Cornerstone Framework

A Open Source Platform for Software Mass Customization

Agenda

- Demo
- Problems and Solutions
- The Cornerstone Solution
- Source Code Walk-through
- Status
- Conclusion

Demo

- Portal
  - Drag-n-Drop Layout Manager
  - Delayed Rendering
- Portal Admin Tool
  - JMX
  - DHTML Interface

Philosophy

Solve 80% of the Total Problem with 20% of Total Effort

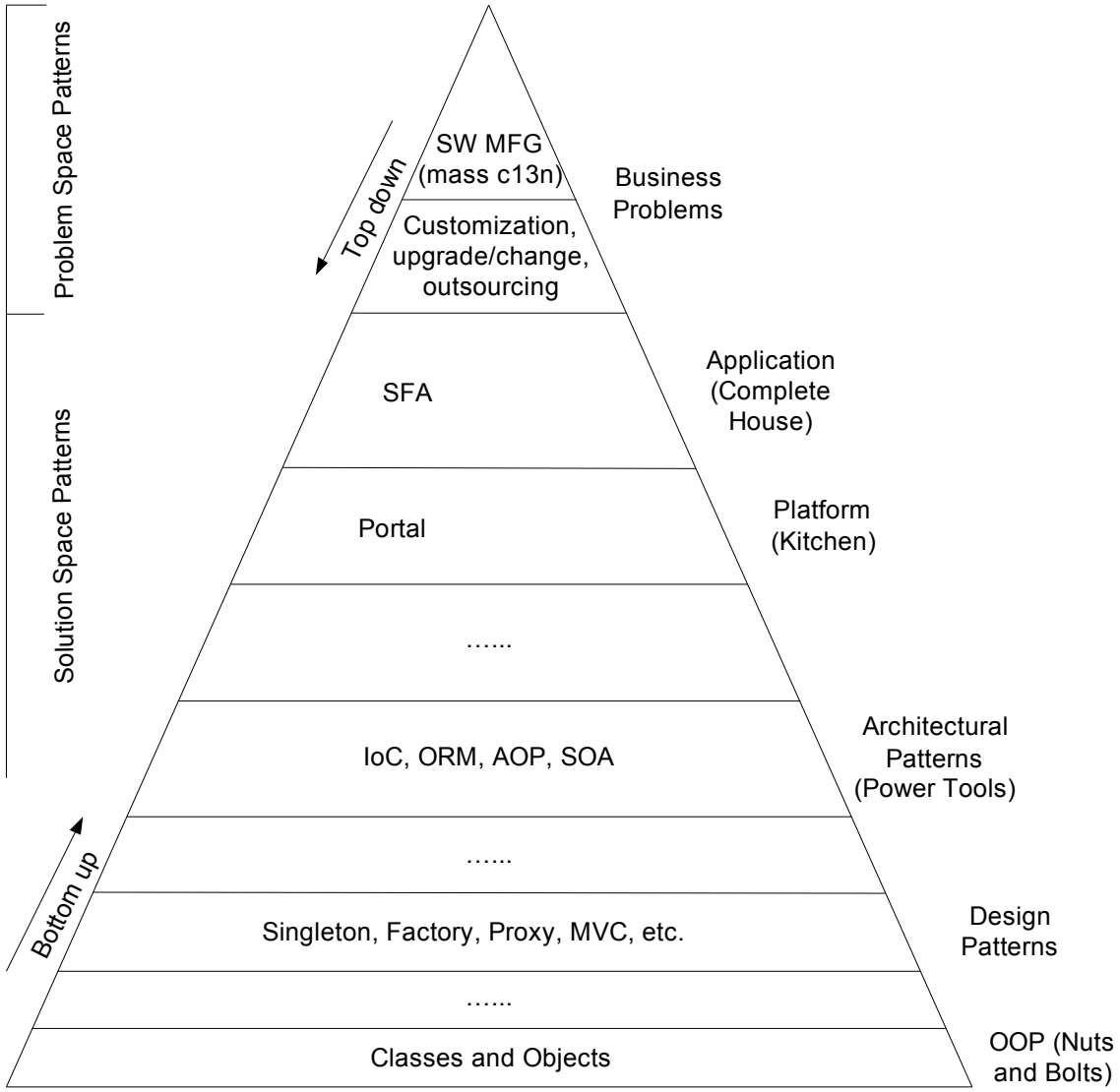
The 3rd Wave: Alvin Toffler

- The 1st Wave Society: Agricultural
  - Low Volume Manufacturing
  - Mom and Pop Shops
- The 2nd Wave: Industrial
  - High Volume Manufacturing
  - Mass Production
- The 3rd Wave: Informational
  - High Volume Manufacturing
  - Mass Customization

