



Java is a trademark of Sun Microsystems, Inc.

# JavaOne<sup>SM</sup>

## SOA at Enterprise Scale Solving Real Challenges with GlassFish ESB

Istvan Molnar  
Geza Simon  
Akos Jancsik  
Drotposta Consulting Ltd

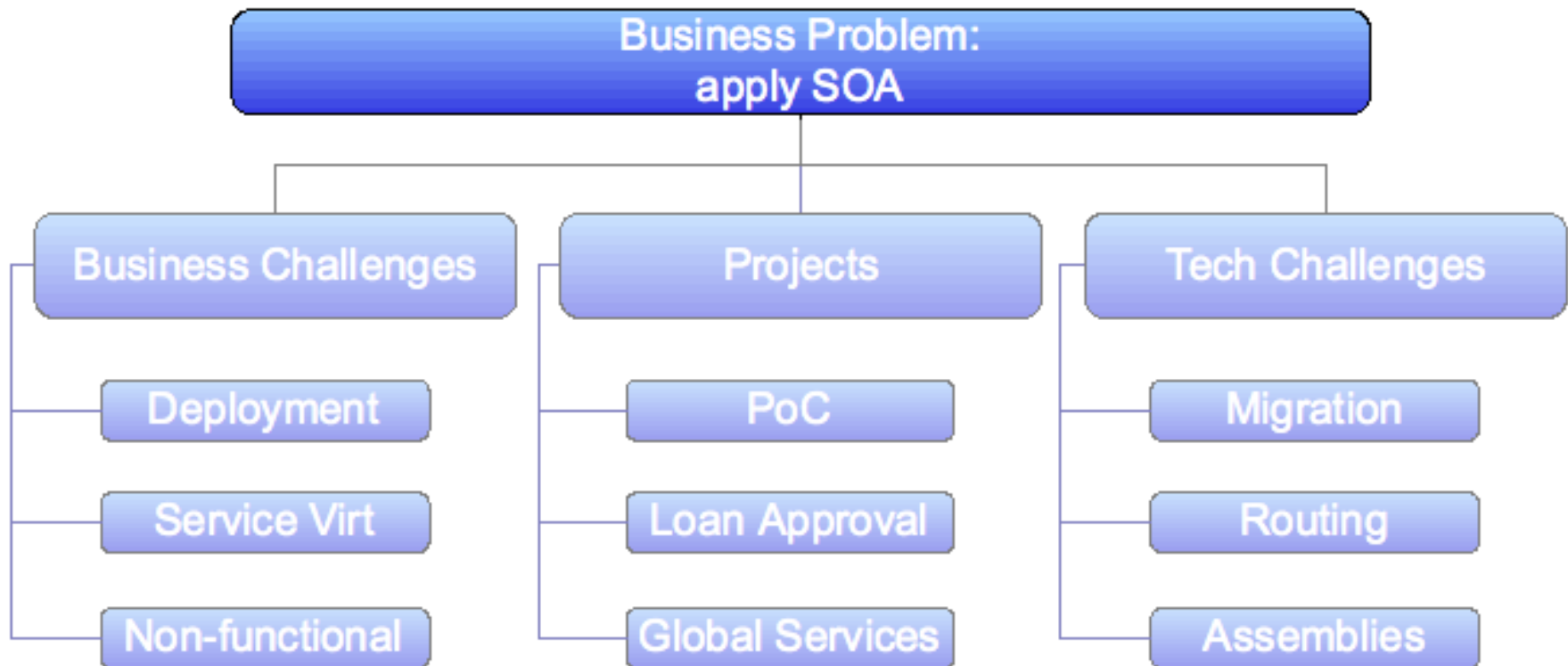
# SOA at Enterprise Scale

## Agenda

- > Fortune 100 client wants to apply SOA to solve integration challenges
- > Business Problem is presented
  - with Business Challenges
- > Projects: PoC, Loan-approval, Global services
- > Technical Challenges
- > Demo and Code Examples
- > Lessons Learned, What Next

# SOA at Enterprise Scale

## Agenda

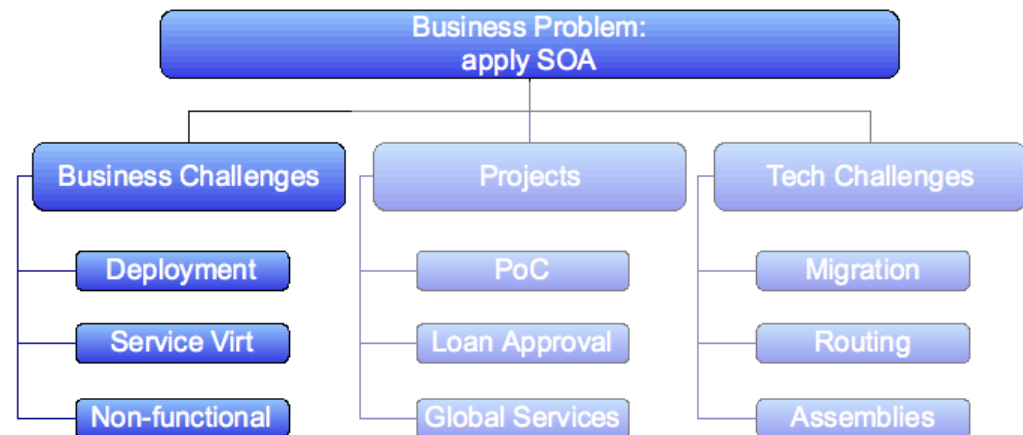


## > Implement SOA

- proven
- standards-based
- open
- can scale to enterprise-size
- complex, multi-level processes

# Business Challenges Overview

- > Deployment considerations
  - Geographically distributed
  - Local processes
  
- > Systemic Qualities
  - Complexity
  - High TPS
  - Scaling

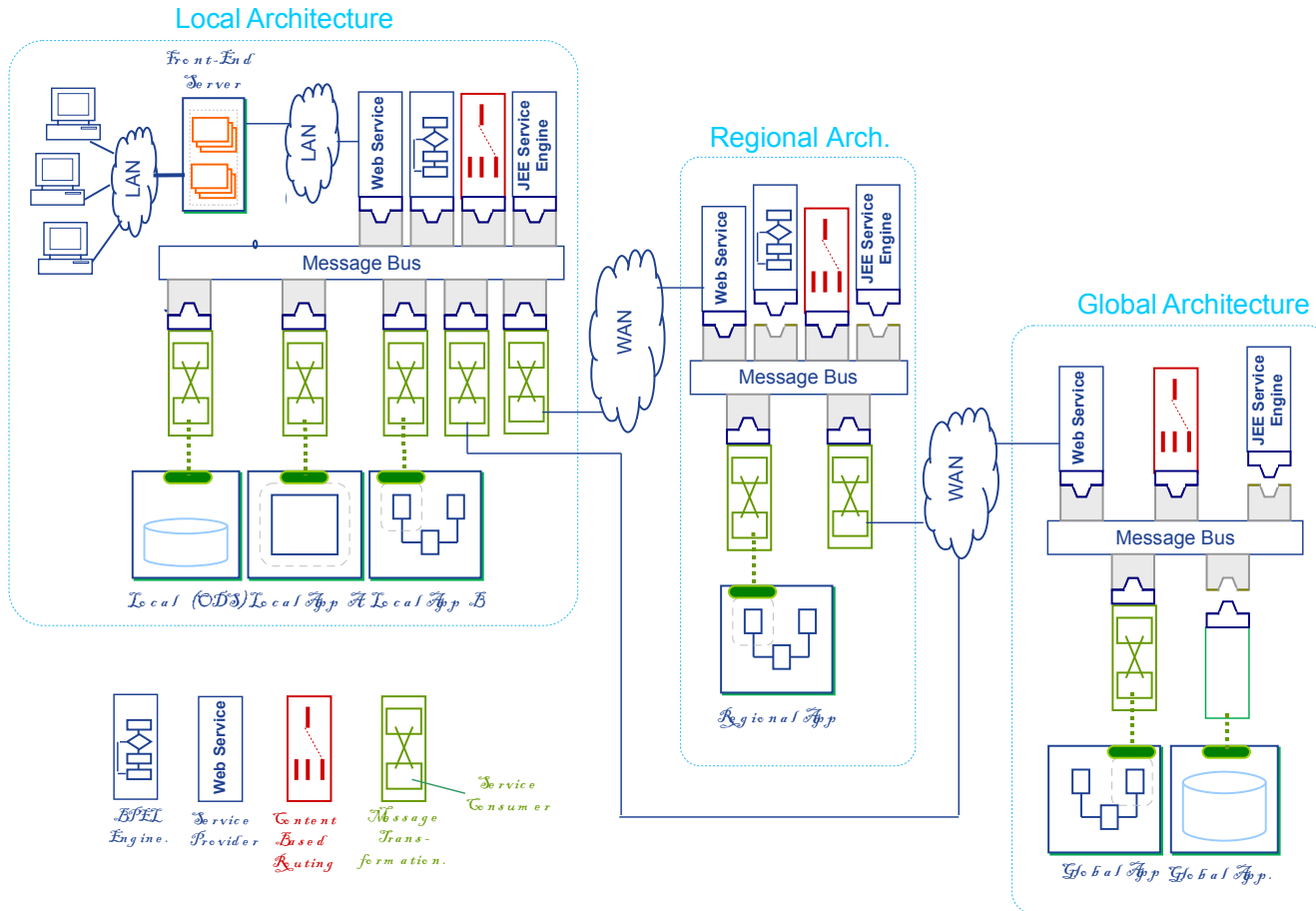


# Business Challenges > Deployment

## Deployment Considerations

- > Levels
  - Global, Regional, Local services
- > Heterogenous structure
  - individually modified processes
  - country/business unit specific versions
  - local privileges to modify
- > But: code should be centrally managed

