SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

SOA ARCHITECT SUMMIT

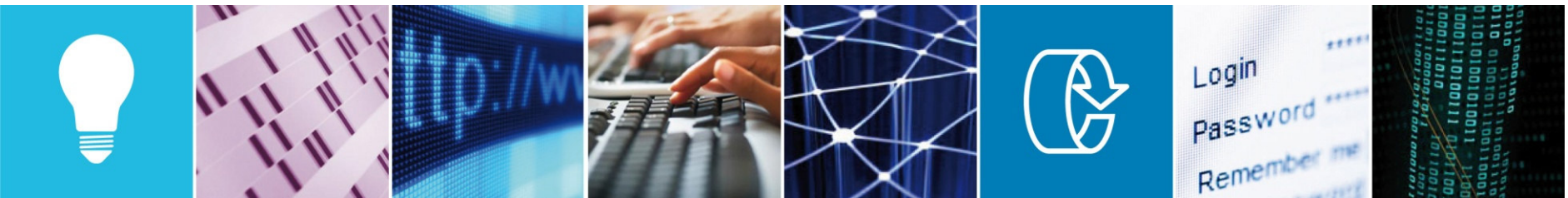
Turn your ideas into practical solutions.

IBM

Service Design and Creation

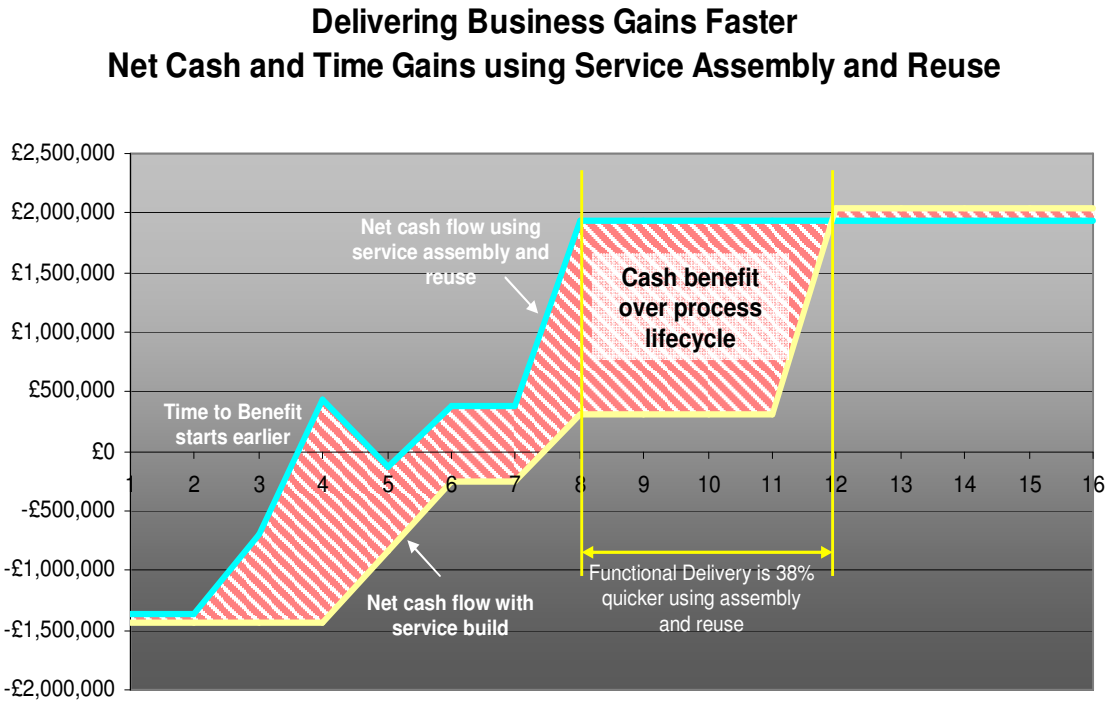
Andrew Joss

Wednesday 9th September



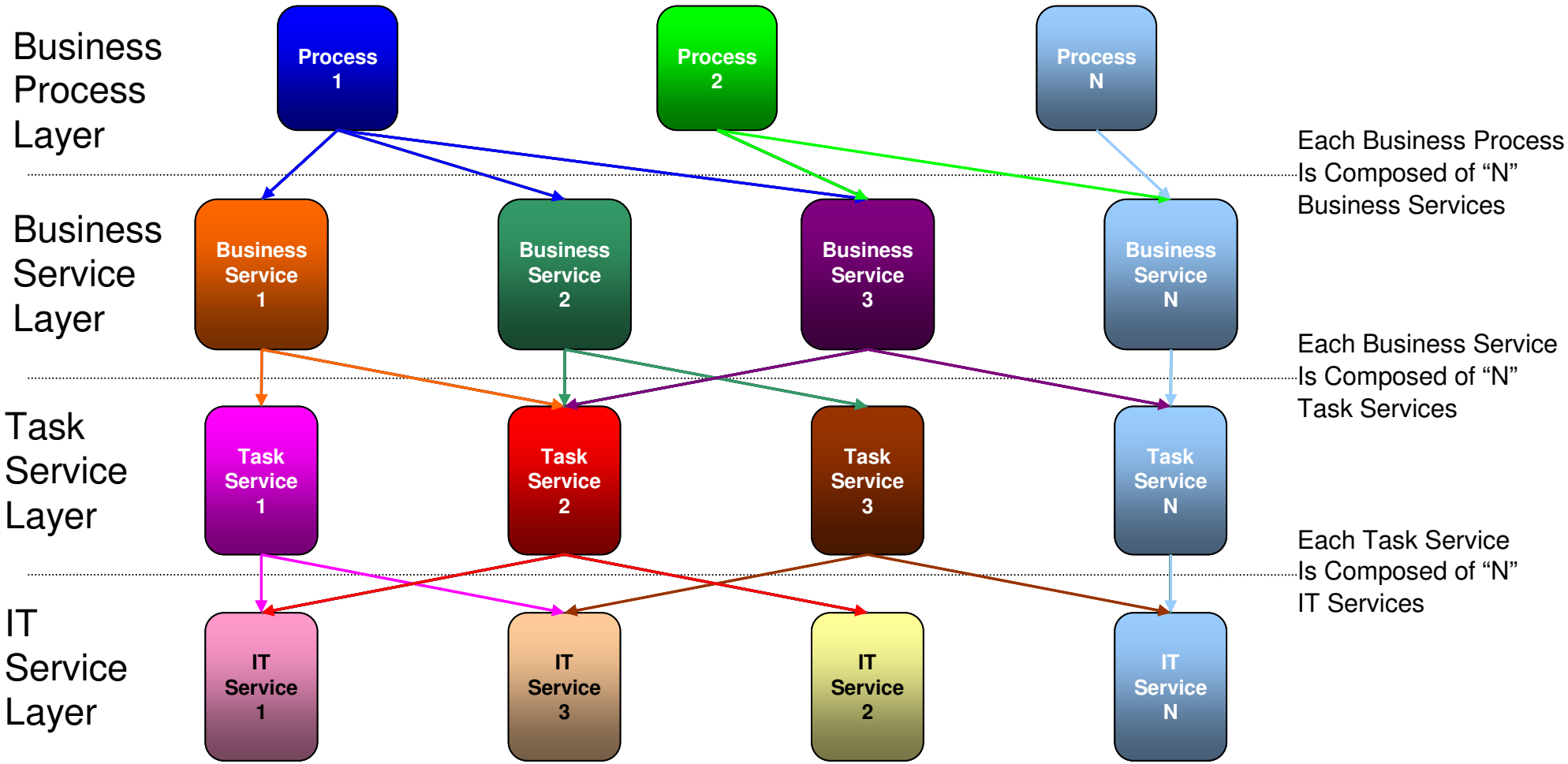
Why is Service Design & Creation Important?

