JA-SIG
Java Architectures
Atlanta 2006

SOA, Web-Services and Student Systems

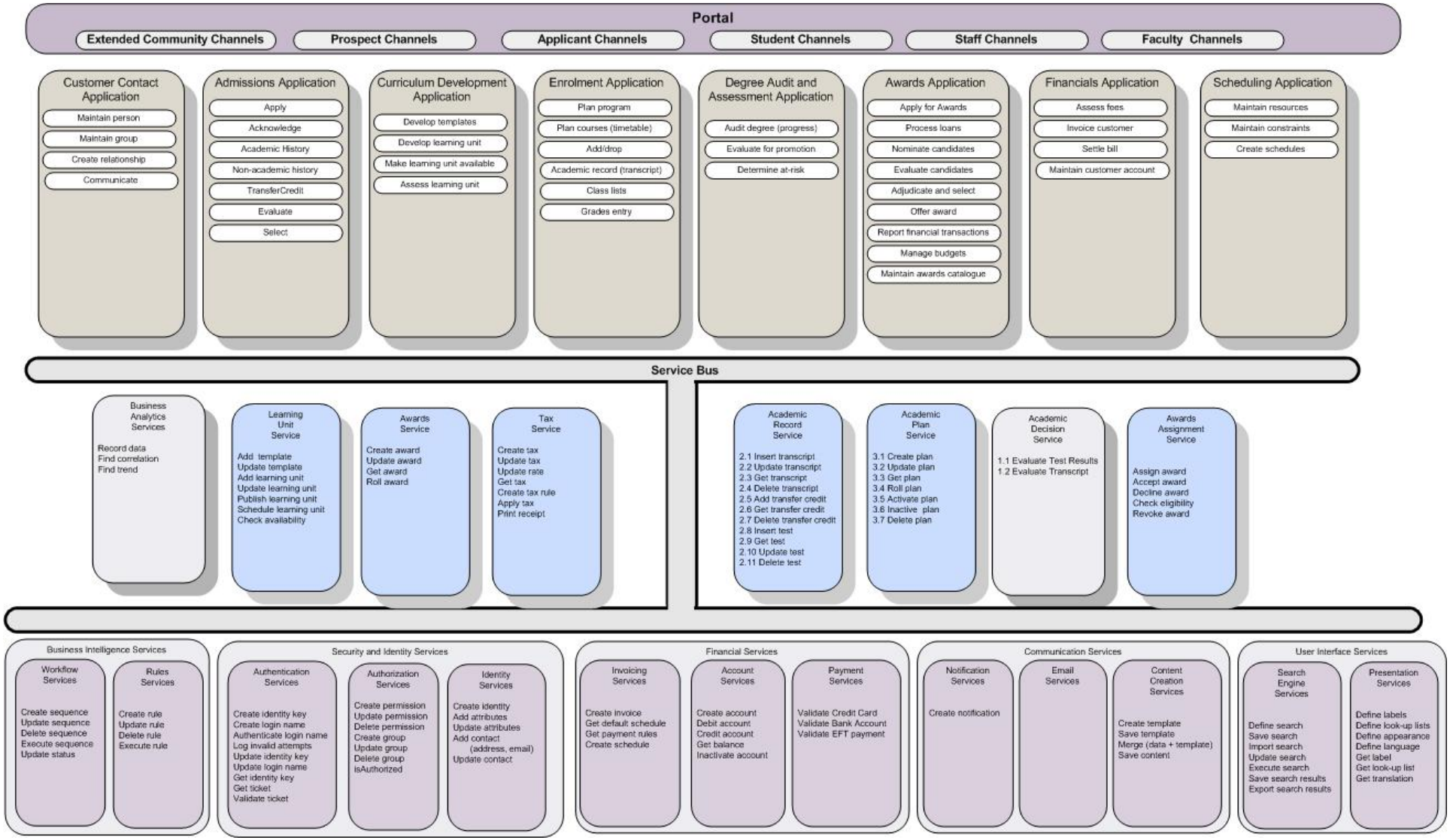
Leo Fernig
University of British Columbia
leo.fernig@ubc.ca

Overview

- The Community Source Student System initiative
  - See http://educationcommons.org/projects/display/CSSSS/Home
- Service Oriented Architecture design issues
- Working with XML
- Web service design paradigms
- Web service deployment issues
- Future directions