## Business Challenges > Deployment





# Business Challenges > Deployment

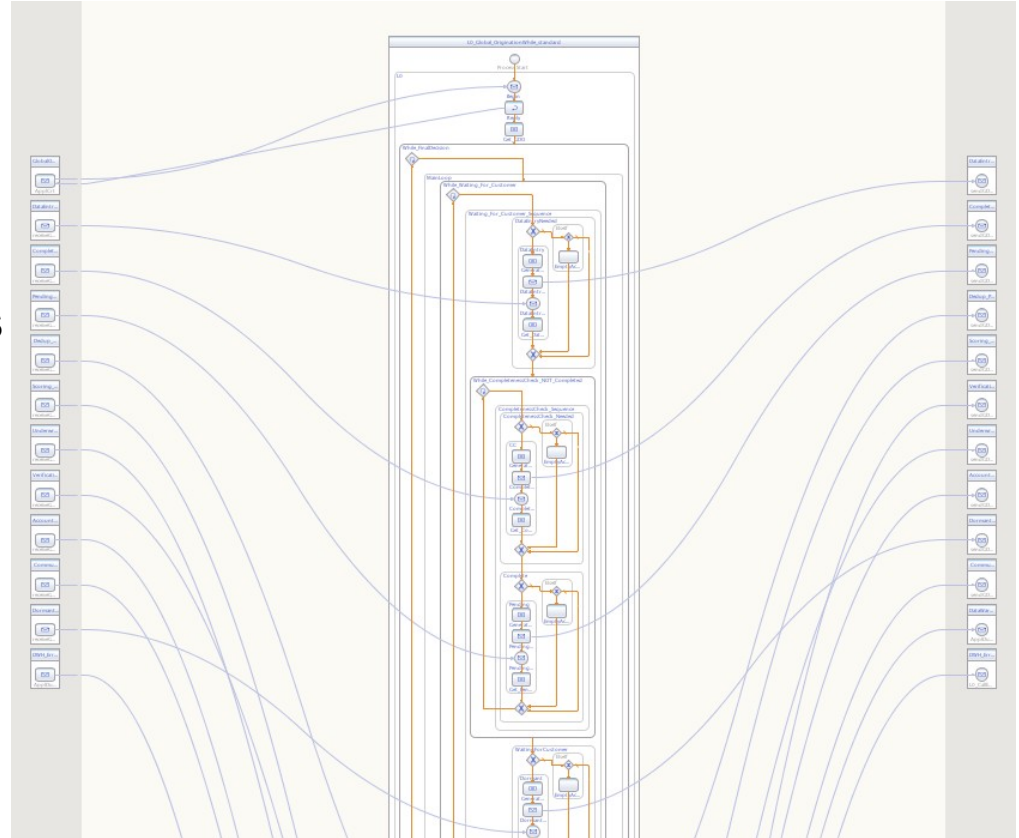
Some statistics

- > 50+ countries
- > 6 regional centers
- > 3+ global applications
  - with many shared services (now: 50)



# Business Challenges > Systemic Qualities

- > Complexity
  - Totally distributed
    - Raises its own technical challenges
- > Throughput required
  - 100 TPS / country  
= 2-3M per day
  - Scalability for future
- > Reliability

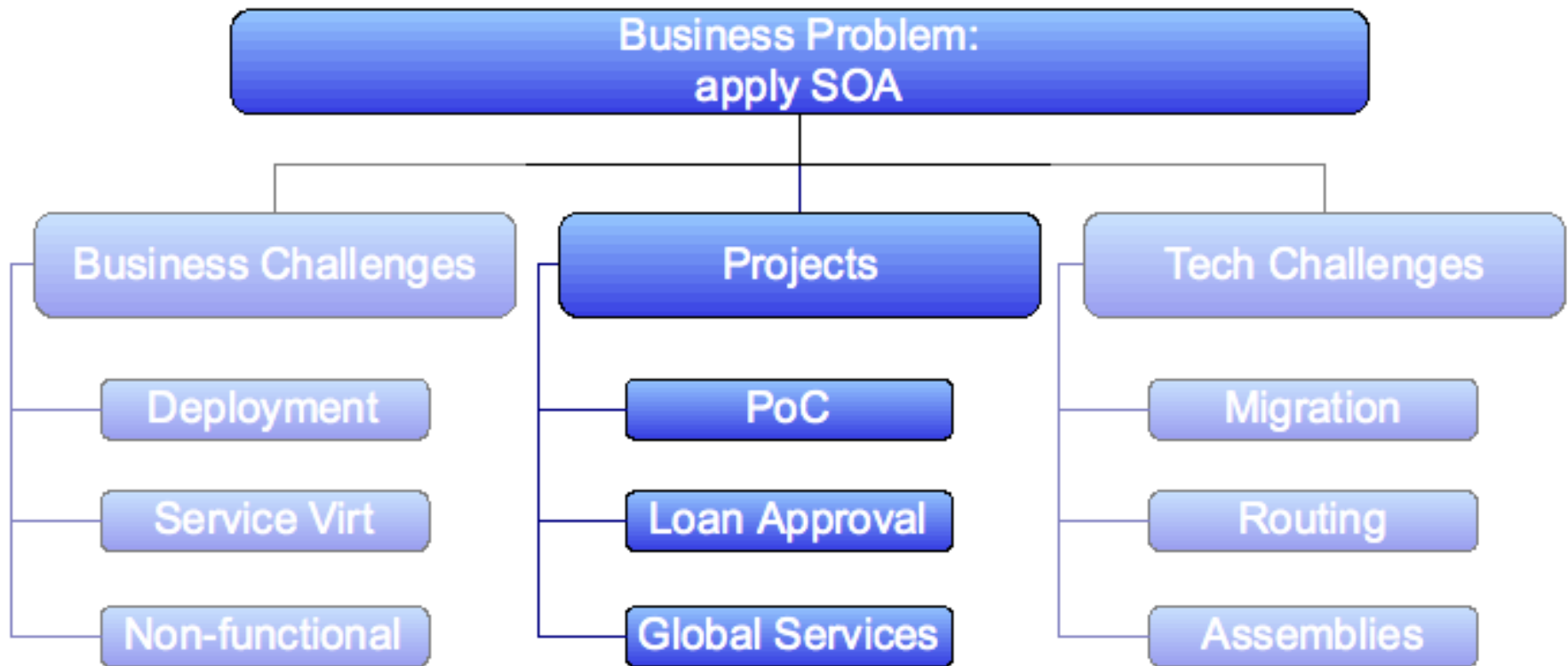


# Business Challenges > Systemic Qualities

- > Throughput required
  - 100 TPS / country
  - Scalability for future (more services – more users)

# SOA at Enterprise Scale

## Agenda



## PoC > Scope

Check if selected technology meets client's requirements

- > Various legacy systems e.g.
  - WebSphere MQ, Message Broker
  - Web services
  - Oracle stored PL/SQL procedures
- > Two milestones
  - Message Flow Implementation
  - Process Flow Implementation