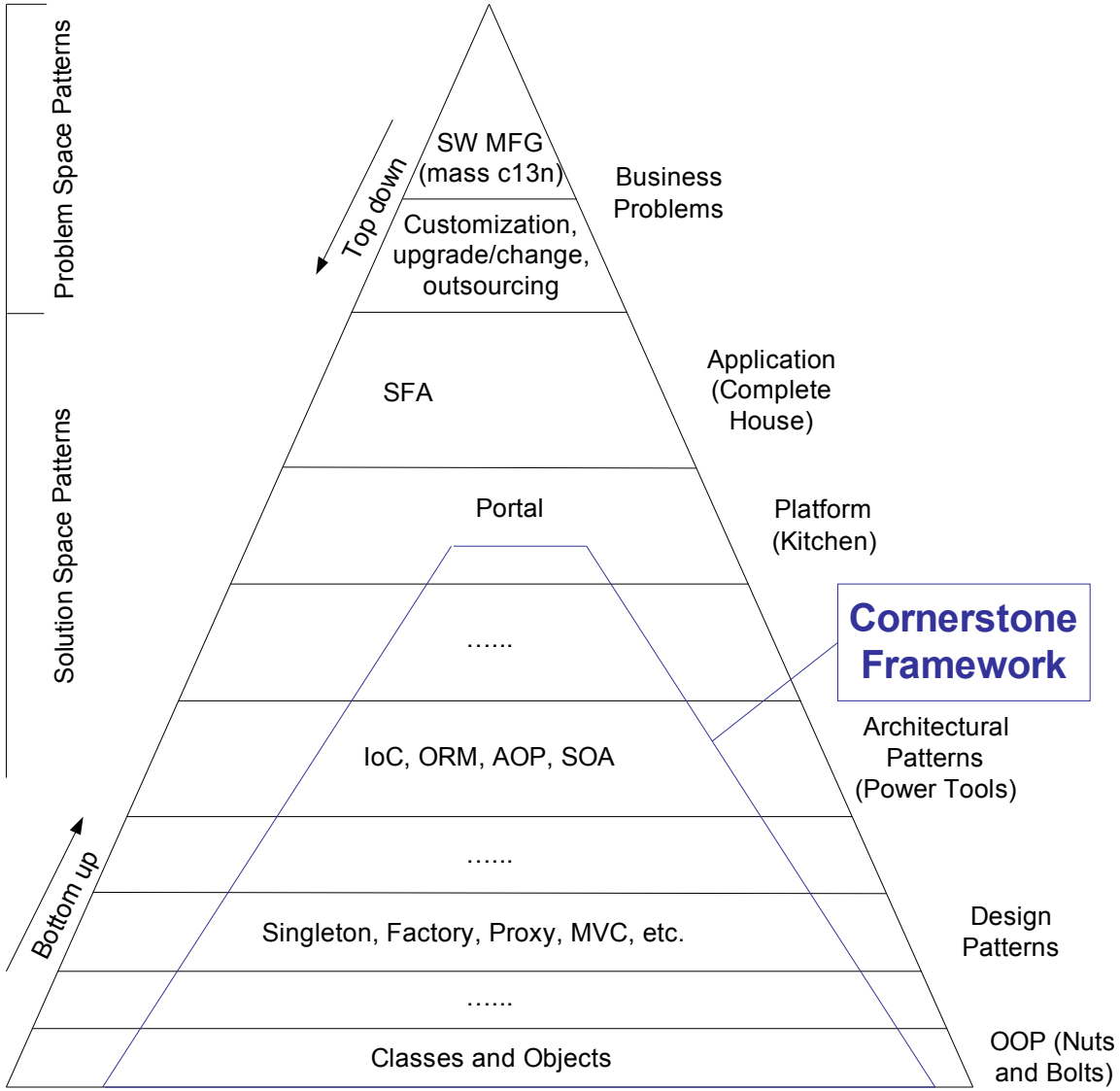
The Software Manufacturing Problem

- **Easy for Mass Production**
  - **Same copies**
- **Difficult for Mass Customization**
  - **Slightly different copies**
  - **No BTO But Rebuilt-to-Order**
  - **No JIT But Just-in-a-Long-Time**

Problems and Solutions



The Cornerstone Solution



Fractal Properties

- **Self Similarity**
  - ***"It seems that the compromise between stability and diversity is served by there the same amount of structure at all scales."*** **Tim Berners-Lee**

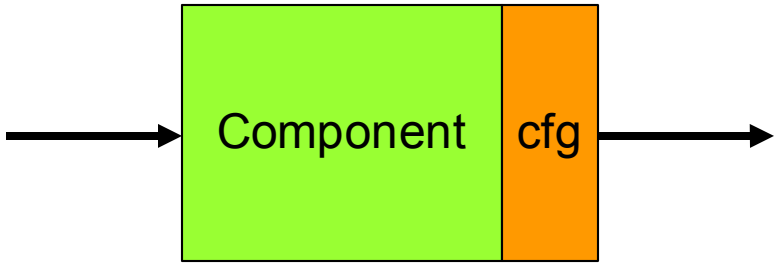
Design Goals

- Configurability/Flexibility/Pluggability
  - Software Mass Customization
- Usability
  - Developer
    - Do-Nothing Caching
    - Do-Nothing JMX Integration
  - End User
    - DHTML