JA-SIG Java Architectures Atlanta 2006

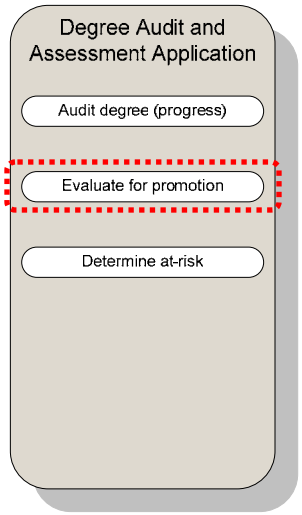
SOA, Web-Services and Student Systems



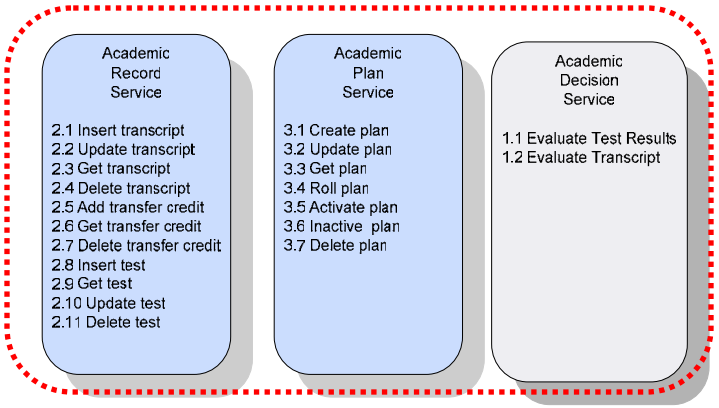
SOAAD issues

- In SOAAD (Service Oriented Architecture Analysis and Design) there is a real tension between traditional top-down approaches and contemporary agile approaches
- The maturity of web service technologies
- The maturity of open source WS components

A specific example



The process: end-of term (or session) evaluation for promotion to the next level (or phase) of an academic program



The **business agnostic services** that support the process:

1. Supplying the student's academic record
2. Applying some evaluation rules
3. Plan the next level (or phase)

Working with XML

- Design issues
- Integration with industry schemas

XML: design issues

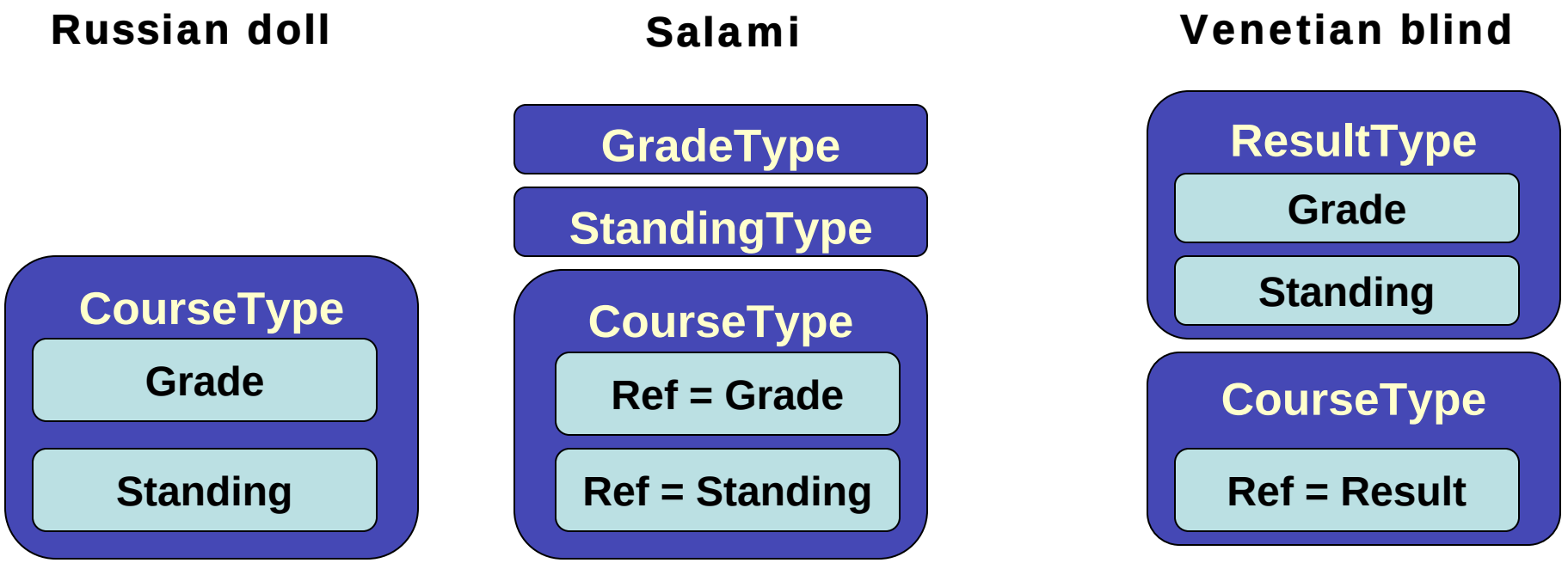
- Flexibility of XML schema
  - Inheritance
  - Composition
  - Cardinalities
  - Ranges of values (eg country codes)
- Verbosity