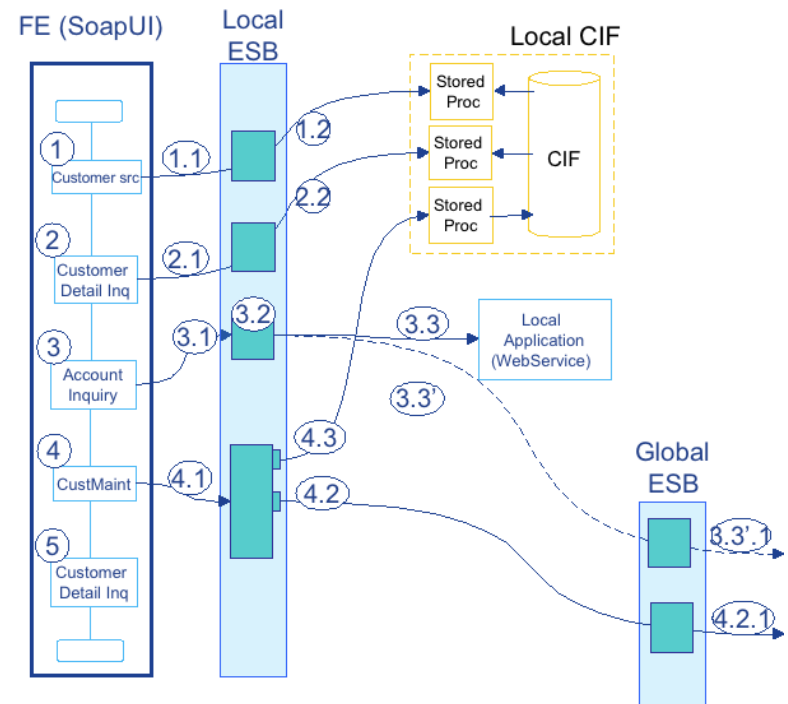
# PoC > Scope

## > Milestone 1: Message Flow Implementation

- Short-running transactions
- Automatic processing (no human interaction)

## > Produce Design Guidelines

- Content-based Routing
- Invoke Global Services
- Legacy interfaces



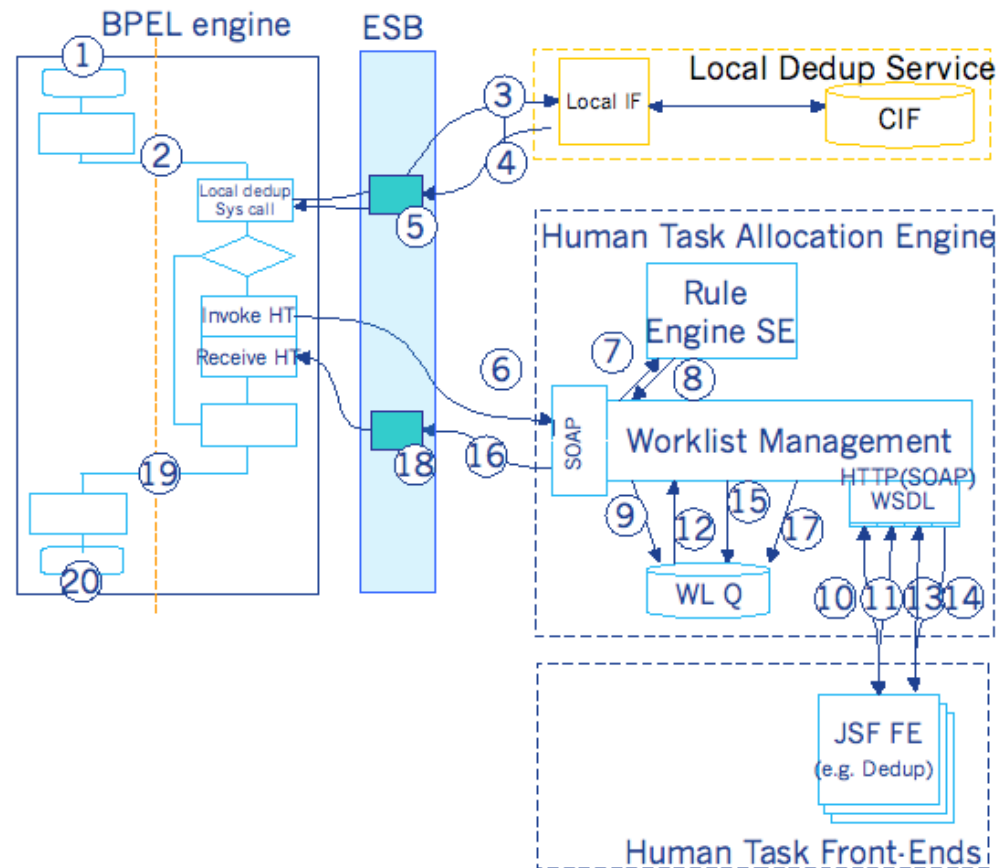
# PoC > Scope

## > Milestone 2: Process Flow

- Long-running transactions
- Persistent processes
- Including human workflow and tasks

## > Design Guidelines

- human tasklist
- rule based automatic task reallocation



## PoC > Scale

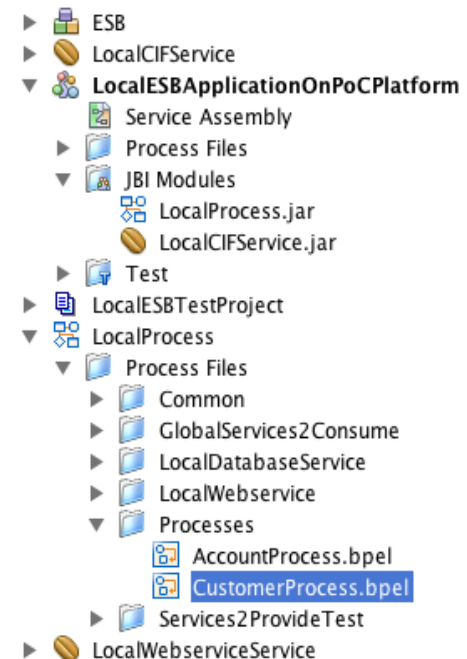
- > Around 6 months
- > Half a dozen NetBeans projects
- > Integration patterns created and prototyped
  - Specific to customer's business
  - For architecturally significant cases
- > Reaching 3000 TPS



# PoC > Implementation

## > Development environment

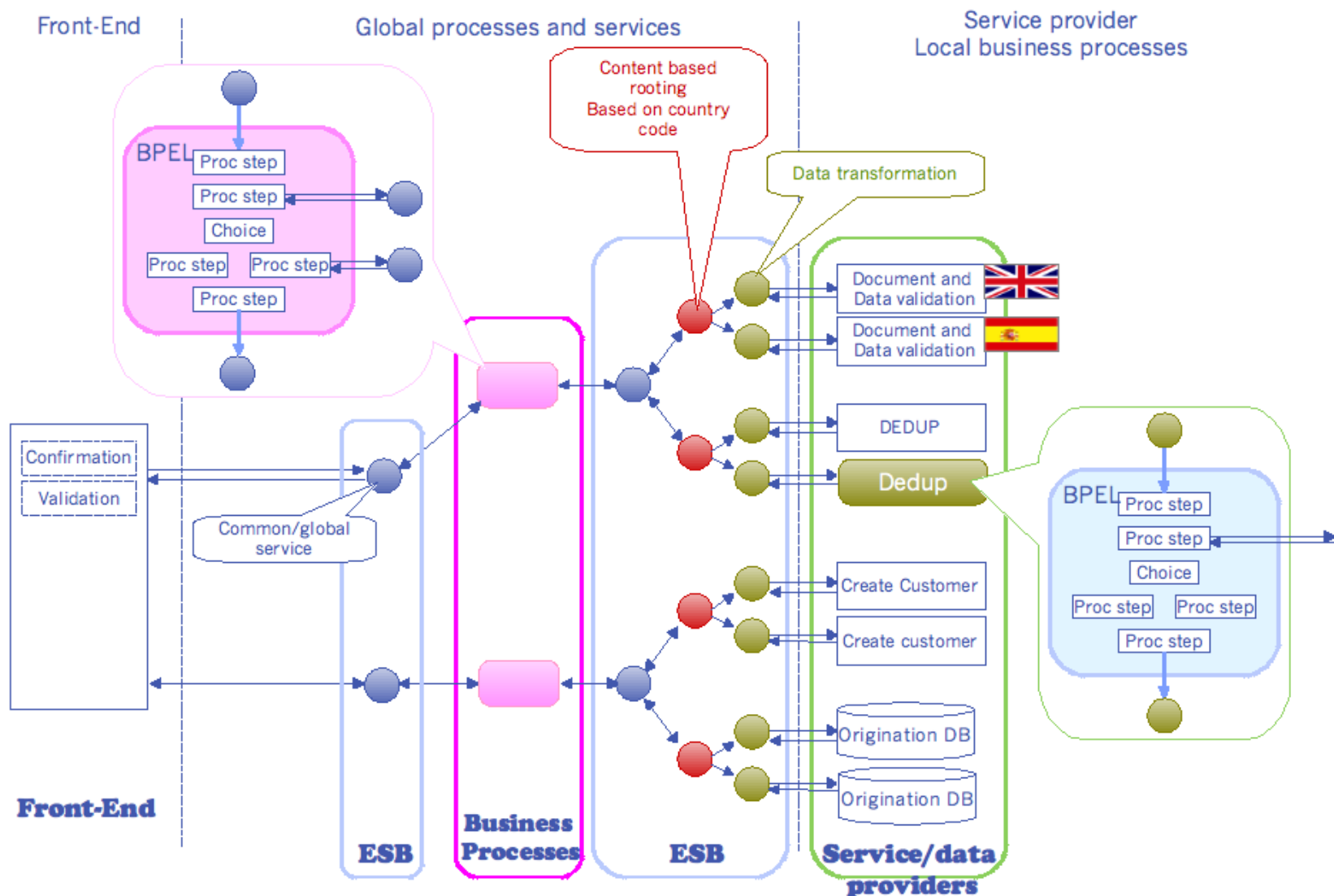
- NetBeans IDE
- GlassFish ESB
- soapUI
- Hulp Profiler
- Custom utilities



