



# *Software Development Asset*

*May 11, 2004*

*Chief Technical Officer  
Daud Santosa  
Department of Interior*





# Reuse Maturity Framework

Dimension of Maturity	Initial/Chaos (1)	Monitored (2)	Coordinated (3)	Planned (4)	Ingrained (5)
Motivation/Culture	Reuse is discouraged	Reuse is noted indifferently reinforced, rewarded	Reuse is incentivized	Reuse is indoctrinated	Reuse is “the way we do business”
Planning for reuse	Nonexistent	Grassroots activity	Targets of opportunity	Business Imperative	Part of a strategic plan
Breadth of reuse Involvement	Individual worker	Work group	Department	Division	Enterprise
Responsibility for making reuse happened	Individual worker	Shared initiative	Dedicated individual	Dedicated Group	Corporate Group
Process by which reuse is leveraged	Development process chaotic unclear where reuse comes in	Reuse questions raised at design review (after the fact)	Design emphasis placed on reuse of off-the-shelf part	Focus on developing families of products	All software products genericized for future reuse
Reuse Inventory (assets)	Salvage yards (no apparent structure to collection)	Catalog Identifies Language-and platform specific parts	Catalog organized along application-specific lines	Catalog includes generic data processing functions	Planned activity to acquire or develop missing pieces in catalog
Classification Activity	Informal, individualized	Multiple independent schemes for classifying parts	Single Scheme, catalog published periodically	Some domain analyses performed to determine categories	Formal, complete, consistent, timely classification