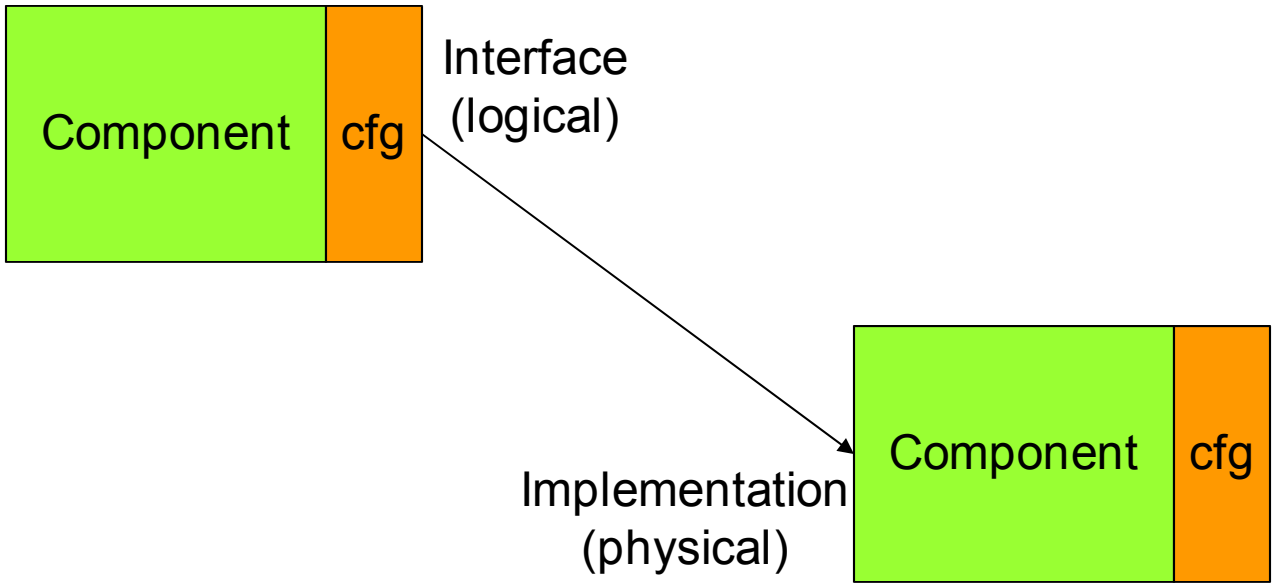
Mass Customization

- **Ability to Meet New Business Requirements with Minimal Impact on Existing Code Base**
- **Customizations in All Dimensions**
  - **Component Level (1D)**
  - **Relationship Level (2D)**
  - **Control Flow Level (3D)**
  - **Preservation of Customizations (Time)**