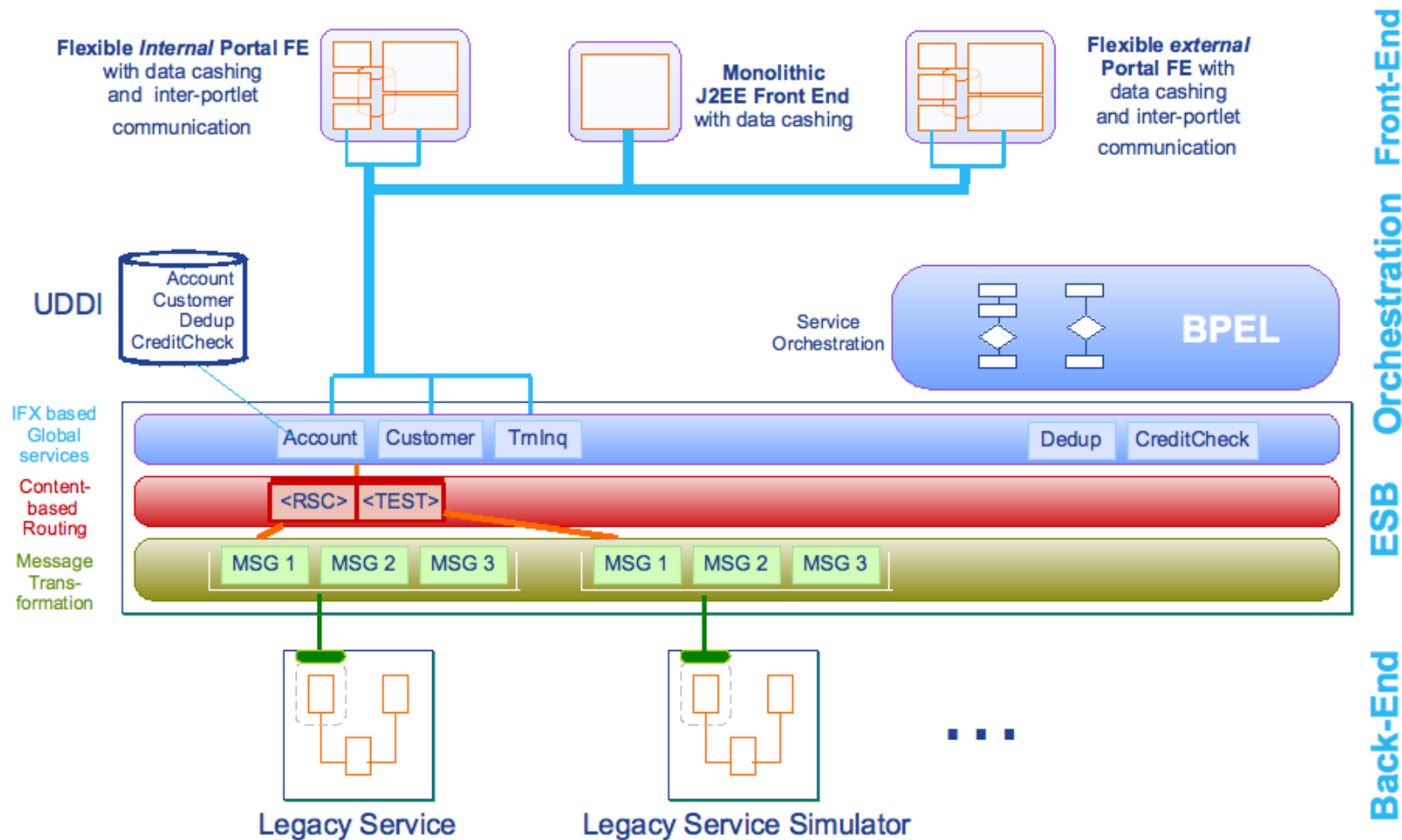
# Real-life Projects

- > After successful PoC
- > Loan-approval
  - Business process to run locally
  - Can be customized country-by-country
- > Global Services
  - To run globally
  - Accessible to all local processes

## Projects > Loan-approval

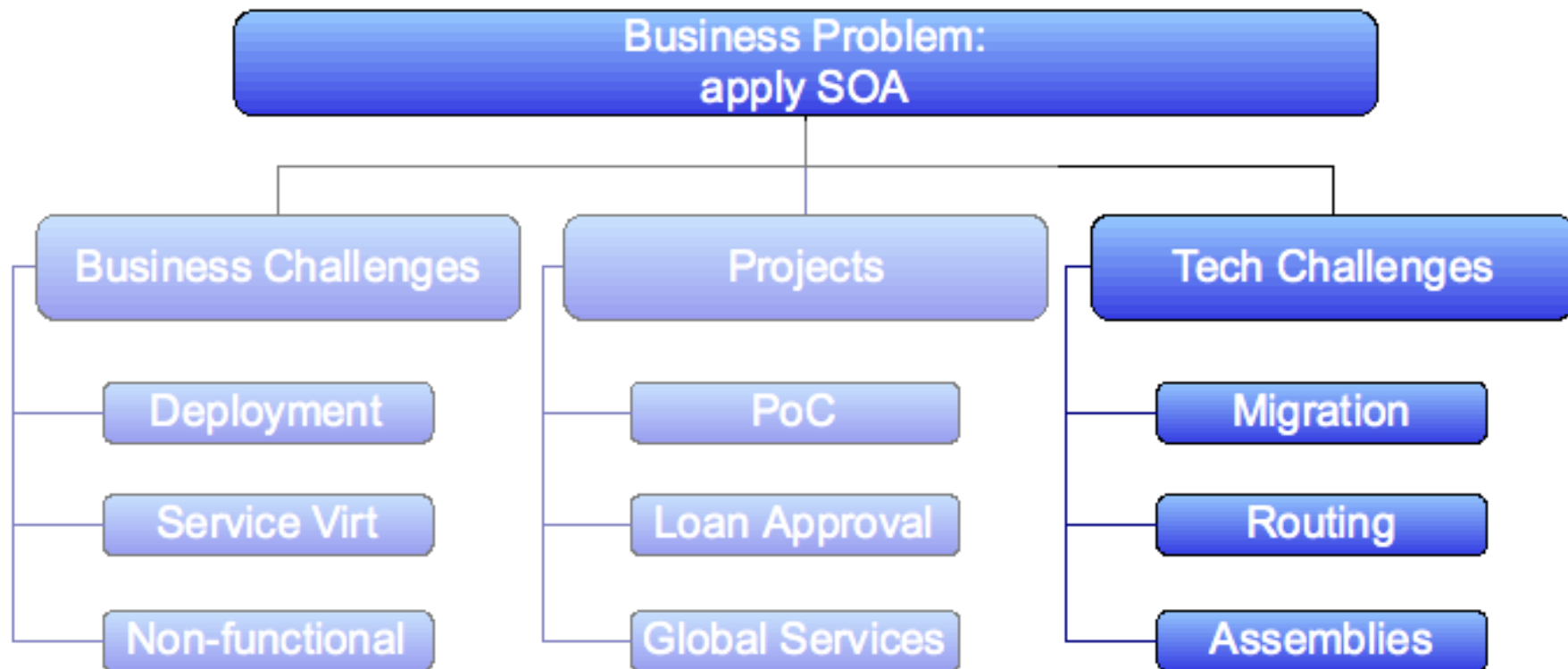


## Projects > Global Services



# SOA at Enterprise Scale

## Agenda



# Technical Challenges Overview

- > Legacy systems
  - migration
- > Totally distributed
  - two examples
- > Dynamic content-based routing
- > Assembling Composite Applications
  - many components, many connections
  - in many combinations!
- > Testing

# Technical Challenges > Migration 1

Migration to and from legacy systems:  
platform, specification and version  
independence

- > BPEL 1.0 to / from 2.0
  - Round-trip conversion utility developed
  - Extending community solution
  - Supports IBM Process Server extensions



# Technical Challenges > Migration 2

Migration to and from legacy systems:  
platform, specification and version  
independence

## > ESQL to XSLT

- 1-way conversion
- reusable
- 200+ IBM proprietary ESQL files
- extra functionality implemented as
  - custom XPATH functions
  - Java code