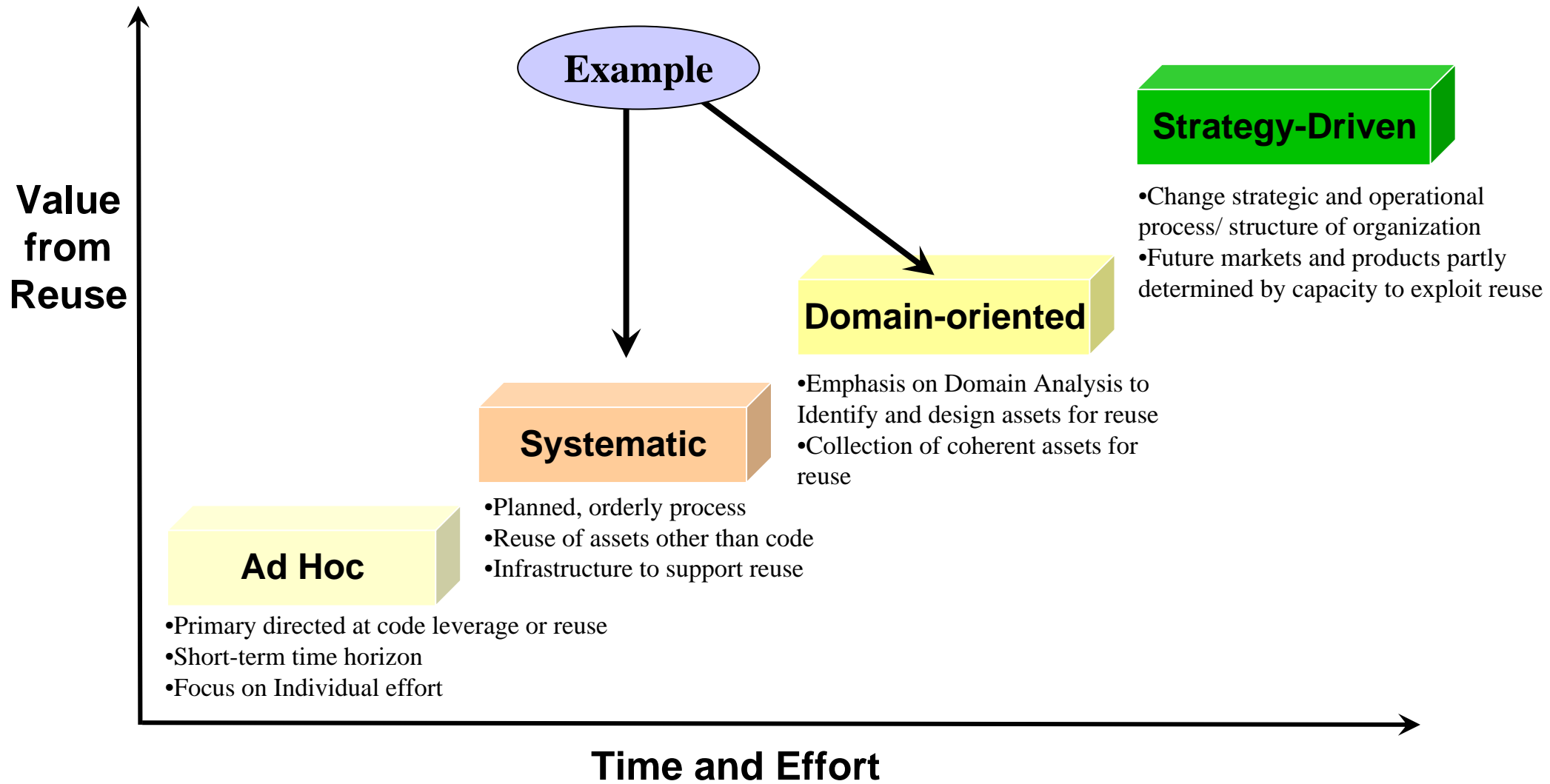


# Reuse Maturity Framework - continue

Dimension of Maturity	Initial/Chaos (1)	Monitored (2)	Coordinated (3)	Planned (4)	Ingrained (5)
Technology Support	Personal Tools, if any	Lots of tools, e.g, configuration management, but not specialized to reuse	Classification aids and synthesis aids	Electronic Library separate from development environment	Automated support integration with development system
Metrics	No Metrics on level of reuse, payoff, or cost of reuse	Number of lines of reused code factored into cost models	Manual tracking of reuse occurrences of catalog parts	Analyses performed to identify expected payoffs from developing reusable parts	All system utilities, software tools, and accounting mechanisms instrument to track reuse
Legal, contractual, accounting consideration	Inhibitor to getting started	Internal accounting scheme for sharing costs, allocating benefits	Data rights and compensation issues resolved with customer	Royalty scheme for all suppliers and customers	Software treated as key capital asset