XML: design issues

- XML-Java binding
  - Flexibility
  - Performance
  - JiBX binding framework (http://jibx.sourceforge.net/)
- Governance and management
  - Name spaces
  - Naming conventions
  - Versioning
- Doc/lit versus RPC
  - Coarse grained interfaces
  - Hiding implementation details
  - Stateless
  - Emphasis on design

XML: design issues

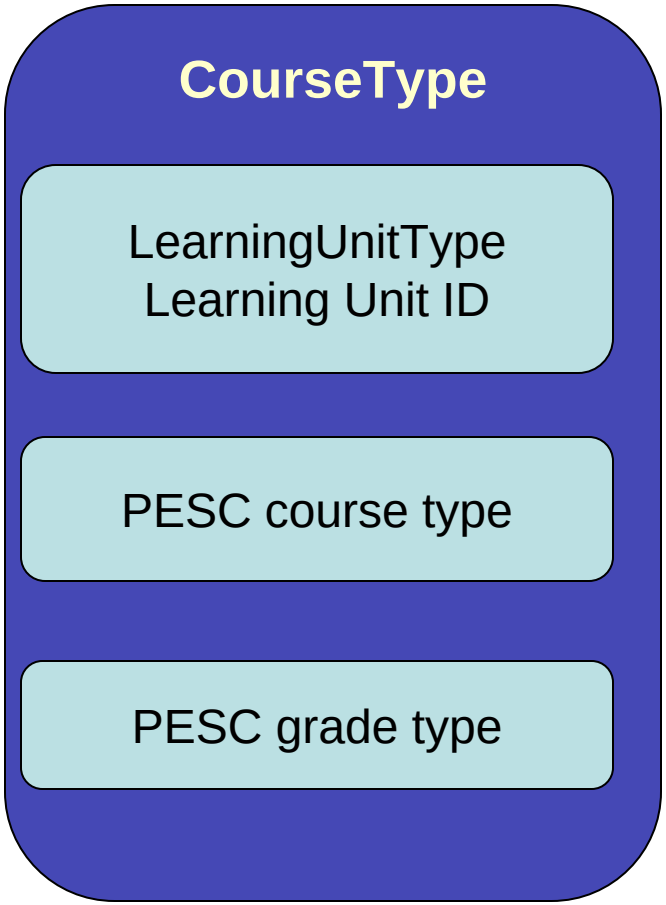
- **Design patterns**
  - **Russian doll vs Salami vs Venetian blind**
  - **http://www.xfront.com/GlobalVersusLocal.html**



XML: Integration with industry standards

- Integrating “local” and “global” commerce
  - Receiving high-school transcripts
  - Trading post-secondary transcripts
  - Receiving test scores (SAT TOEFL etc)
- PESC (Post Secondary Education Standards Council)
  - http://www.pesc.org/
- IMS global
  - http://www.imsglobal.org/
- Alignment strategies

XML: Alignment strategies



- Use the Venetian blind design pattern
- Create a new container object
- Include the learning unit
- Include the PESC types

XML: design issues
REST and SOAP

- **REST: Representational State Transfer**
  - **Flexible**
  - **Simple**

- **Example**

REQUEST: http://www.parts-depot.com/parts

RESPONSE:

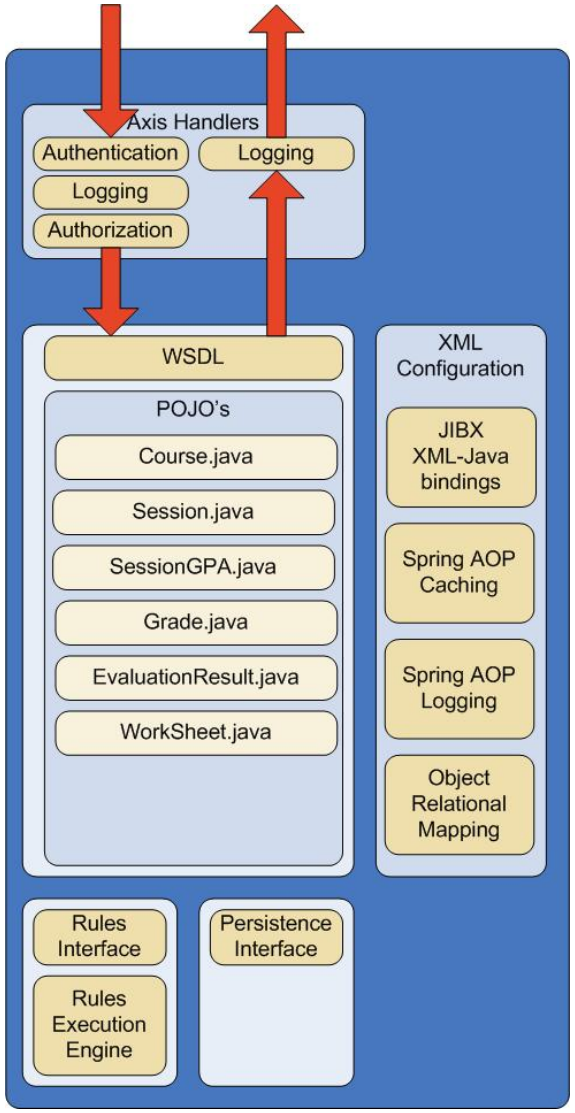
```
<p:Parts
    <Part id="00345" xlink:href="http://www.parts-depot.com/parts/00345"/>
    <Part id="00346" xlink:href="http://www.parts-depot.com/parts/00346"/>
    <Part id="00347" xlink:href="http://www.parts-depot.com/parts/00347"/>
    <Part id="00348" xlink:href="http://www.parts-depot.com/parts/00348"/>
</p:Parts>
```

XML: design issues

REST and WSDL/SOAP

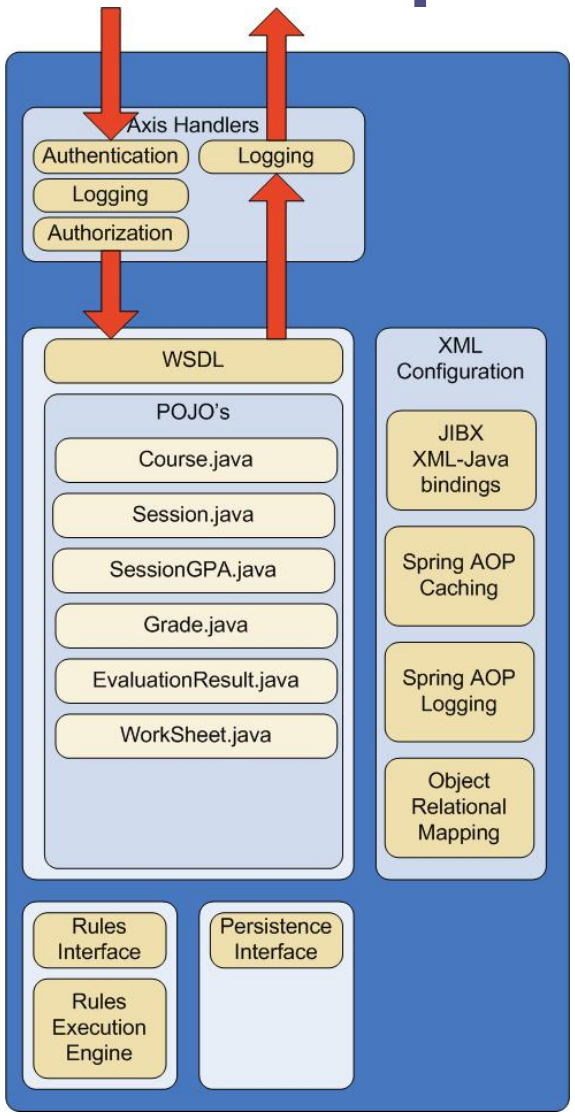
- **REST: Representational State Transfer**
  - Flexible
  - Simple
- **Disadvantages**
  - Service contracts are opaque
  - Flow of control is opaque
  - Not self-documenting
- **If some services are exposed as REST**
  - Will have to be over and above WSDL's
  - Useful for simple "one of" implementations