# Technical Challenges > Totally Distributed 1

- > Enterprise-sized, distributed ESB / JBI
- > Transactions over multiple ESBs
- > Long-running transactions
- > Many one-way messages
  - request–response–timeout: does not work!
- > Solution: WS-Reliable Messaging

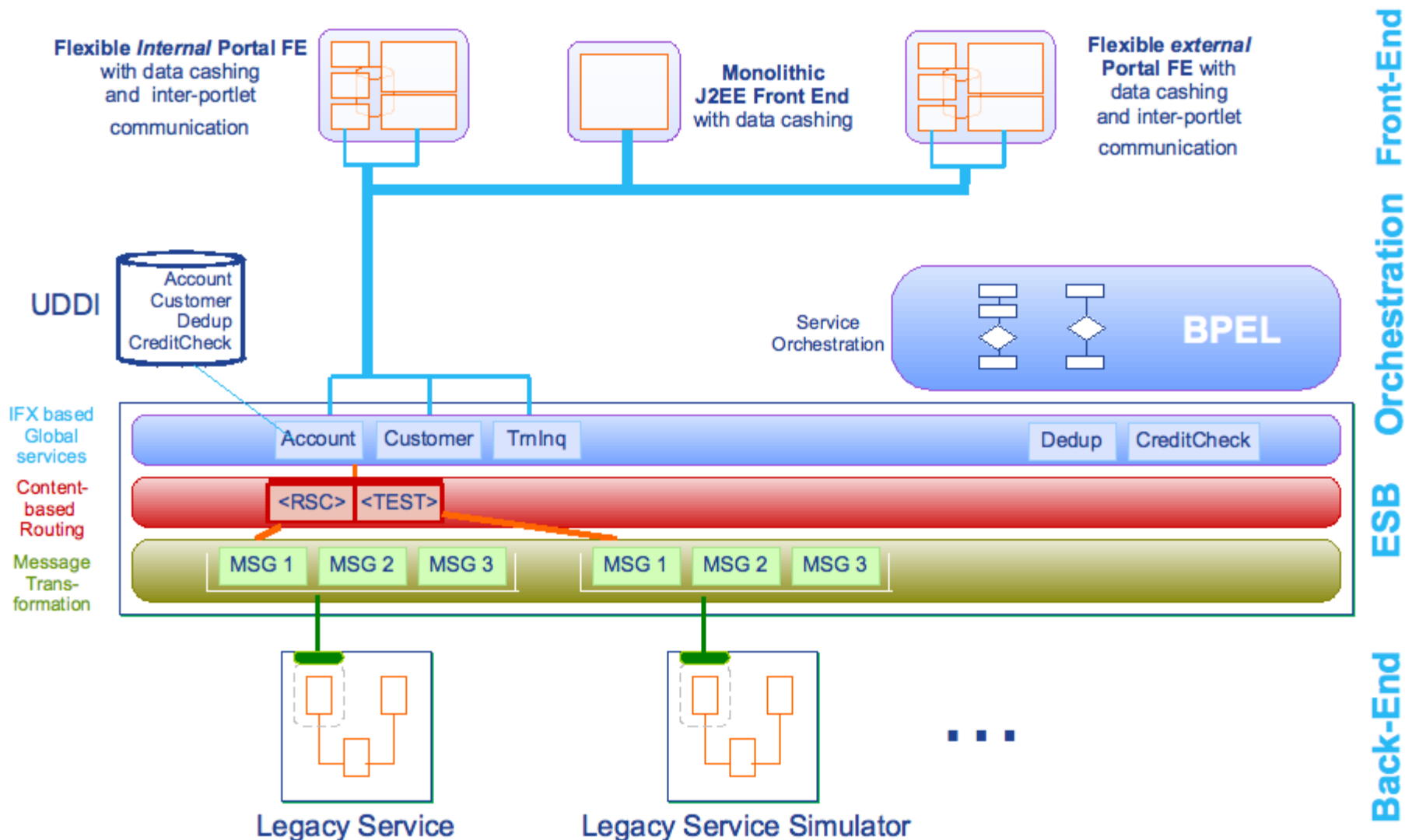
# Technical Challenges > Totally Distributed 2

- > Geographically distributed
  - Network bandwidth is an issue
  - Large messages
- Solution: Fast Infoset
  - W3 standard binary XML representation
- Typical measurement: 10 000 transactions
  - XML text: 60MB data transferred
  - Fast Infoset: 40MB

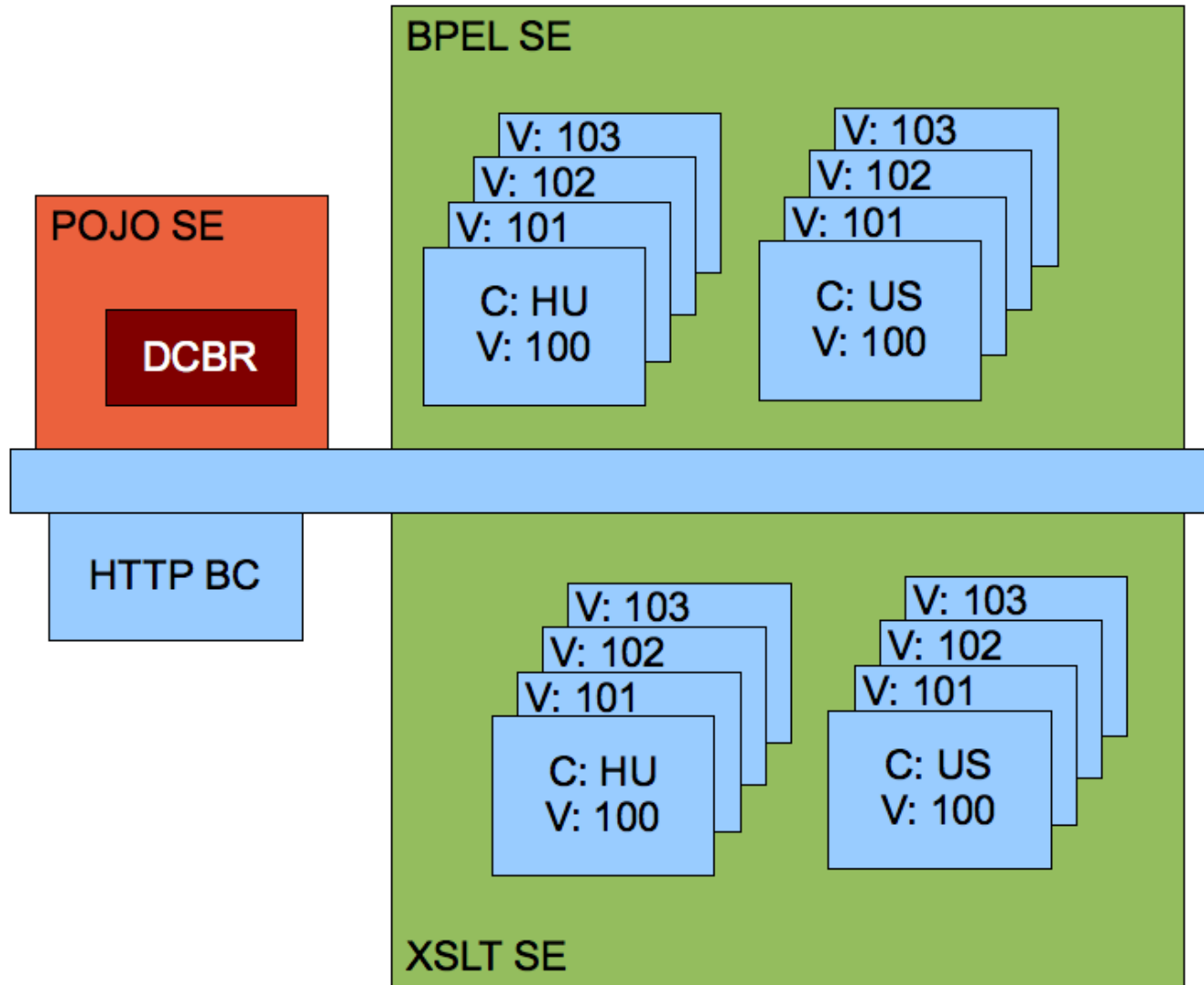
# Dynamic Content-based Routing

- > Local services or different versions of services
  - same business meaning – same *port type*
  - local implementations – different *message types*
- > They live together: same engine, same time
- > Solution:
  - common WSDL
  - country and version code in msg header
  - red layer interprets – routes accordingly
- > Implementation: POJO Service Engine