# Level of Reuse



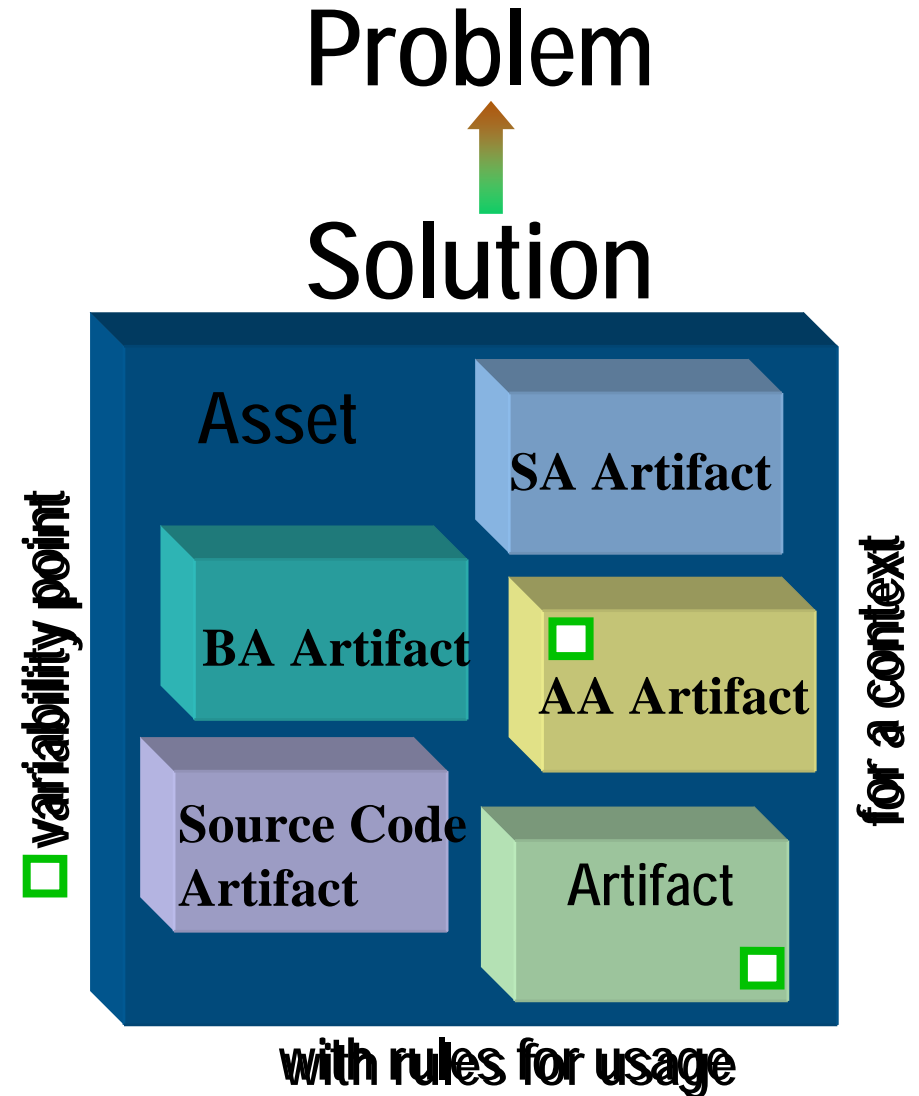


# *What is Software Asset Based Development ?*



# What's an Asset?

- An Asset is
  - *a collection of Artifacts*
  - *which provide a solution to a problem*
    - *for a given context*
    - *with rules for usage*
    - *and variability points*
- What are Artifacts?
  - *Work products from the software process*
    - *Requirements, Models, Source code, Tests, and so on...*
- Kinds of assets
  - *Components, patterns, web services, frameworks, templates, ...*





# What Is Asset-Based Development (ABD)?

- It is developing software solutions (re)using cohesive, documented software artifacts which provide a solution to a problem for a given context
- It is organizing software development in a way that leverages previous investments and influences the nature of future investments