- **If you get this right it will**
  - Reduce project delivery times
  - Reduce the costs of projects
  - Deliver project value quicker
- **But, how do we**
  - Know which services to build?
  - Know which services can be reused?
  - Know if everything is a service?
  - Manage funding & costing models
- **Service Decomposition is a technique to define a language used to break down project/silo barriers**



Service Decomposition Model

Applying a top down and a bottom up approach means we can determine the relationships between the layers



Business Services

- **Business Services are considered to be coarse grained business capabilities that**
  - Are composed of lower level service
  - That can provide functional capability in their own right
  - That can be composed or choreographed into business processes or composite applications
  - Drive value in their own right
  - Typically tend to be:
    - A function step within a business process
    - A value entity in it's own right
    - A function step which is part of a business or composite application
    - Services that function across multiple applications

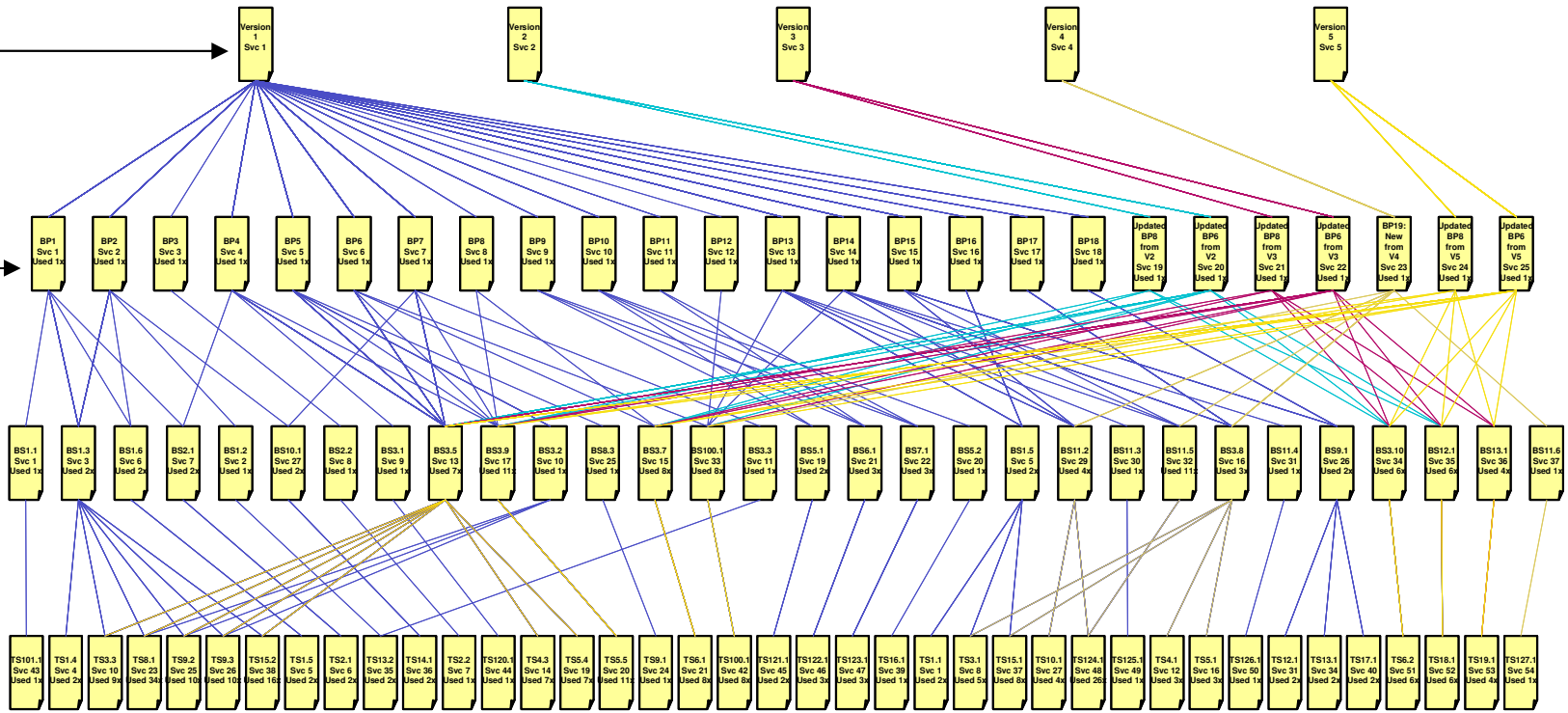
Service Decomposition Model

The diagram below shows the Service Decomposition Model for a customer programme

- **There were 5 phases to the programme (i.e. 5 projects)**
- **The colour of the lines denotes a relationship between a higher and lower level service**
- **Lower level services with multiple coloured lines pointing to them show reuse**
- **Higher level services are composed of the lower services connected by the coloured line**