# Dynamic Content-based Routing



# Dynamic Content-based Routing

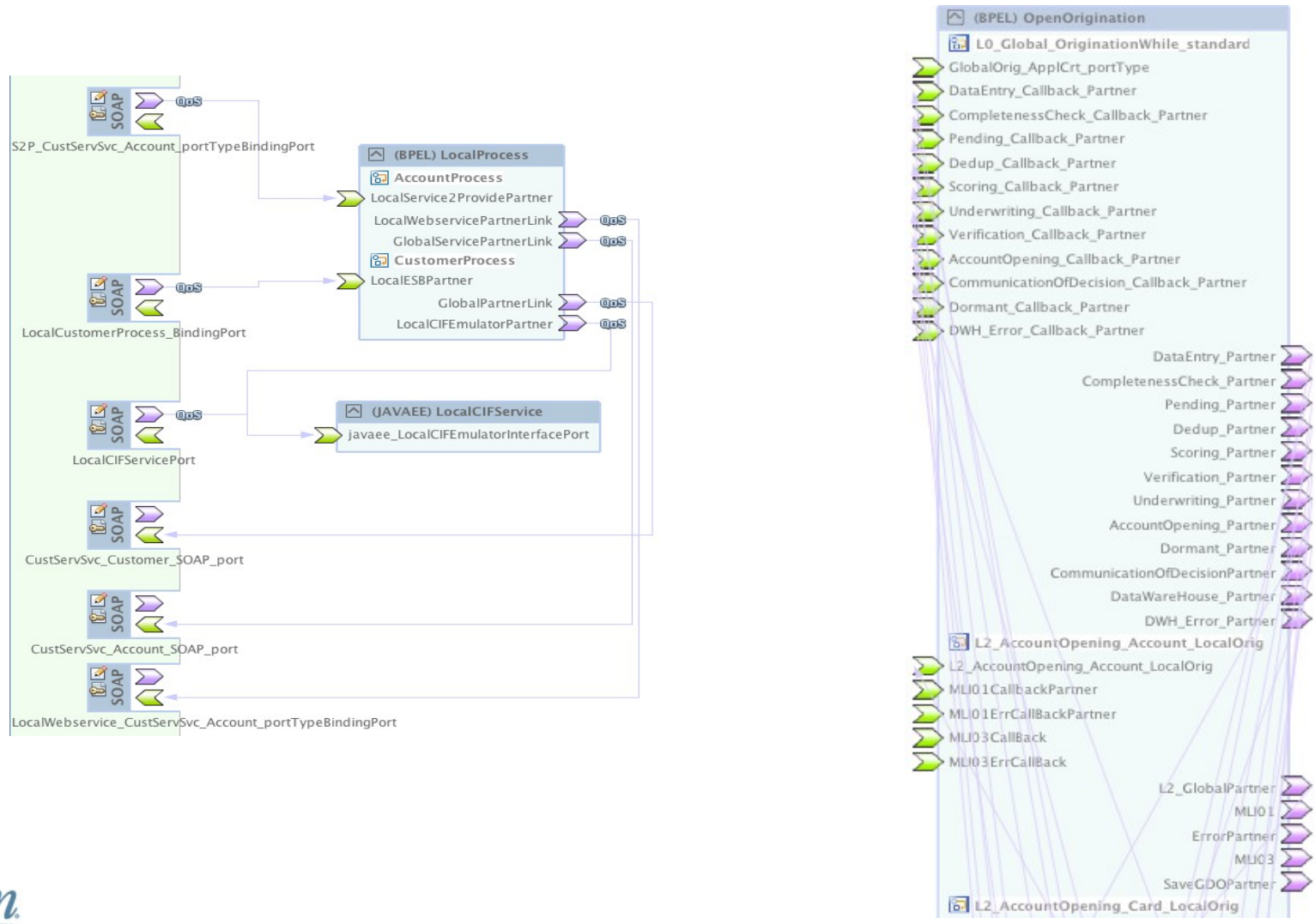


# Technical Challenges > Assembly

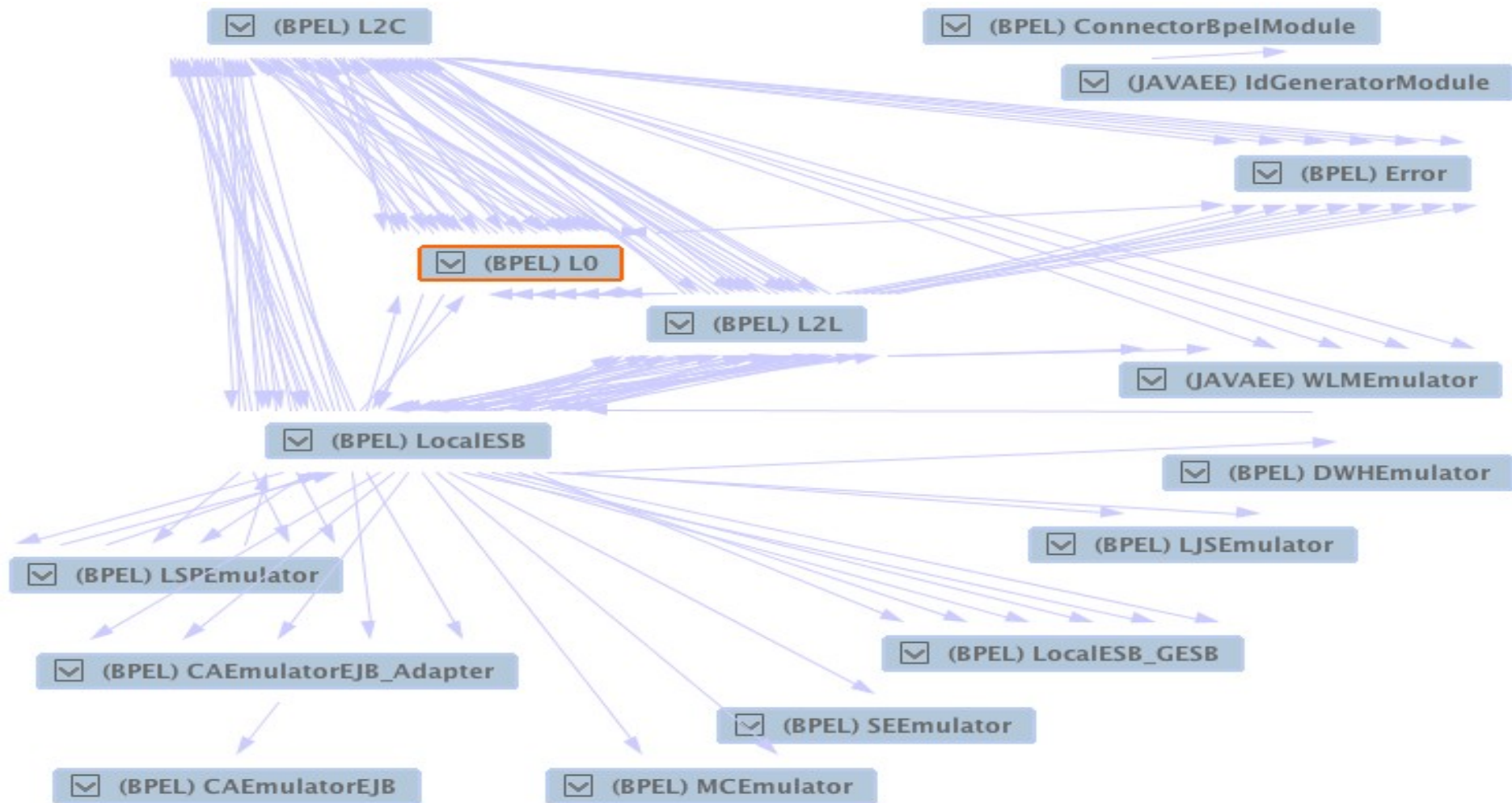
- > Requirement: assemble many components, many connections
- > NetBeans visual Service Assembly editor
  - suitable for smaller complexity
  - not automatized