# Component Based Development Approach

Federal Enterprise Architecture Reference Model

Software Architecture Development Life Cycle

Understand –  
apply patterns

Provision

Assemble

Implement

Business  
Model

Component  
Architecture

Acquire

Subscribe

Modify

Wrap

Build

Application  
/Component  
Assembly

Execution

Deployment

Inventory Management

Process Management

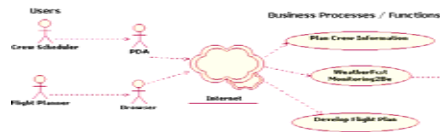
Manage



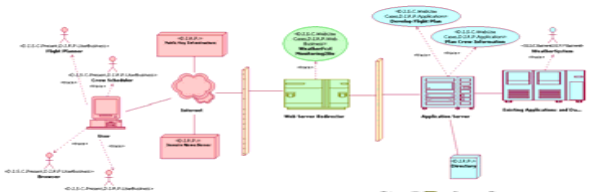


# Component Based Development Approach - continue

## 1. Business Process Modeling



## 2a. Business Process / Solution Architecture Mapping 2.b. Operational Model

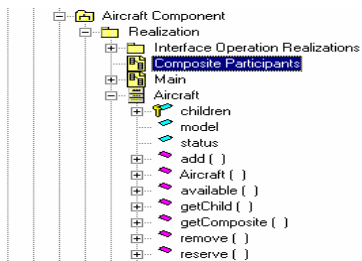


## 3. Application Use - Case Modeling

## 4. Application Behavioral Modeling

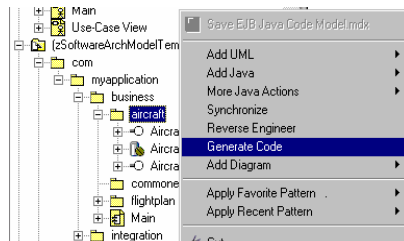


## 5. Component Specification (platform independent)



## 6. Component Design (platform - specific)

## 7. Component Creation (platform - specific)



## 8. Code Generation (platform - specific)

```

package com.application.business.aircraft;
import javax.ejb.EntityBean;
/** @modelid (1F73914-5974-4CF1-8825-2147829145F) */
public class AircraftBean implements EntityBean {
    /** @modelid (71717018-9708-47E7-86AC-4057441320F) */
    public java.lang.String model;
    /** @modelid (4826897F-C281-4937-8838-0480C920A298) */
    public java.lang.String status;
    /** @modelid (AB2CC018-488F-4121-8959-8924C2794D03) */
    public java.util.ArrayList children;
    /** @modelid (3F717F78-74C4-444D-88C2-4977214D4AF) */
    private javax.ejb.EntityContext entityContext;
    /** @modelid (75647025-8A48-4189-4F65-5F773894470D) */
    public AircraftBean() {
    }
    /** @modelid (F314890B-7736-4FE1-4052-23F44F4370C) */
    public void setEntityContext(javax.ejb.EntityContext entityContext) {
        *Begin Template Expression(753F3248-9F44-41A4-4279-7A1144314137)*
        this.entityContext = entityContext;
        *End Template Expression(753F3248-9F44-41A4-4279-7A1144314137)*
    }
    /** @modelid (20744FC7-844E-4C4D-4617-79665733005) */
    public Object ejbCreate() throws javax.ejb.CreateException {
        *Begin Template Expression(5807FE1D-4569-4FFA-888D-9E398D1E176)*
        return null;
        *End Template Expression(5807FE1D-4569-4FFA-888D-9E398D1E176)*
    }
}

```



# Software Reuse Asset-Based Development Framework

## EA Reference Model

Business reference model  
Service component reference model  
Technical reference model  
Data reference model  
Security reference model

### Business Solution Architecture Workflow



1. **Primary:** BA; **Assistants:** SA  
2. **Primary:** SA; **Assistants:** BA



Solution Architecture

### Application Family Architecture Workflow



3-8. **Primary:** AFA; **Assistants:** SA  
(scope is cross-application)



Application Family Architecture

### Application Architecture Workflow



3-8. **Primary:** AA; **Assistants:** SA, AFA  
(optional), Development Team (scope is  
application-specific)