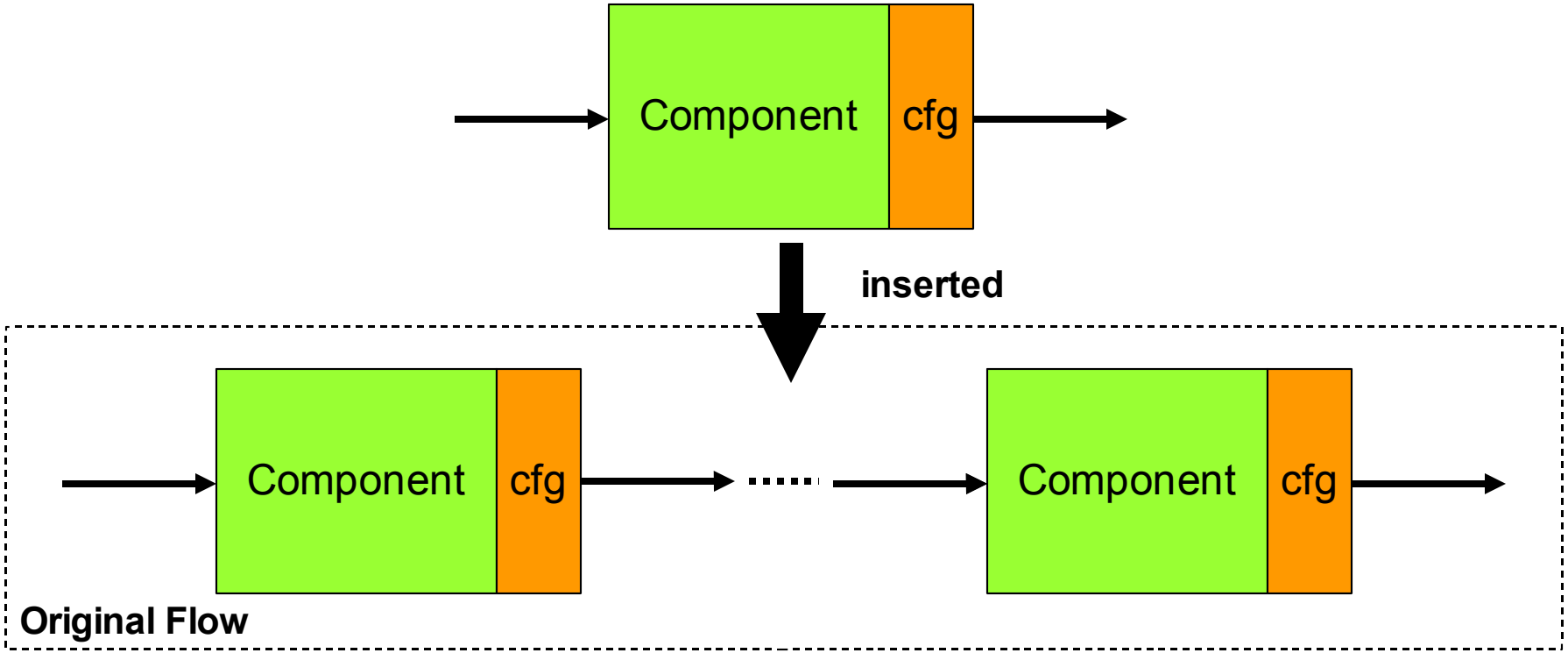
Component Level Customization



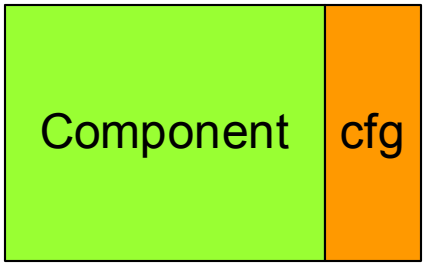
Relationship Level Customization



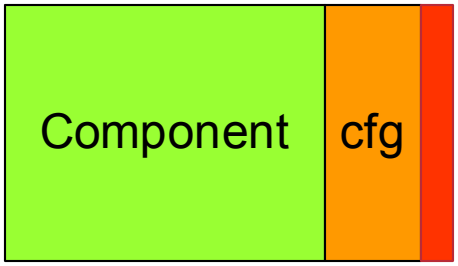
Control Flow Customization



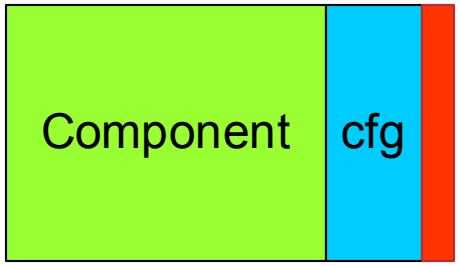
Preservation of Customization



Vendor Out of Box



Customer Customized



Upgraded with Preservation

Major Contributions (Jetspeed)

- Now
  - Drag-n-Drop Layout Managers
  - Delayed Rendering of Portlets
  - StraightDbPsmlManager
- Future
  - DHTML Interface (Individual Refreshes)
  - Inter-Portlet Communications
  - …

Major Contributions (Framework)

- Service Framework
- Out-of-Box JMX Integration
- MVC (Actions + DHTML)
- Asynchronous Logger
- ORM, Pub-Sub, Job-Queue-Worker, …

Major Contributions (Service Framework)

- Service Registry
- Service Validation
- Logical Service
- System Management (JMX)
  - Logging (parameters)
- Service Caching
- Service Flow Control

Solution Space Patterns: Configurability

- Individual Classes
- Relationships between Classes
  - Inversion of Control
- Control Flows
  - Cornerstone Unique

Solution Space Patterns: Configurability

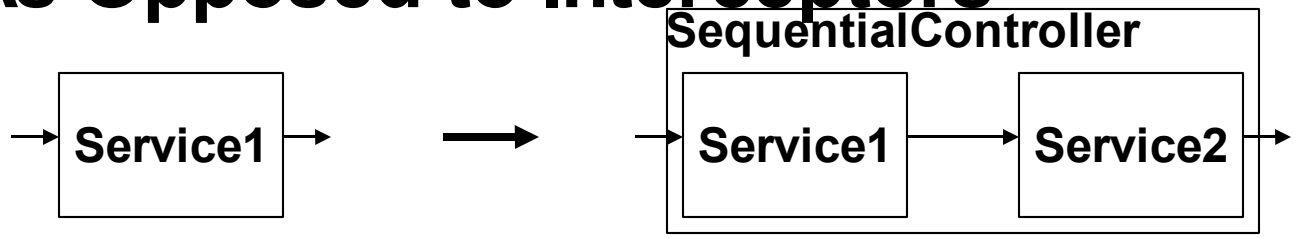
- **Individual Classes**
  - **Example: Logical Services**
    - **LS1: Data source “esalesdb” ⟶ “DS1”**
    - **LS2: Data source “esalesdb” ⟶ “DS2”**

Solution Space Patterns: Configurability

- Relationships between Classes
  - No Class Names Anywhere
  - Inversion of Control
  - Example: JMX Integration
    - IServiceMetric IService.getMetric();
    - BaseService:metric.className=BaseServiceMetric
    - MyService:metric.className=SpecialServiceMetric