Each Phase Is

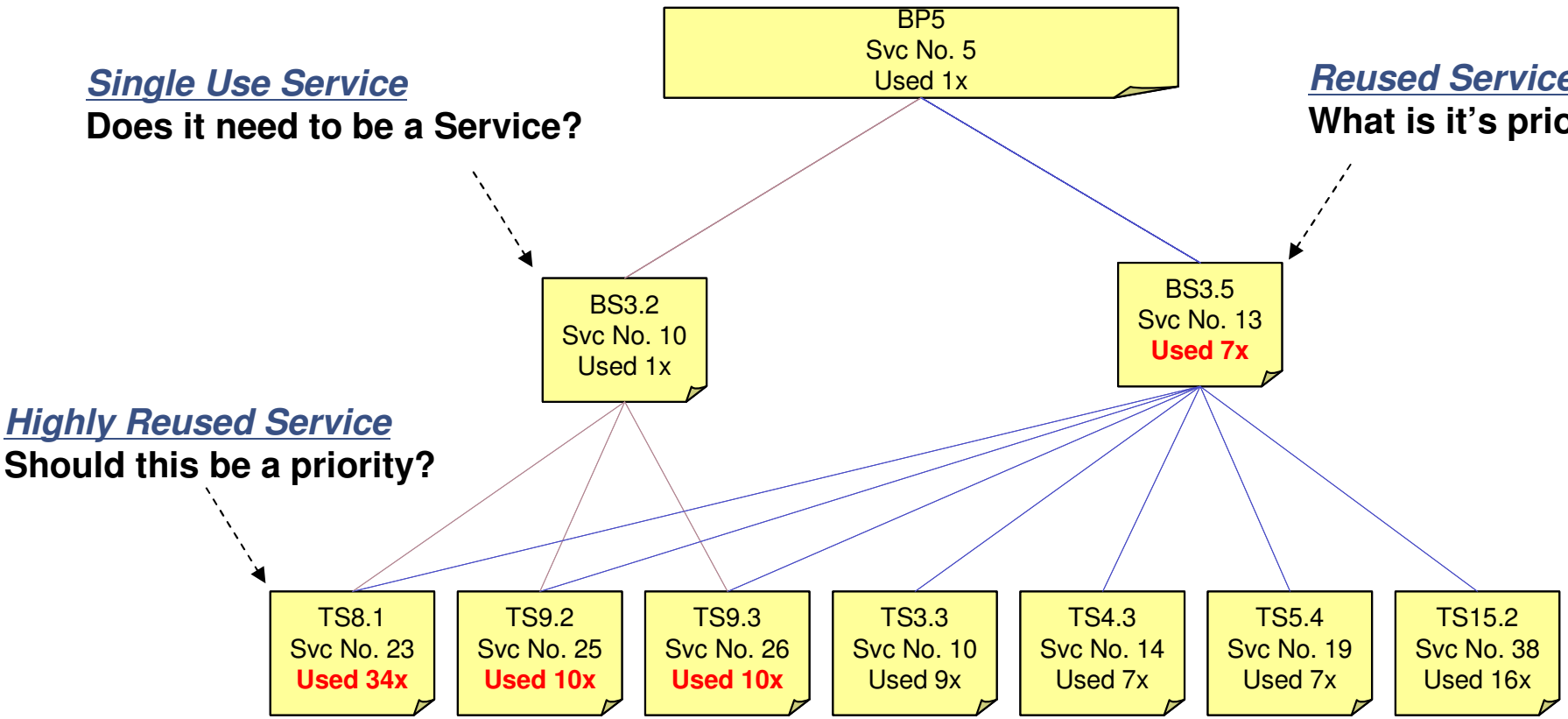
Composed of Business Processes



Service Decomposition Model – Detailed Look

By looking at the Service Decomposition model we can determine

- **Which services are used more than once? (i.e. reused)**
- **How many times is a service reused?**
- **What do I save when I reuse a service as opposed to building a new one (for example)?**

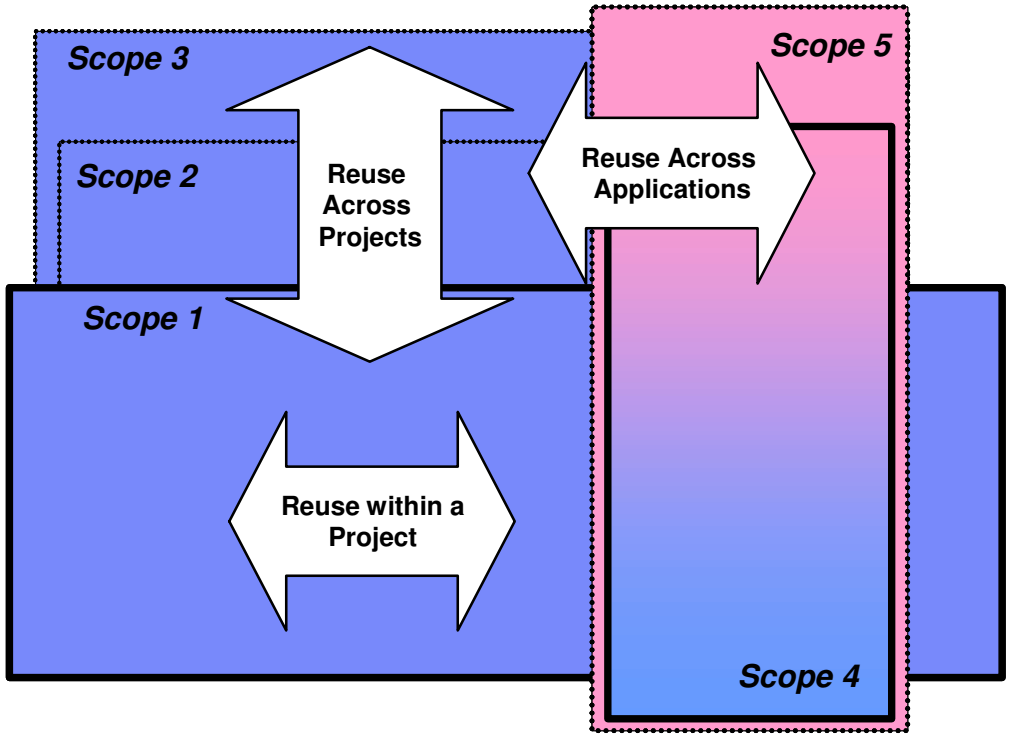


Service Decomposition Analysis

- **Perform high level process analysis of the business operations**
  - Either by documenting the processes of the business or identifying and interpreting process documentation
  - High level views of current and planned business & IT projects
  - Holistic view of the IT architectures that support the business operations
- **Based on process analysis identify conceptual/real business services that relate to business processes and roles within an organisation**
  - Build a Service Decomposition model that links the Business Services to the higher and lower level services
  - Identify IT use & impact as well as IT capability gaps
- **Based on these business services start to identify Business Service reuse potential across processes, projects and lines of the business**
  - Identify reuse potential through the Service Decomposition model and attribute value through reuse
  - Identify IT constraints around reuse and value release
- **Prioritise these Business Services based on business requirements (projects or processes)**
  - Value analysis to determine priority and sequence of service construction

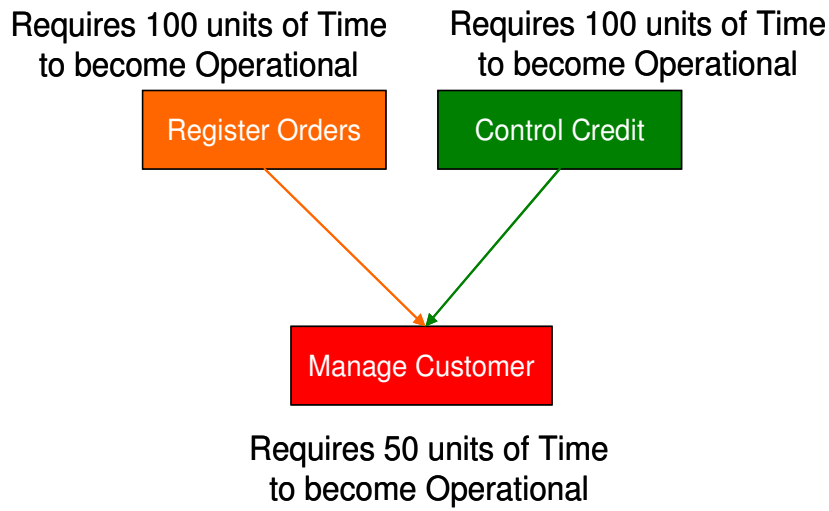
The Value of Service Reuse

- **Services can be identified that would be reused**
  - within a project,
  - across projects,
  - across applications
  - and across differing parts of the business
- **This could be enabled by a Composite Application capability**
- **Building Services and reusing them reduces the time taken to deploy them after the initial build**
  - This should save time after the initial build
- **By saving time, the cost of the projects should also reduce**
- **If the Service is a value adding service for the business, then the time saving becomes a time to value benefit for the organisation.**