Application Architecture

### Application Development Workflow



6-8. **Primary:** AD; **Assistants:** AA,  
Development Team



Application

Core Asset Development Workflow

Core Assets



Core Asset  
Producer, AR,  
RC



Asset Certification Workflow

AR, Reuse Coordinator



Asset Library

## Asset Management



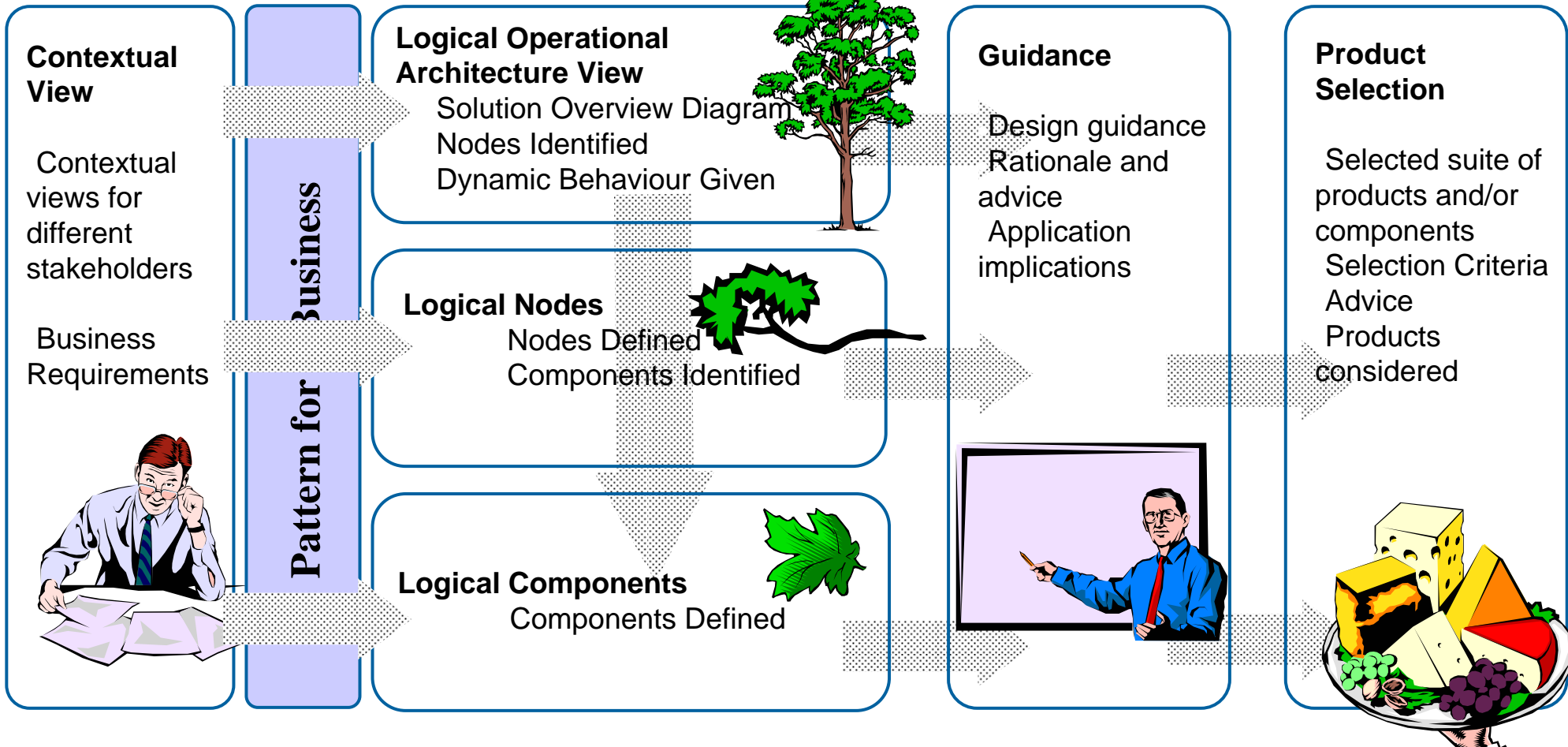
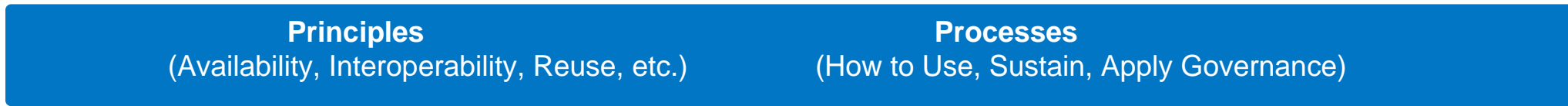
Reuse  
Consumer