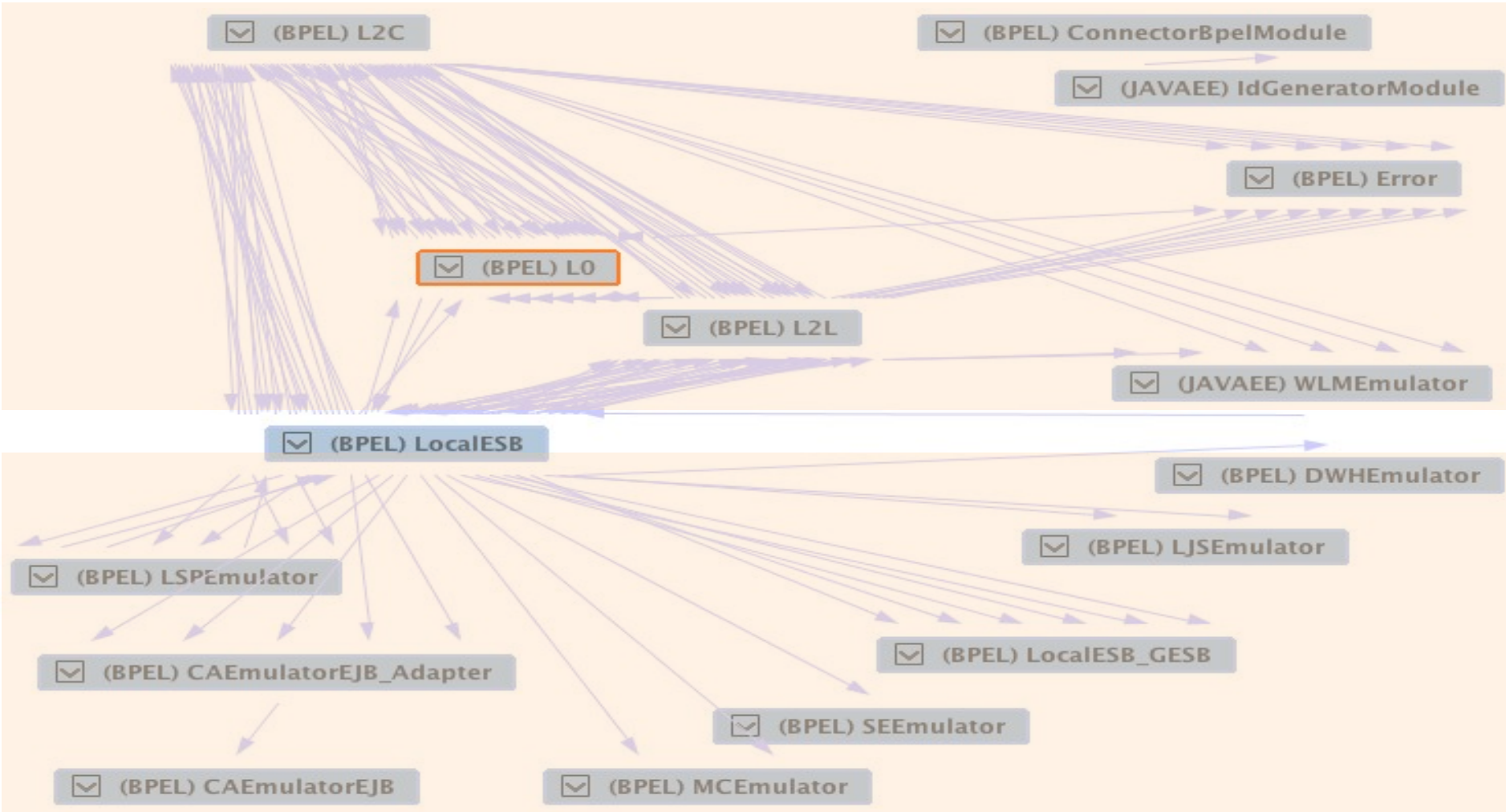
## Technical Challenges > Assembly



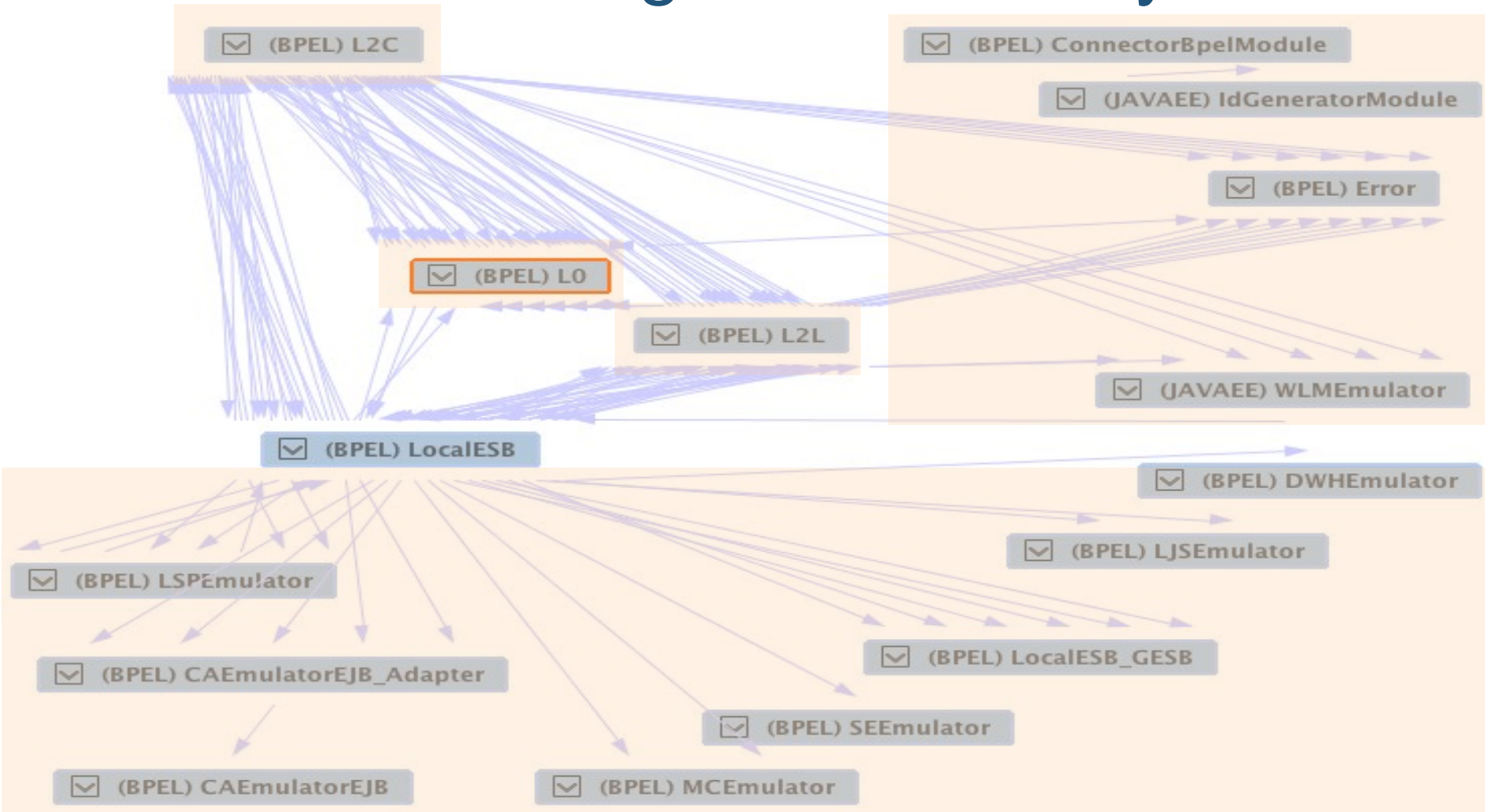
# Technical Challenges > Assembly



# Technical Challenges > Assembly



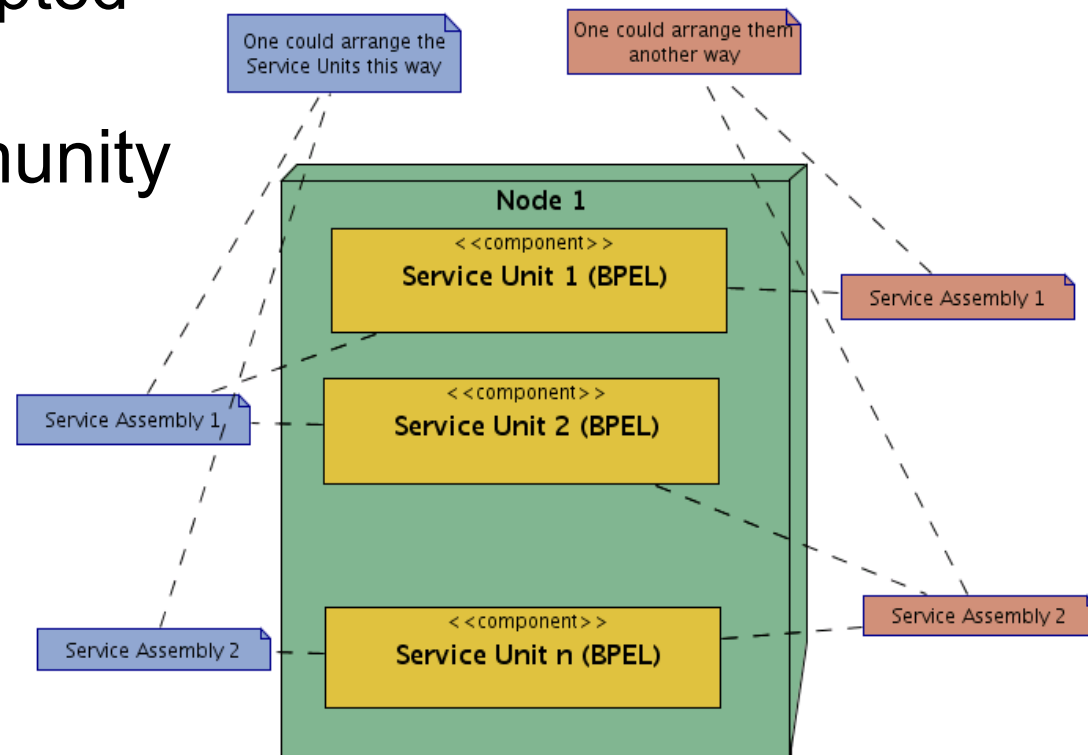
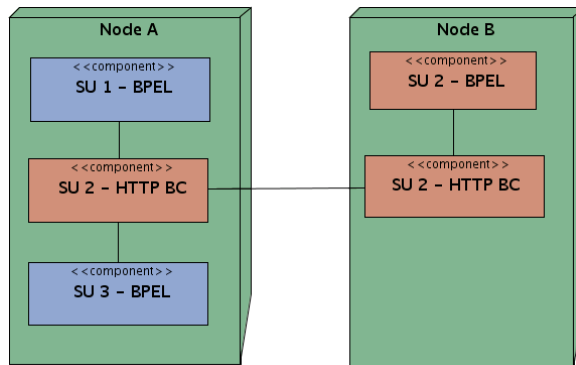
# Technical Challenges > Assembly