How Value Is Built Up

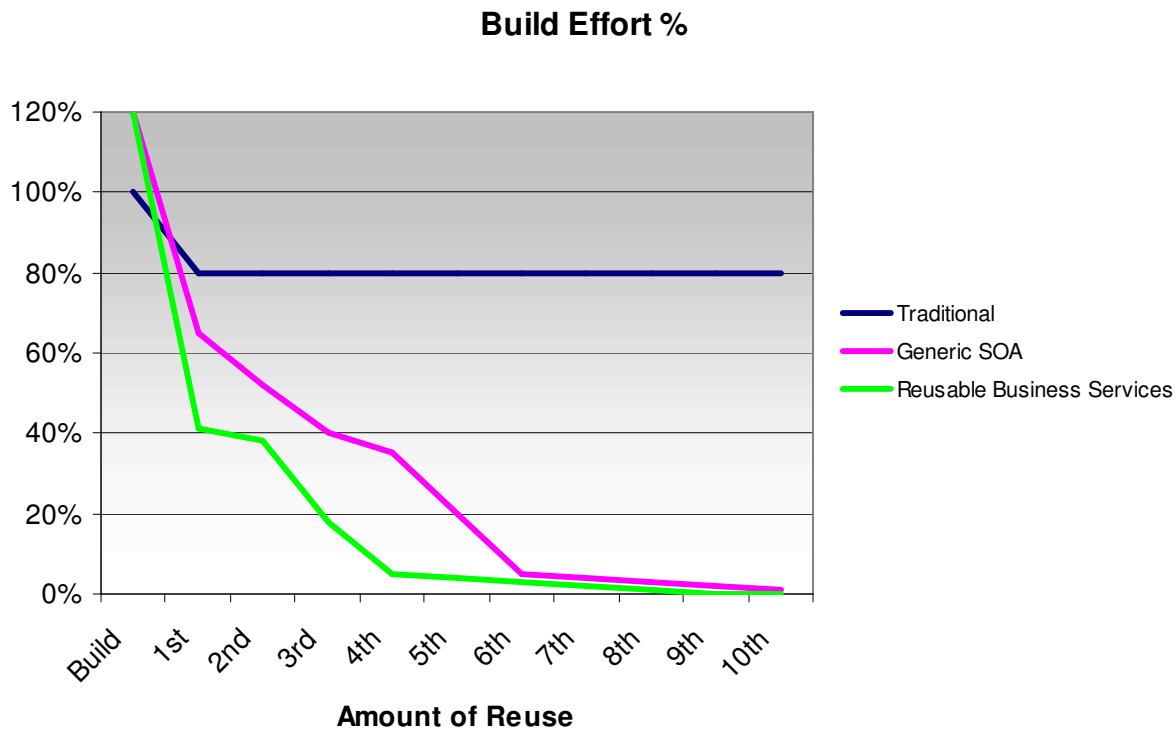
- **By deploying a Reusable Service – the time taken to deploy a higher level Service is significantly reduced**
- **In the example to the right:**
  - **The Register Orders service is composed of Register Orders and the lower level service Manage Customer**
  - **The Control Credit service is composed of Control Credit and the same lower level service Manager Customer**
  - **By Reusing the Manage Customer service the time taken to make operational the Control Credit service is reduced**
  - **As Manage Customer has been** reused **by Control Credit**
    - the time included for this service is reduced by 50%
    - the value of Control Credit is delivered 17% quicker



Name	Units of time	Including Manage Customer	Total
Register Orders	100	50	= 150 units
Control Credit	100	(50 – 50%) 25	= 125 units
Time Saving			= 25 units = 17%

Reuse Profiles

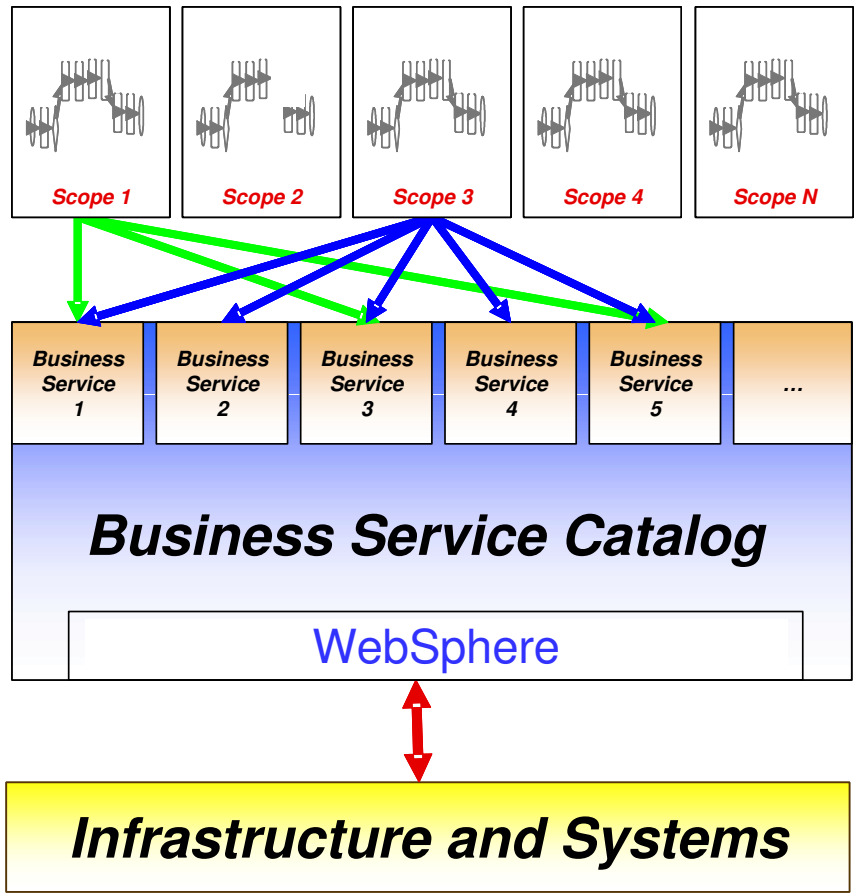
- **Reuse Profiles define how much time is saved by reusing a service each time it is reused**
- **They operate between the Service Decomposition layers**
- **They represent a percentage saving in time during the construction process.**
- **They can vary in scale between each layer i.e. the reuse profile between business process and business services might be different than the profile between business services and task services**



The chart above shows an aggregate Reuse profile between 2 layers of the Service Decomposition Model