A process agnostic service

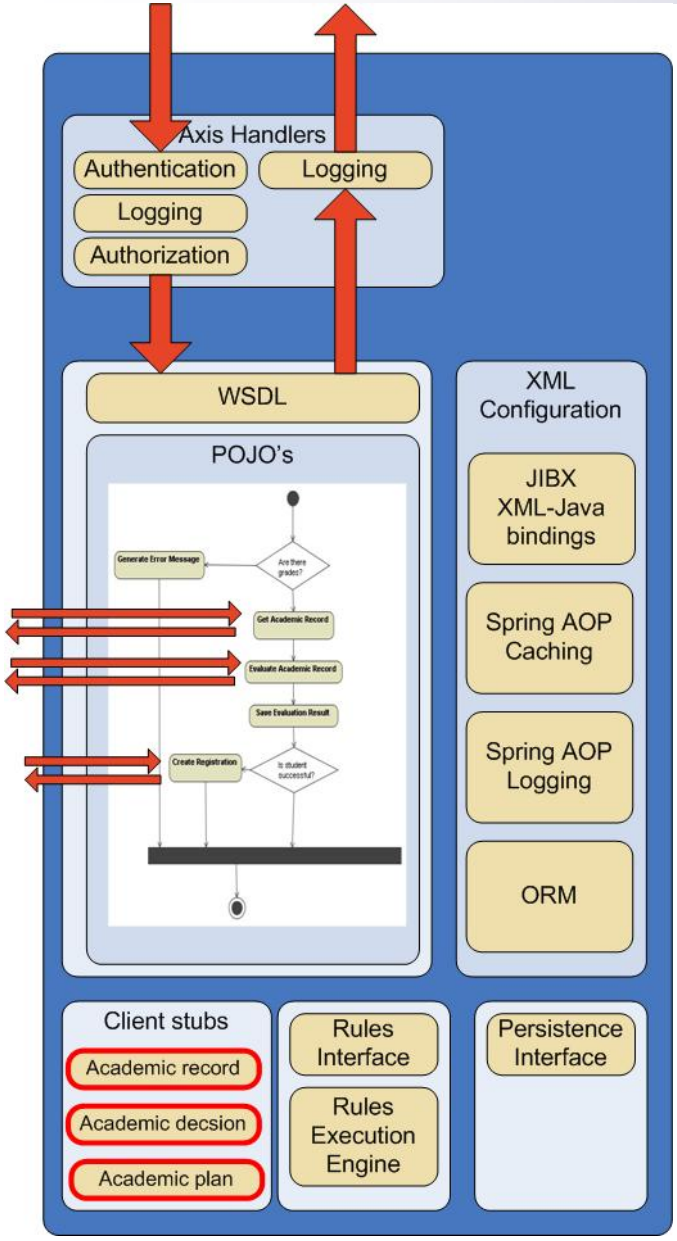


- Handlers for processing headers
  - Security
  - Message logging
- XML – java binding
  - Flexibility and performance
  - Intelligibility
- Spring AOP (isolate housekeeping)
  - Caching
  - Logging
- Object Relational Mappings
- Local services

A process agnostic service: issues



- The need for a standard template
  - WS standards are very flexible
  - Do not want to re-invent infrastructure
  - Allow developers to concentrate on business logic
- Managing XML files
  - Preponderance of XML
- Global vs local objects
  - Canonical XML = global objects
  - Local objects do not need schemas