Solution Space Patterns: Configurability

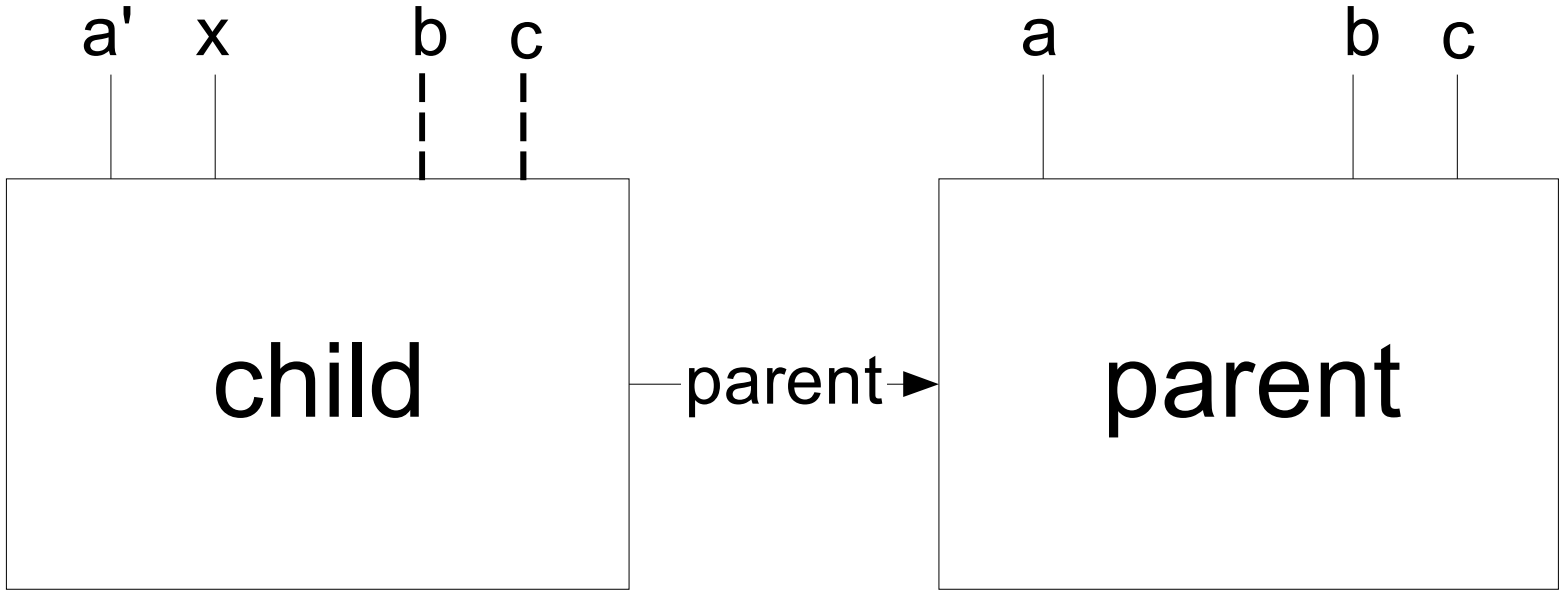
- Control Flows
  - SequentialController
  - AsynchronousController
  - SwitchController
  - …
- As Opposed to Interceptors



Solution Space Patterns: Indirection

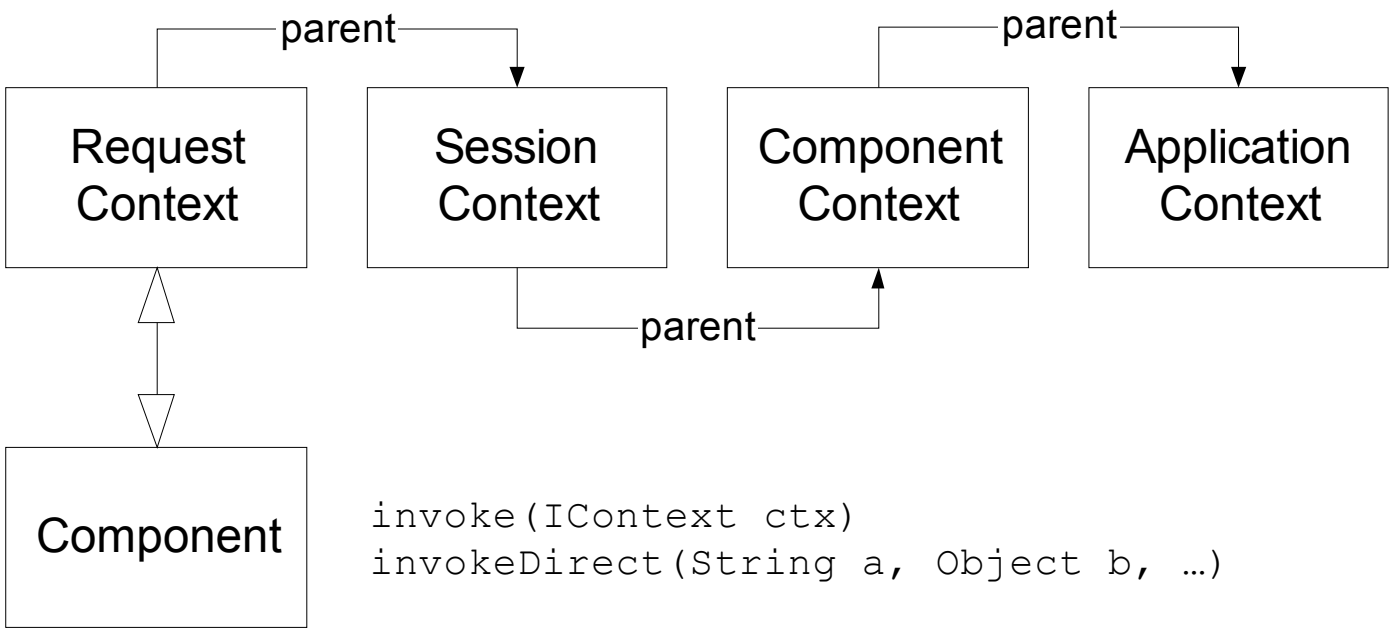
- **Indirection: Logical to Physical Mapping**
  - **Called Extension Points in Eclipse**
  - **Example: All services in registry are Logical Services**
  - **Registry can easily move between database and properties files**

