.com/articles/jboss-esb-jbpm

Boris Lublinsky. Orchestrating Long Running Activities with JBoss / JBPM

http://www.infoq.com/articles/boris-lublinsky-jboss-jbpm

Boris Lublinsky. Supporting Advanced User Interaction Patterns in jBPM

http://www.infoq.com/articles/jBPM-user-interaction-patterns

Acknowledgements

- To our many colleagues in NAVTEQ for supporting our efforts and helping with implementations
- To JBoss Architects for patiently explaining to us what we were doing wrong
- To numerous JBoss users posting examples of JBoss middleware - based implementations, which help to understand better how different products work and should be used.
- To all bad implementations, that we have produced for a chance to learn about how ***not*** to do things.

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/JBOSSWORLD-SURVEY