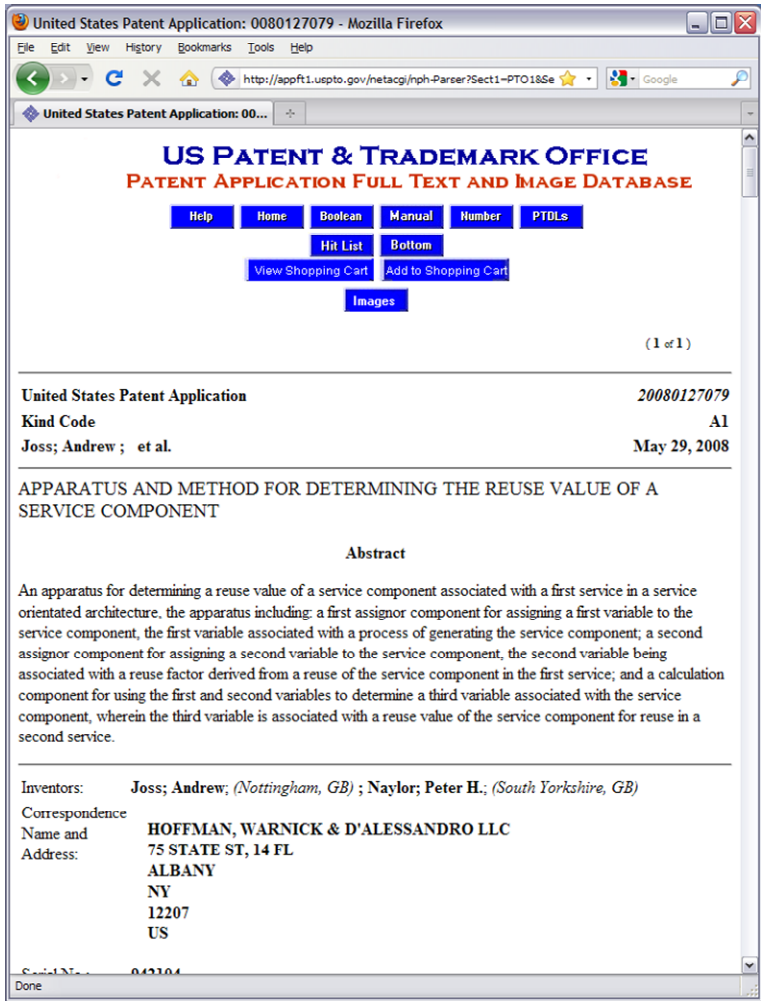
Creating a Business Service Catalog

- **The proposed approach is used to construct a set of reusable Business Services that would be available through a Business Service Catalog**
- **This would provide a set of discrete, reusable business functions available to current and future projects**
- **These services would reduce the current and future project time lines as they were designed for reuse at the outset and had been identified as services that would be reused**
- **This approach would provide an insulating layer between the needs of the project(s) and the functional capabilities of the infrastructure and systems.**



Patented IBM Service Decomposition Method

- **The amount of time saved by deploying reusable services is calculated using the following patented value method**
  - **APPARATUS AND METHOD FOR DETERMINING THE REUSE VALUE OF A SERVICE COMPONENT**
  - **US Patent Office**
    - **http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220080127079%22.PGNR.&OS=DN/20080127079&RS=DN/20080127079**
  - **European Patent Office**
    - **http://v3.espacenet.com/publicationDetails/biblio?adjacent=true&KC=A1&date=20080529&NR=2008127079A1&DB=EPODOC&locale=en_EP&CC=US&FT=D**
- **This method calculates the time saved to deduce the cost reduction/avoidance associated with that saving**



Time Saving

- **A Service Decomposition Model has all of the Business and IT Services defined**
- **By using a patented IBM capability to model Reuse across each hierarchy level, it is possible to determine how many times any type of service is Reused across the defined Business Processes (figure 1)**
- **Each time a Service is Reused, the effort to deploy it is increasingly reduced (figure 2)**
  - ***The profile is completely dependant upon the services in scope, the potential for reuse and how the services are built***

Fig 1

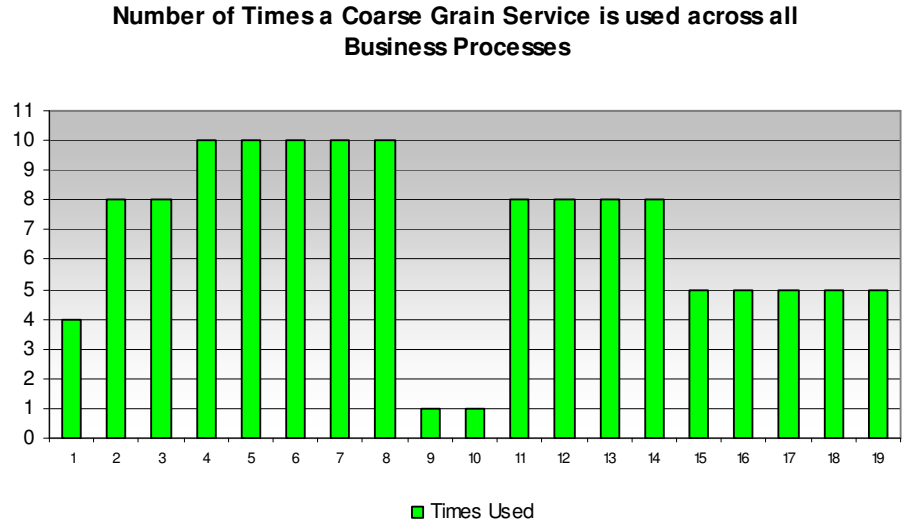
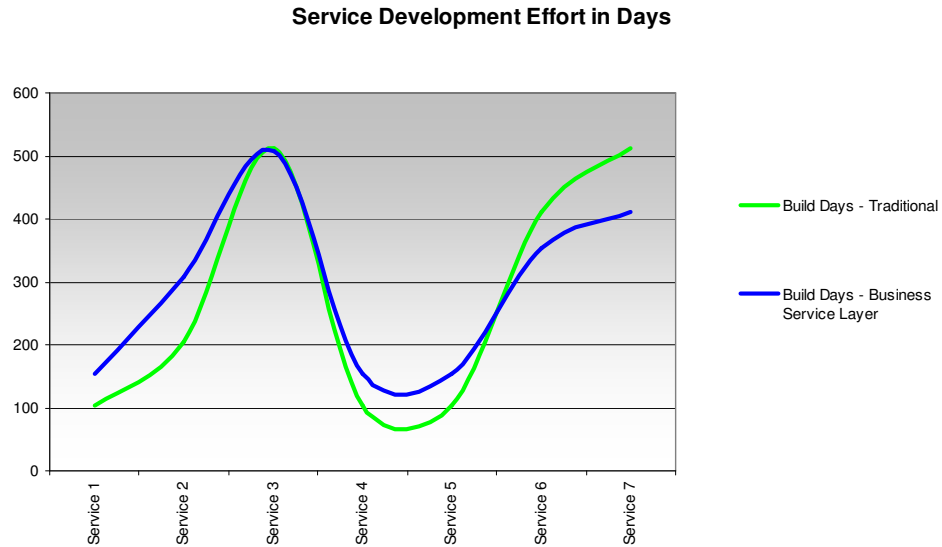
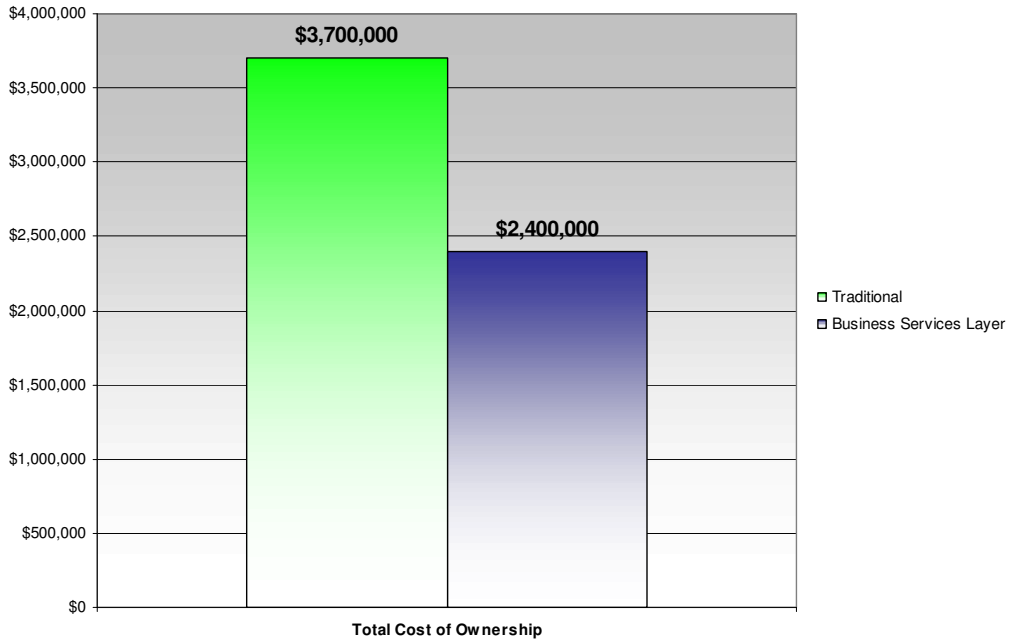