## Reuse Management

Policies / Procedures / Guidelines



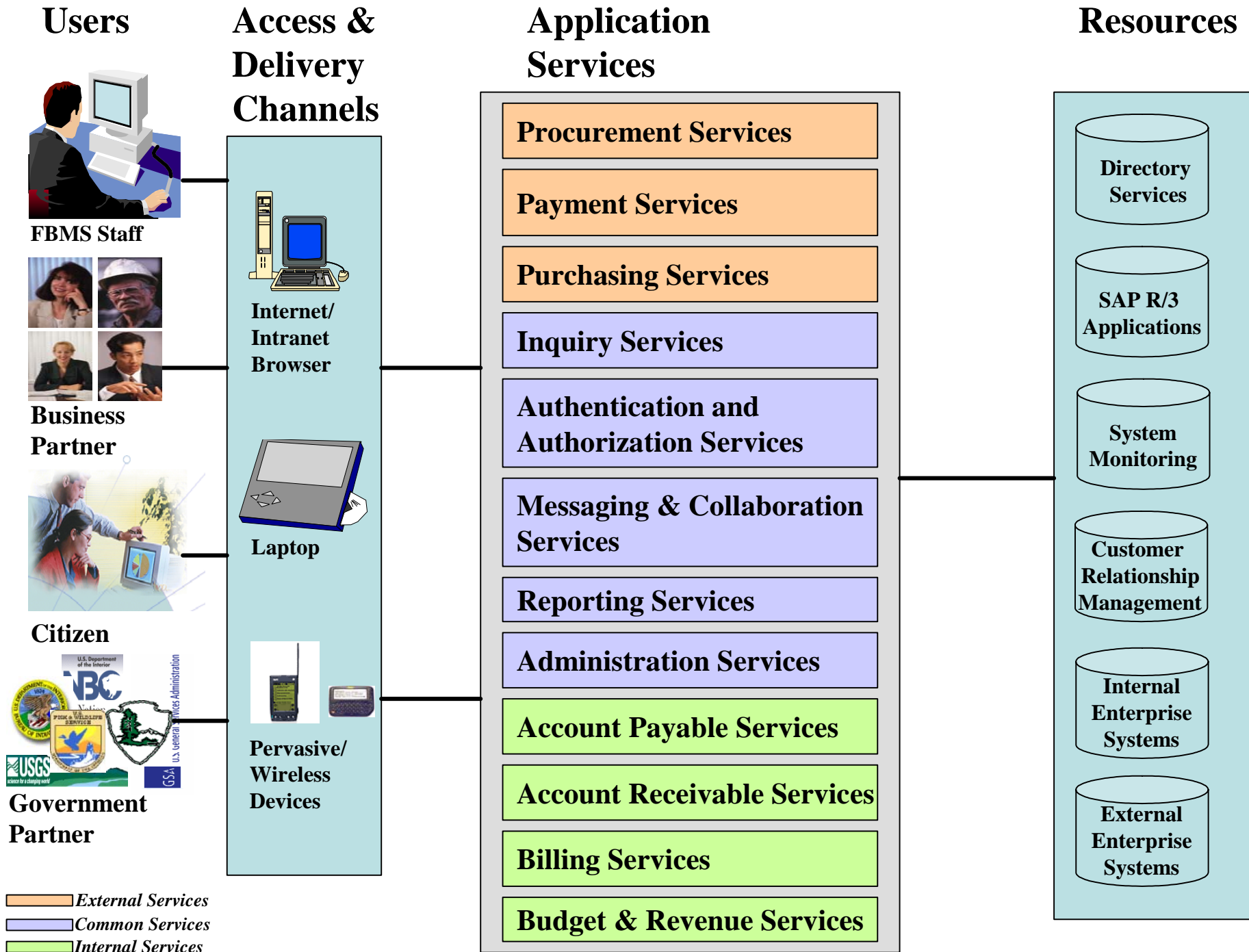
# When defining an architecture artifacts it is often useful to structure the Architecture as shown