Solution Space Patterns: Overwrite



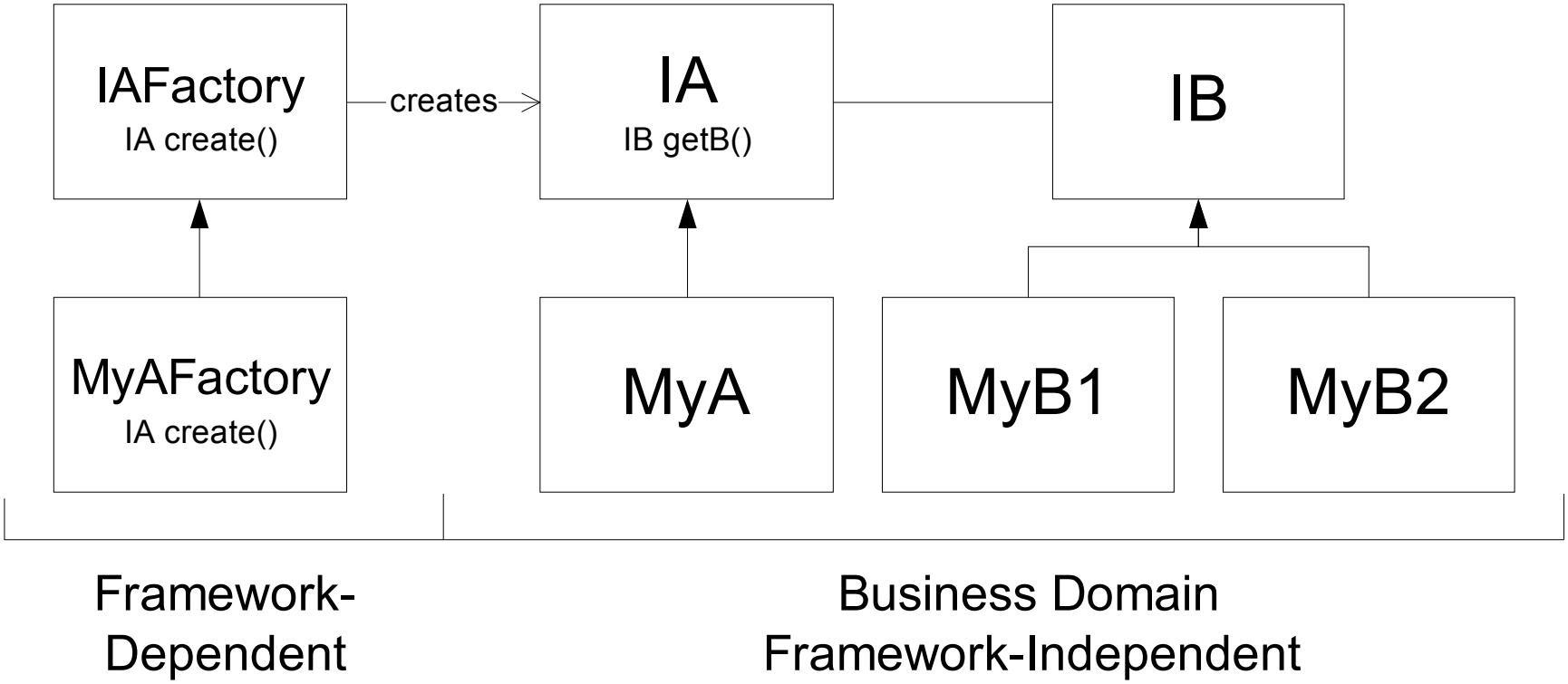
- **Example: Logical Service based on another Logical Service**
- **Supports instances that only differ in configuration**

Solution Space Patterns: Overwrite



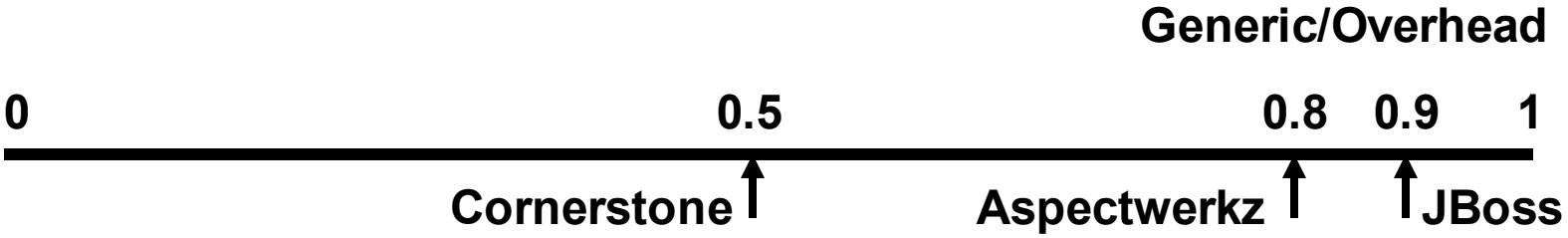
- **Configuration (static) and parameters (dynamic) handled the same way**
- **Different scopes handled the same way**