Fig 2



Time Saving Becomes a Cost Saving

- **The ultimate outcome of all this activity is to demonstrate the benefits associated with identifying and deploying reusable services**
  - If there aren't any benefits – why are you doing it?
  - If there are benefits – how big are they and are they sufficient to make it worth doing?
- **With an understanding of the build effort it will be possible to build up a *total cost* picture. This should be supplemented with the *value* picture as well.**
- **The types of questions this should address include**
  - How much would this cost compared to how we develop capabilities today?
  - Does this drive any time to value benefits for the business and if so - how big are they?



Identifying the Optimal Sequence

- **Using the Service Composition Model, it is now possible to understand the optimal sequence in which to deploy processes to accelerate Time to Value even further**
  - The example in Figure 1 shows an increase in ROI between the Least Optimal and Optimal Process sequences

Optimum Process Sequence

Optimal Sequence			
Least Optimal Sequence		With Reusable Services	
Process Permutation Sequence	5 Year Benefits	5 Year Return on Investment	5 Year NPV
Process 1 ⇨ Process 4 ⇨ Process 2 ⇨ Process 3	£27,963,251	123%	£11,323,197
Process 3 ⇨ Process 2 ⇨ Process 1 ⇨ Process 4	£32,055,901	179%	£16,211,625
Process 3 ⇨ Process 2 ⇨ Process 4 ⇨ Process 1	£31,710,663	206%	£16,899,114
Process 3 ⇨ Process 4 ⇨ Process 1 ⇨ Process 2	£30,228,520	163%	£14,485,139

- **Comparison can also be made with deploying the processes using a traditional approach**
  - The example in Figure 2 shows an increase in ROI between optimal process sequences when using reusable services

Permutations Summary and Analysis

Benefit Area	*Optimal Sequence - With Reusable Services*	*Optimal Sequence - Without Reusable Services*	*Least Optimal Sequence - With Reusable Services*	*Least Optimal Sequence - Without Reusable Services*
5 Year Benefits	*£32,055,901*	*£26,206,781*	*£27,963,251*	*£19,189,959*
5 Year Costs	*£10,340,733*	*£18,825,361*	*£12,620,617*	*£18,825,361*
5 Year Net Benefits	*£21,338,643*	*£7,381,420*	*£15,413,062*	*£364,598*
5 Year Return on Investment	*206%*	*39%*	*123%*	*2%*

In Summary

- **Service Design and Creation is important because if you get this right it will**
  - Reduce project delivery times
  - Reduce the costs of projects
  - Deliver project value quicker
- **Service Design and Creation will help us**
  - Know which services to build
  - Know which services can and should be reused
  - Know if everything should be a service
  - Shape funding & costing models
- **Service Decomposition can be used a technique to define a language used to break down project/silo barriers**

SOA ARCHITECT SUMMIT

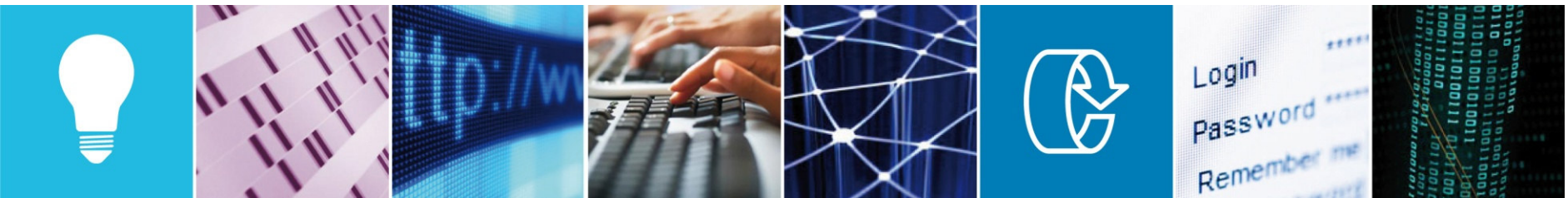
Turn your ideas into practical solutions.

SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

Paul Smith / BPM Specialist

9th September 2009

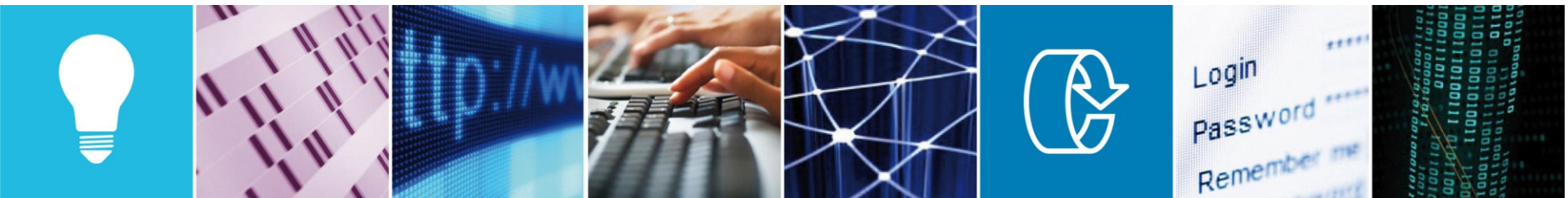


SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

IBM

IBM BPM BlueWorks



BPM BlueWorks is:

A set of cloud-based BPM tools and content enabling Business Leaders, Business Analysts, and Business Professionals to experience the value of BPM by creating BPM Business Designs in the cloud, leveraging pre-built content, and collaborating through community tools.
With BlueWorks, users can:

Learn about BPM

Learn strategies, trends, and best practices for making smart process decisions

Collaborate with the Community

Leverage community insight and access shared content

Experience BPM

Capture business intent, understand capabilities, sketch processes

Optimize Processes

Extend strategy to drive processes improvement, and deploy with IBM BPM Suite

The BPM BlueWorks Experience
Acquire Expertise, Map Strategy, Execute Processes

1. Access business & industry-specific content to understand the value of BPM

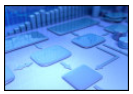

Demos / Videos

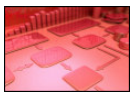

Best Practices

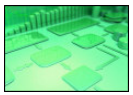

Web casts / Pod casts


Papers / Case Studies

2. Collaborate with the community and leverage pre-built strategies, processes, and measures

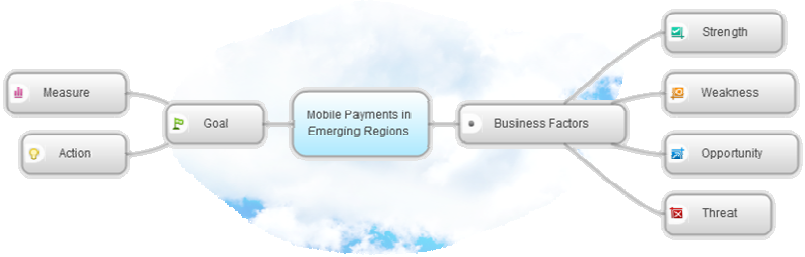

Process Maps


Capability Maps

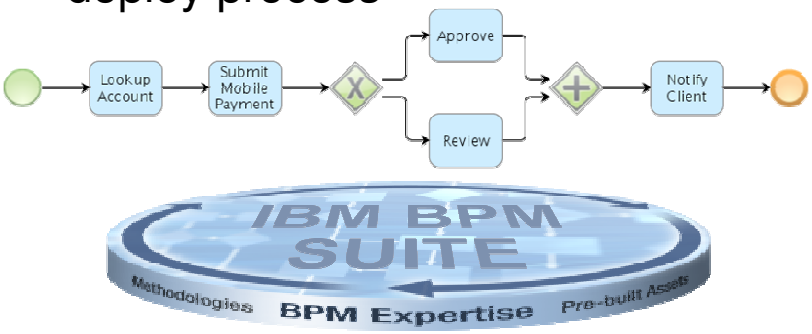

Strategy Maps


Business Measures

3. Capture business intent, capabilities, & process in the cloud



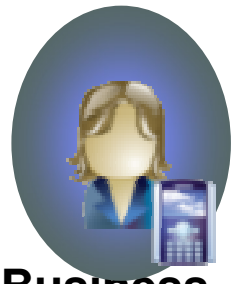
4. Easy on-ramp to BPM suite to test & deploy process



Seamlessly Link Business Strategy to Process Execution

BPM BlueWorks Target Users

BPM BlueWorks is designed and targeted towards business users. In the same way that DeveloperWorks provides content and expertise to developers, architects, and other technical professionals, BPM BlueWorks provides business and industry tailored content around business process management.



Business Leaders

Business Professionals

- find tailored, industry-specific content to learn the business value of BPM.
- leverage new strategy tools to capture business intent and find process improvement opportunities

Business Analysts

- Leverage pre-built, industry-specific process maps & business measures to accelerate BPM projects
- Directly import BPMN process maps from BlueWorks to WebSphere Business Modeler

New BPM Business Design Tools
Translate Strategy into Business Process Improvement

Strategic Intent & Motivation

Capture business strategy understanding that guides the operations and actions of the business

Operational Capabilities

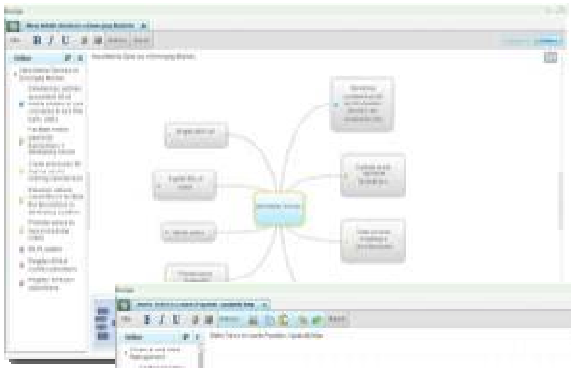
Refine strategic intent into operational capabilities to identify transformation opportunities

High-Level Processes

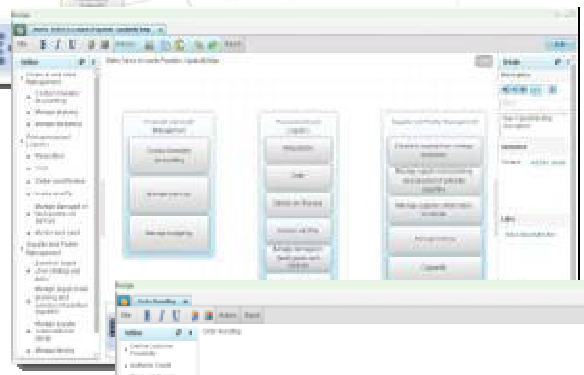
Link operational capabilities to business processes to jump start automation

Drive Process Improvements

Capture business intent to understand, transform, automate and measure business processes