## Glossary Acronym and Term Definitions

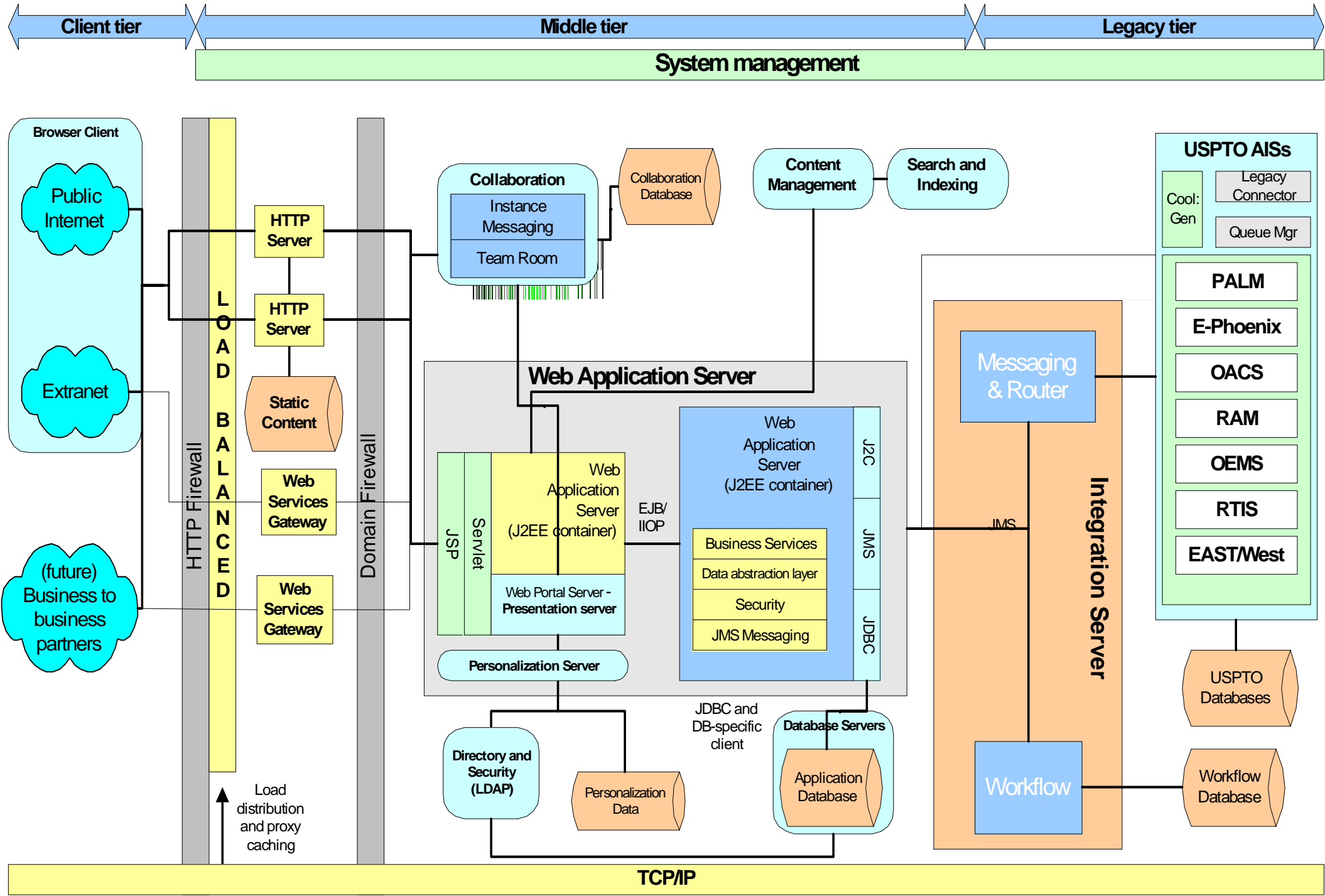


# High-Level View of Business Services for FMBS Initiatives



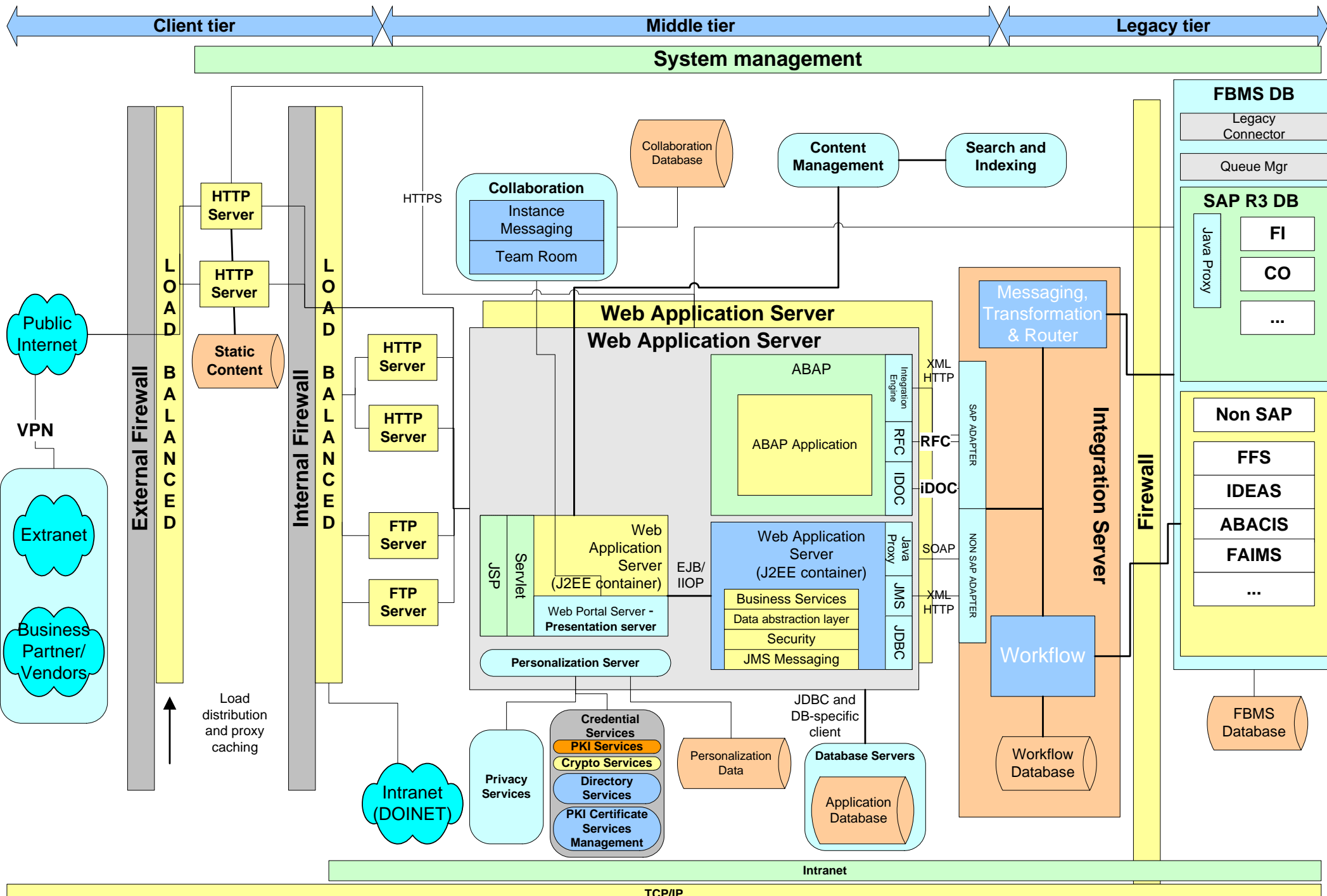


*This view depicts the logical run-time topology, which provides a complete solution architecture that adheres to the composite patterns.*





# Example of Run Time Pattern from USPTO to be reused with different type of Solution and products.





# EA Artifact Centric Views of the ABD workflow

## Feedback & Harvesting Component Life Cycle