Solution Space Patterns: Non-Intrusion



Framework Comparison

- **Where Cornerstone Stands out**
  - **Vision, Depth, Breadth**
  - **As opposed to a Point Tool with BYOM**
  - **AOP the Cornerstone Way**
    - **inovkeStart/Middle/End, *Service Controllers***
    - **Can integrate best 3rd-party easily**



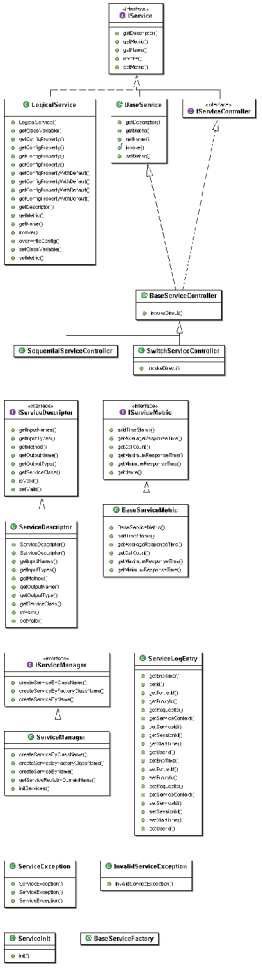
Source Code: Drag-n-Drop

- portal/cppext/template/jsp/controllers/html/x-multicolumn-customize.jsp
- org.apache.jetspeed.modules.actions.controllers.XMultiColumnControllerAction.java
  - Dependencies
    - JspTemplate.java
    - Util.java

Source Code: Delayed Rendering

- com.cisco.salesit.framework.portal.portlets.BasePortlet.java
- template/jsp/porlets/html/delayedContent.jsp

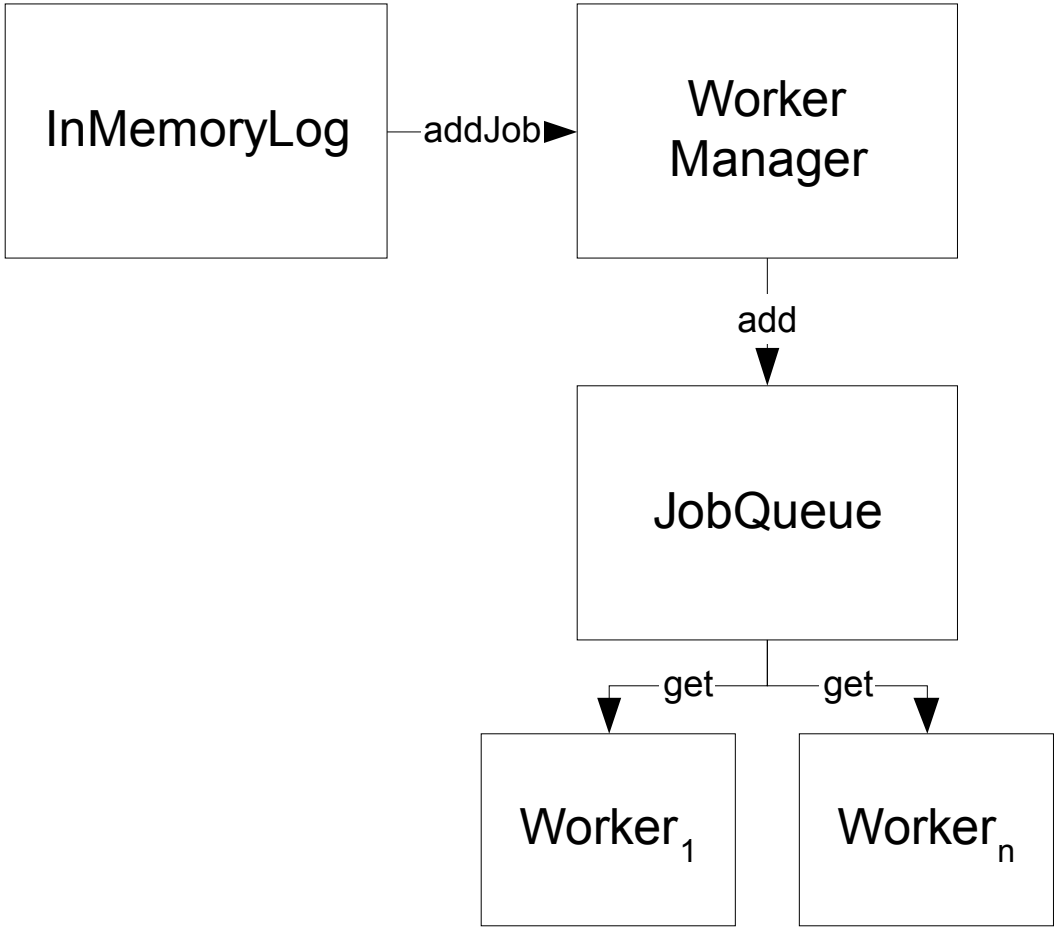
Source Code: Service



Source Code: Service

- DateService.java
- dateService.reg.properties
- timeService.reg.properties
- dateAndTimeService.reg.properties
- BaseService.java
  - invokeStart/Middle/End