A business process service: Orchestration

- The core of agility and flexibility in SOA
- The differences between this and a business agnostic service:
  - It contains the logic that expresses a business process
  - It consumers other services

Orchestration

- Hand coding processes
- Using a BPEL (Business Process Execution Language) engine
- Workflow
- Enterprise Service Bus

Performance

There will be performance problems to solve. But, we can…

1. Optimize deployment configurations. E.g. put services behind http load balancers with SSL accelerators.

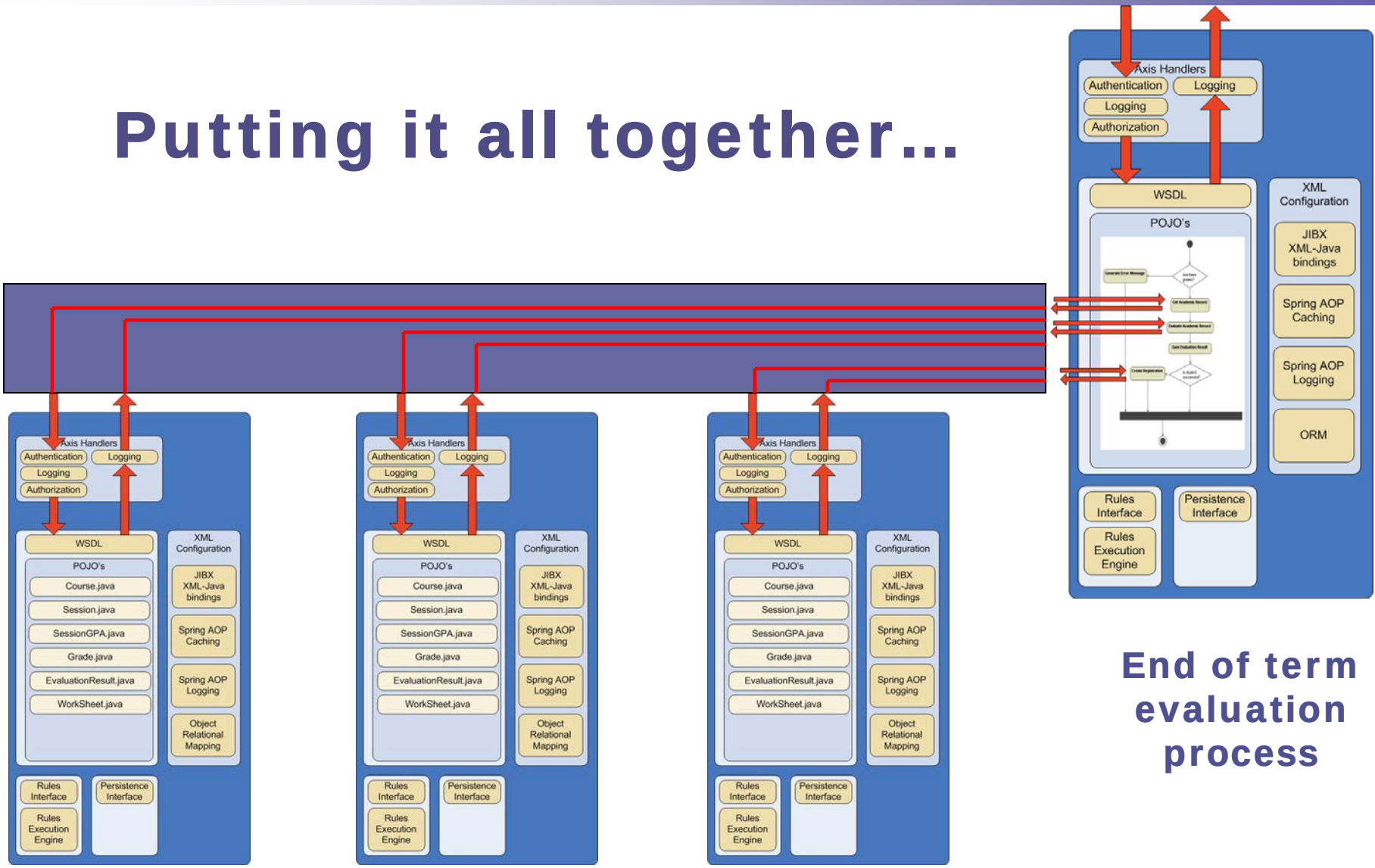
2. Package operations in a service with a view to minimizing traffic

3. Use doc/lit to minimizes traffic

And the predictions are that Moore's law will now hold between 2010 and 2030.