# Technical Challenges > Assembly

- > Solution: custom JBI connection builder
  - handles very large applications
  - use manually / scripted
- > Contributed to community



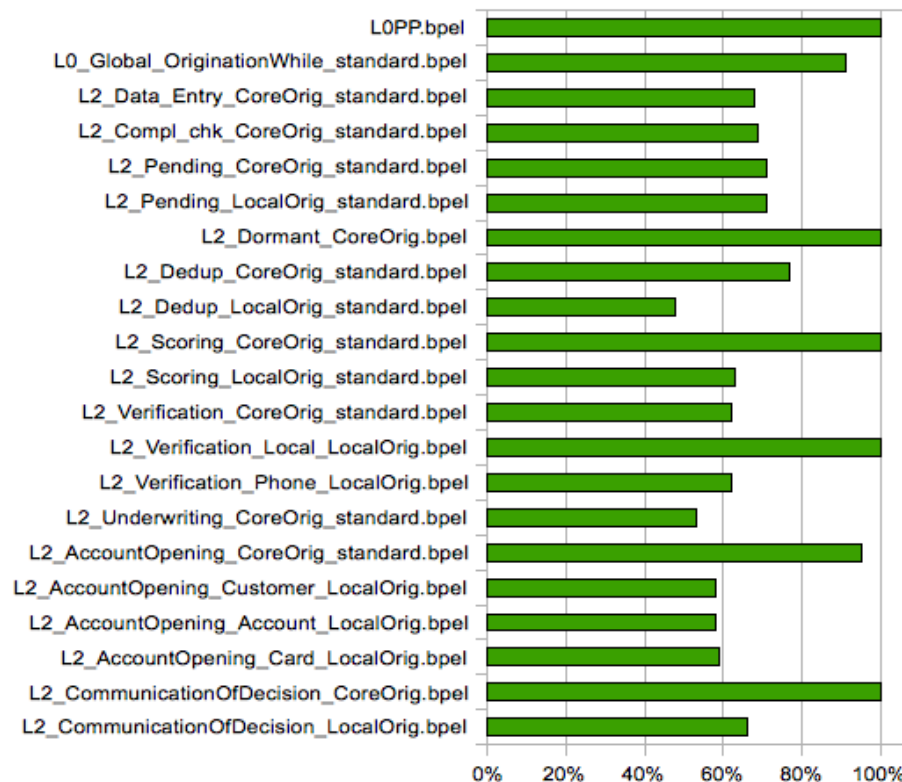
# Technical Challenges > Testing

## > Tools

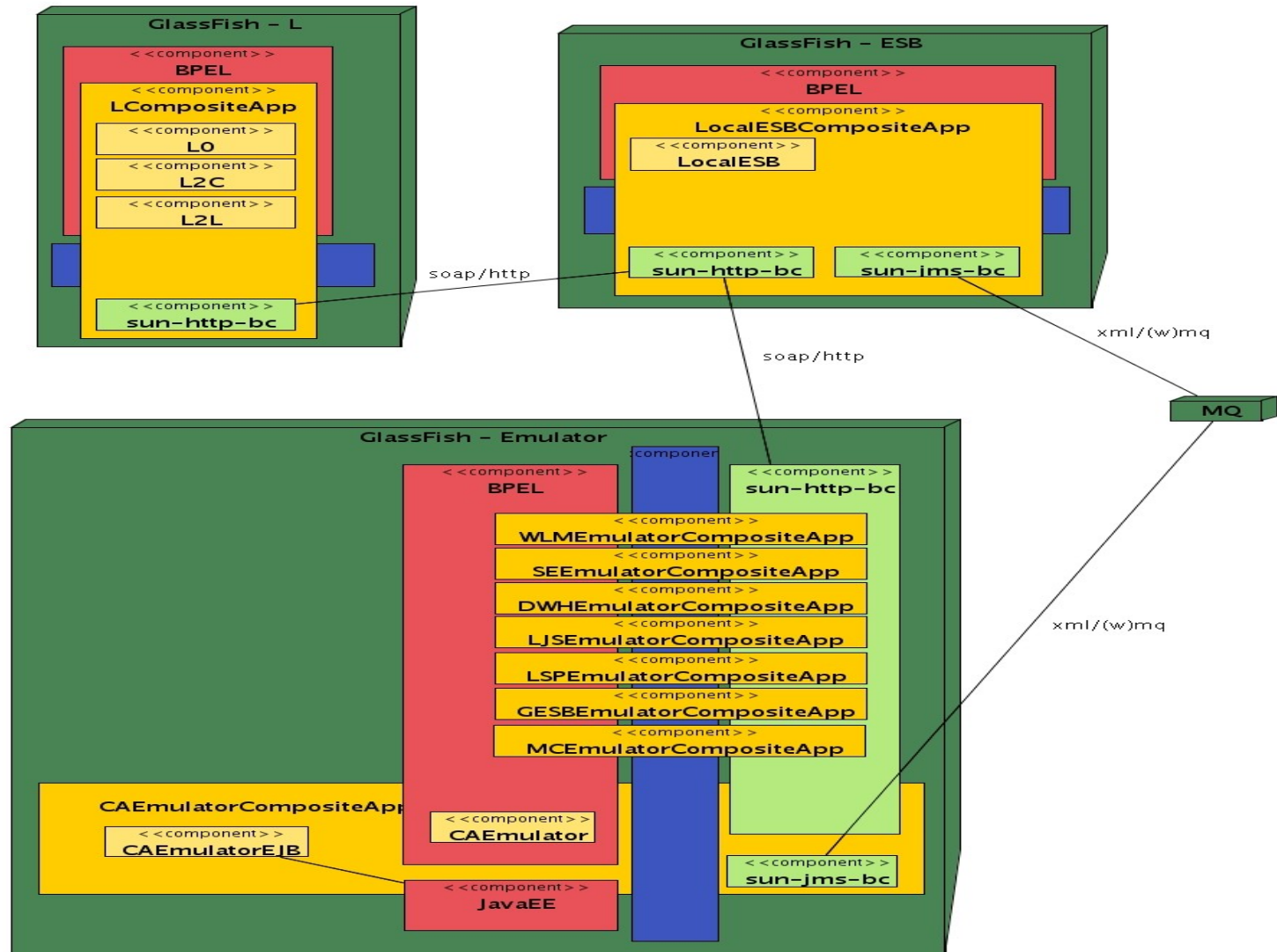
- soapUI
- Hulp Profiler

## > Custom solution

- Code coverage measurements

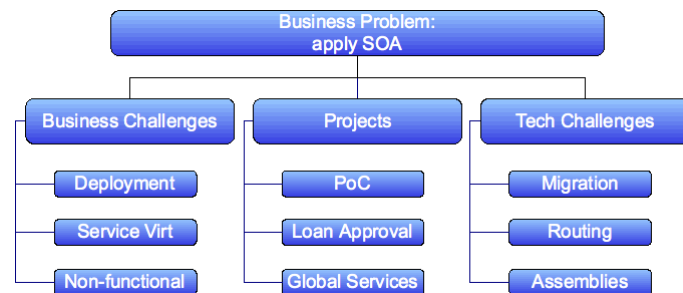


## Demonstration



# Lessons Learned / Next Steps

- > JBI – standard and open
- > Developer community – *thanks!*
- > Looking forward to JBI 2.0 and Fuji
- > Efficient implementation (2 months, 4 developers)
  - from technical design to testing
- > Ready for mission critical environments
- > Next: End-to-end Security





## Q & A

>



# JavaOne<sup>SM</sup>

# Thank You

Istvan.Molnar, Geza.Simon  
@mail.drotposta.hu

Akos.Jancsik  
@gmail.com

Drotposta Consulting Ltd.