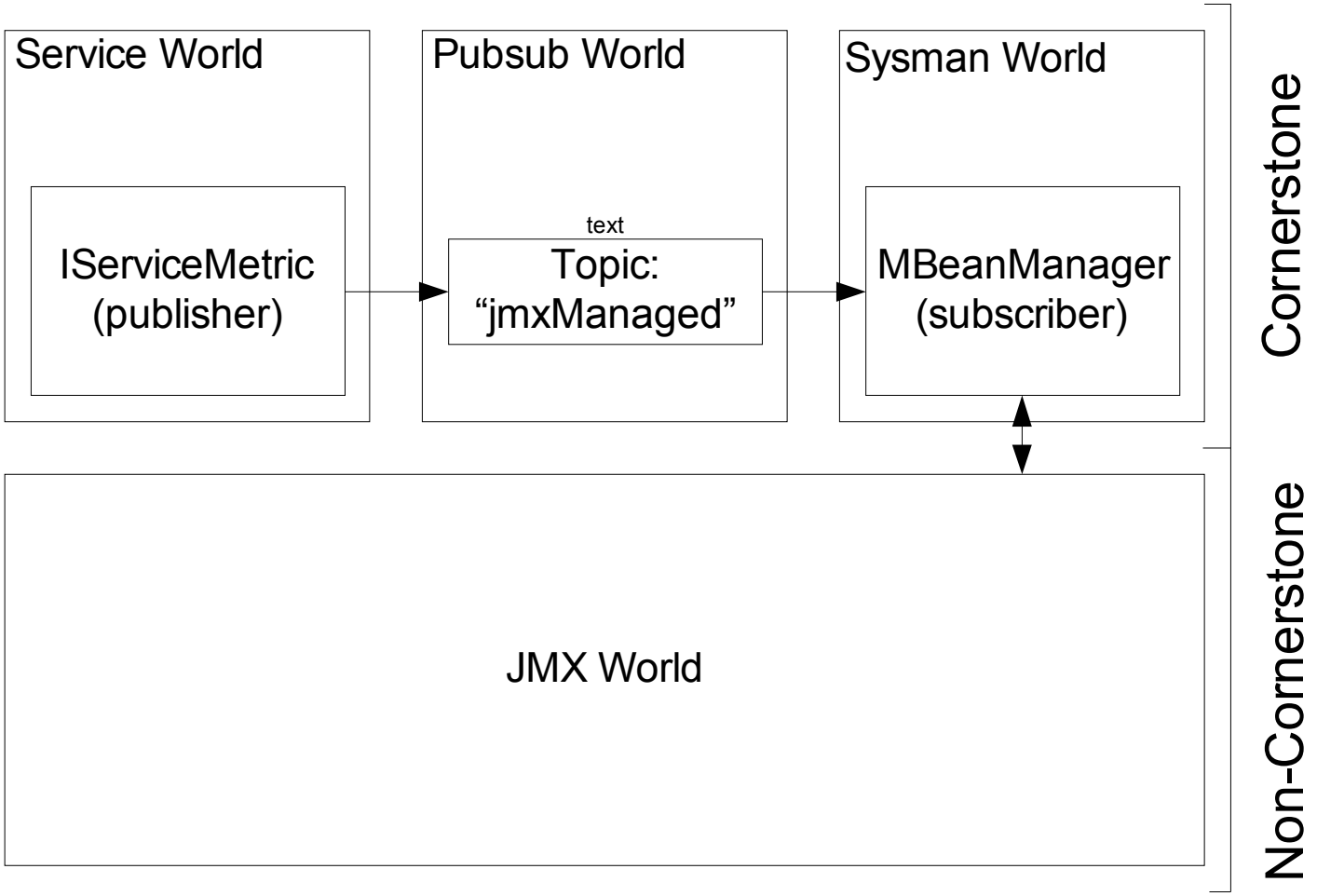
Source Code: Service Logging



Source Code: JMX Integration

- BaseService.properties
- BaseServiceMetric.properties
- BaseServiceMetric.java
- AutoPublication.java
- MBeanManager.java

Source Code: JMX Integration



Status

- Cornerstone Framework source code already open-sourced and checked into Apache Jetspeed CVS
- Development continuing for new features and enhancement
- Contact
  - Jun Yang: junyang@cisco.com

Conclusion

- **Cornerstone is an exciting new approach to software mass customization that modern design patterns and techniques failed to address**
- **We solve the business problem top-down by mapping it to solution space patterns of the Cornerstone Framework**

Spread the Word and Catch the Fire!

Cornerstone Framework