SOA, Web-Services and Student Systems

Putting it all together…



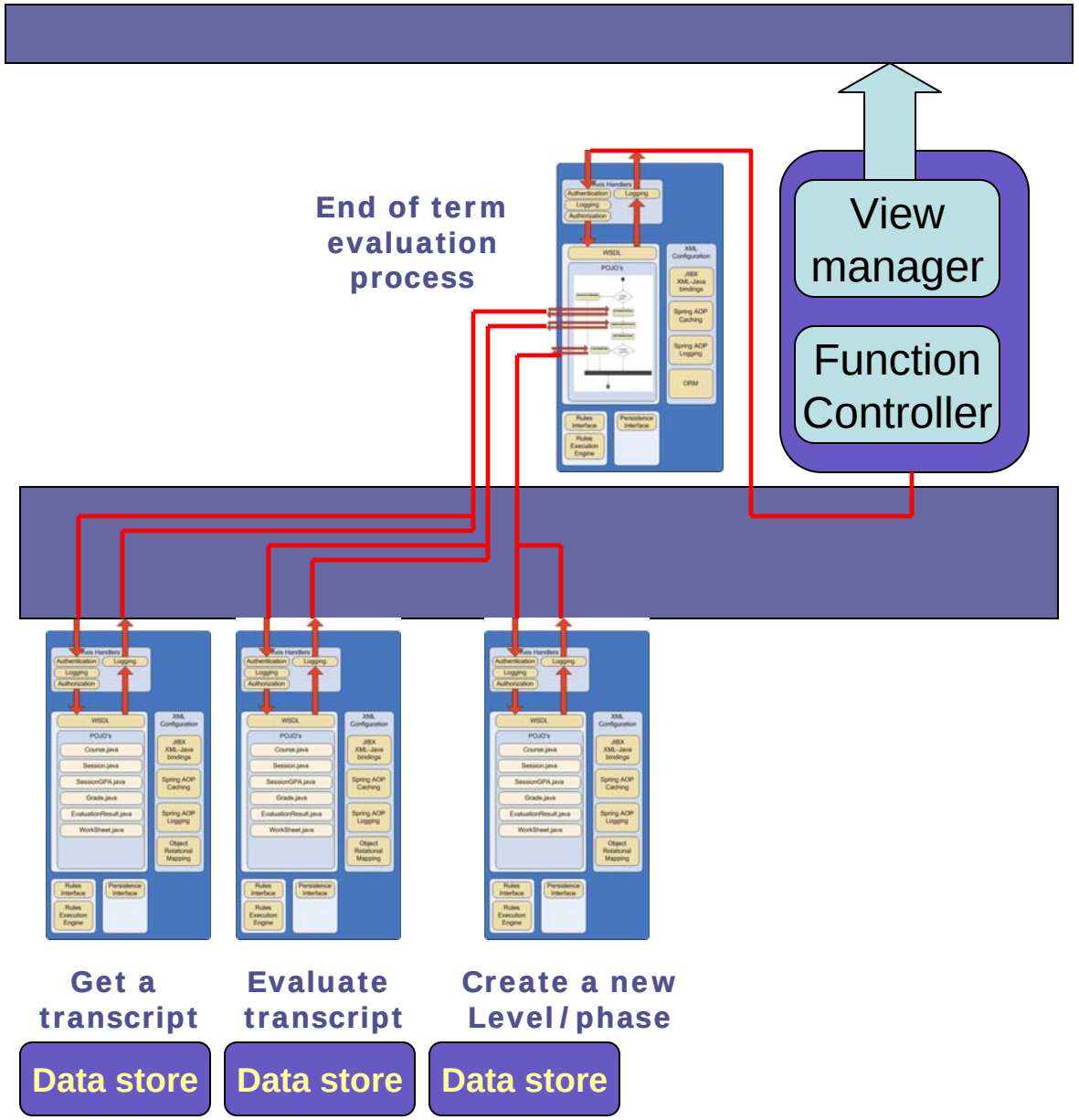
End of term evaluation process

Get a transcript

Evaluate transcript

Create a new Level / phase

SOA, Web-Services and Student Systems



Portal

Business Processes (orchestration)

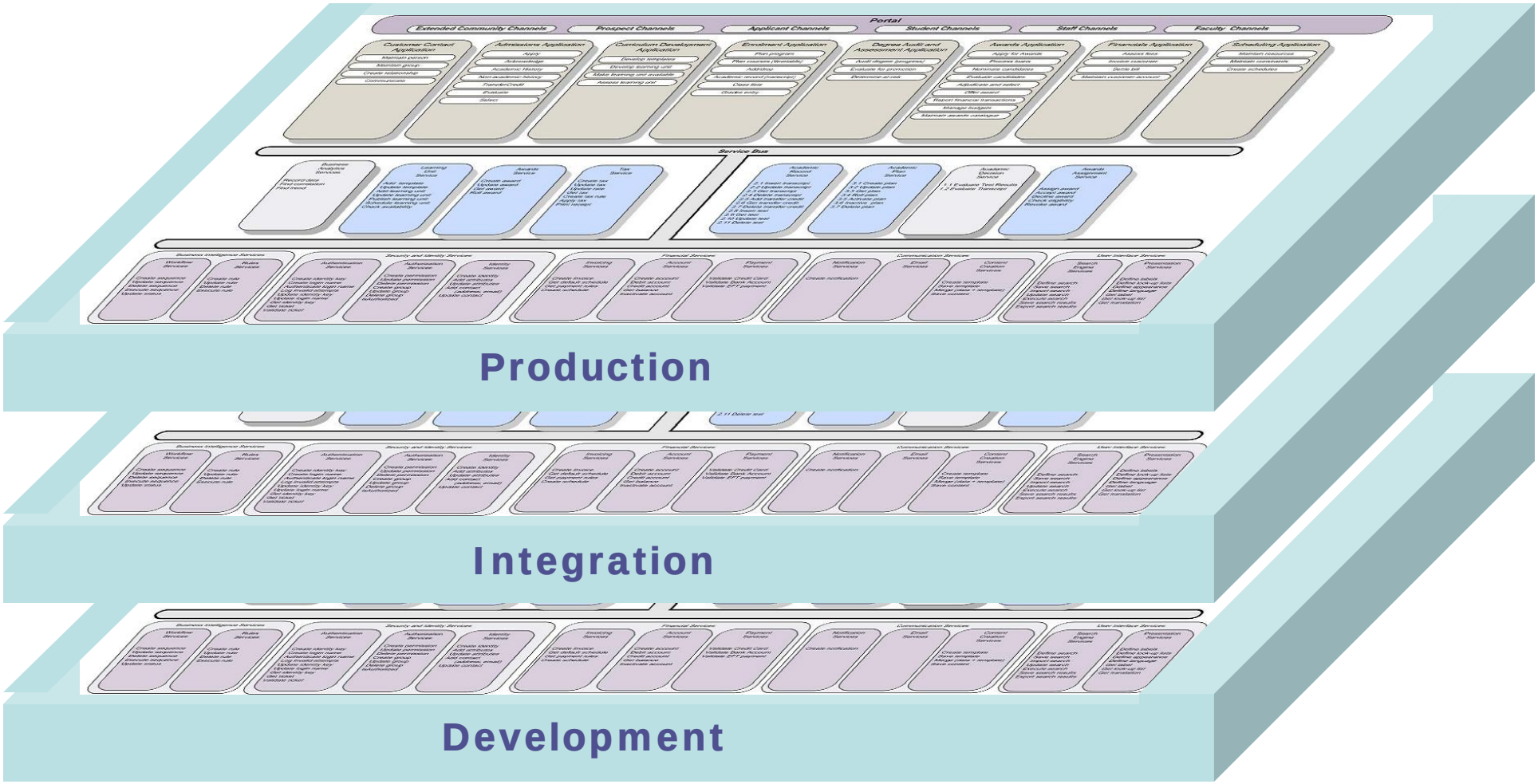
Bus

Business Agnostic Services

Data

SOA, Web-Services and Student Systems

Deployment: managing a service ecology



The future

- Process agnostic systems
- Rule agnostic systems
- Highly flexible and robust deployments
- Intelligent systems that modify there own rule base