Strategy Maps

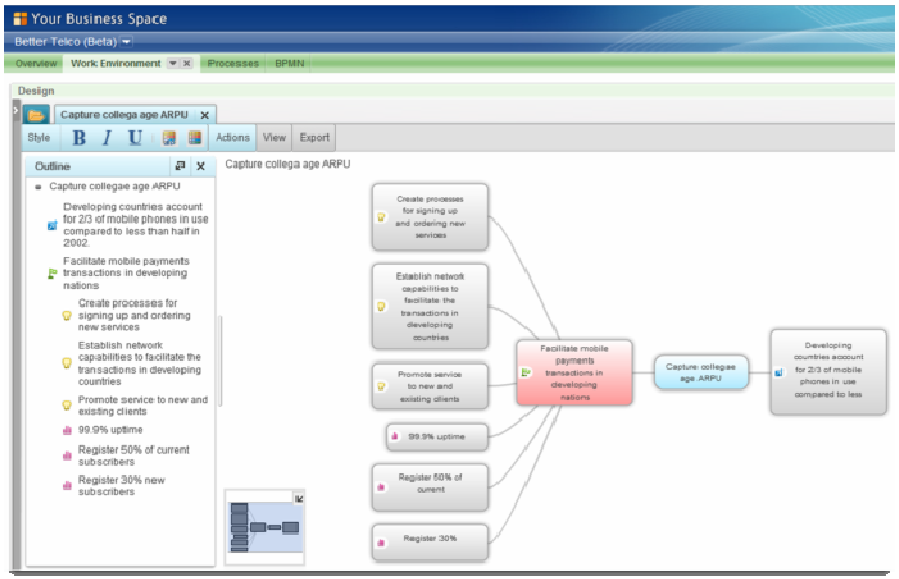


Capability Maps

Process Maps

Capture Business Intent
With Strategy Maps

- *Strengths*
- *Weaknesses*
- *Opportunities*
- *Threats*



- *Goals*
- *Actions*
- *Measures*

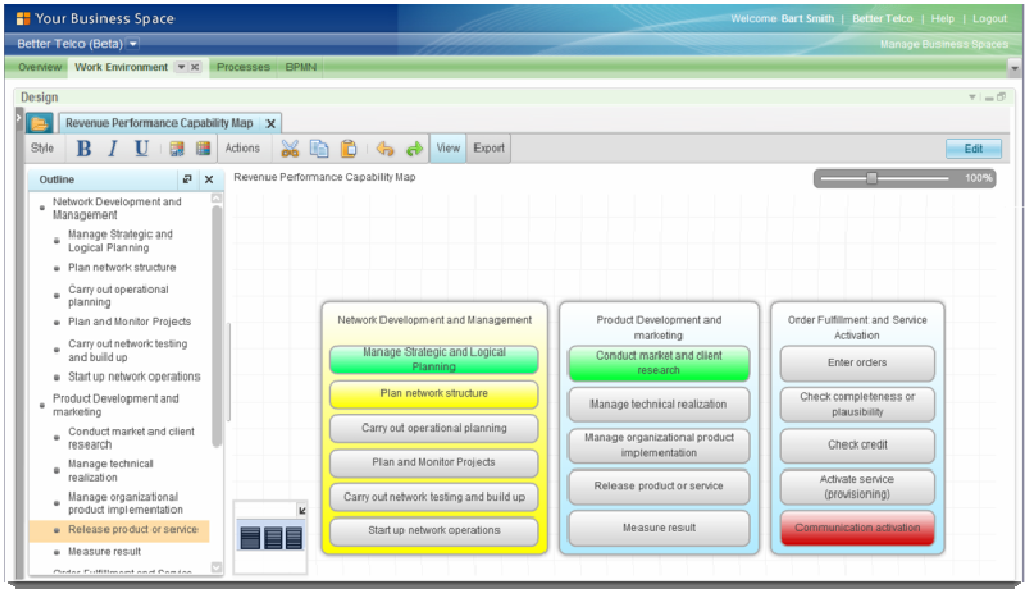
Build Comprehensive Strategies

- Capture key business factors
- Develop a measurable plan
- Associate strategy with core competencies
- Easily share strategies

Leverage Powerful Mapping Tools

- Copy and paste into outline view
- Export to PowerPoint
- Link to capability maps
- Add attachments
- Color code for visual enhancement

Document Business Activities & Competencies
With Capability Maps



Capture Core Competencies

- Define business activities
- Highlight competencies and capabilities
- Associate capabilities with processes
- Easily share definitions

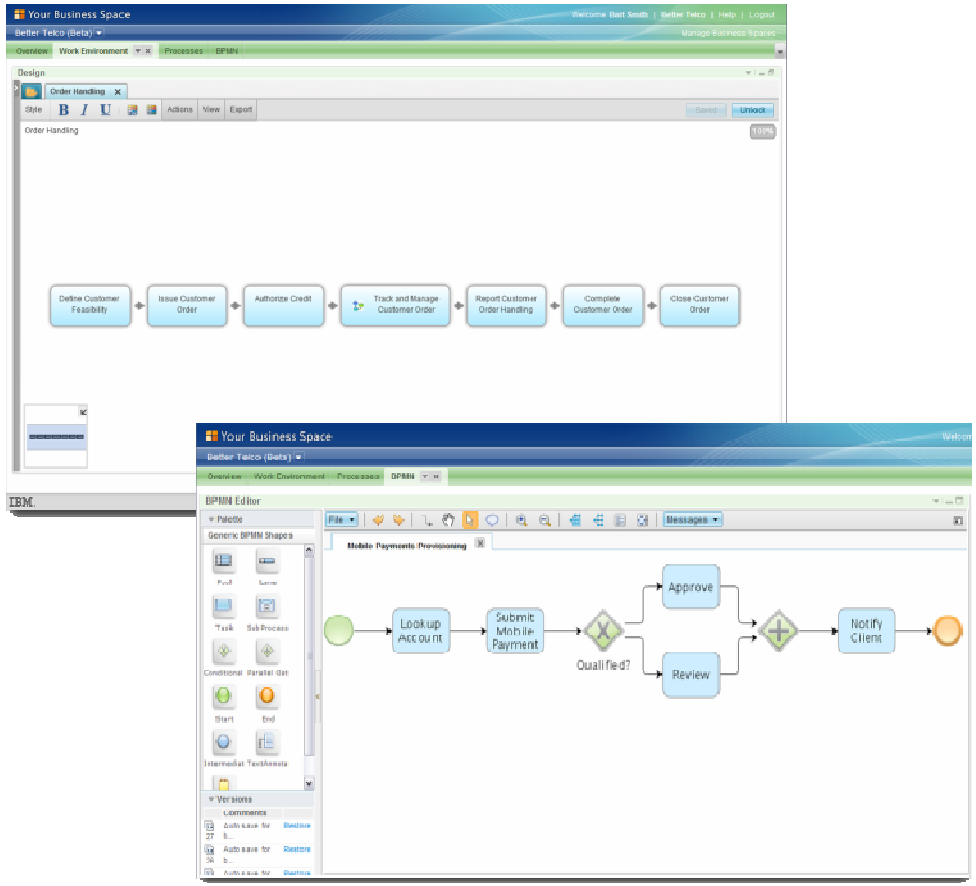
Leverage Powerful Mapping Tools

- Export to PowerPoint
- Link to process maps
- Add attachments
- Color code for visual enhancement

Define Core Business Execution
With Process Maps

Map High-Level Business Processes

- Capture business activities, owners, and associated information
- Define BPMN sub-processes
- Create critical decision branches
- Easily share process maps

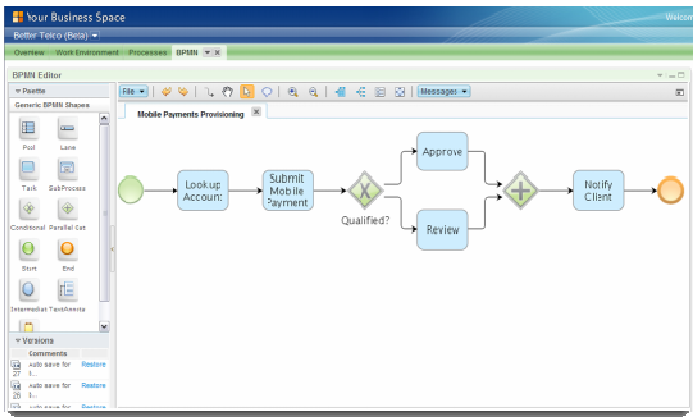


Leverage Powerful Mapping Tools

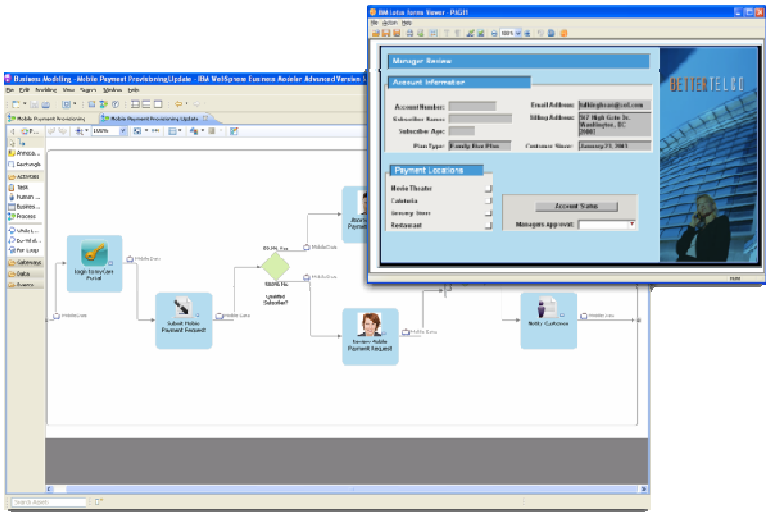
- Export BPMN sub-processes
  - Rational Asset Manager
  - File System
- Import BPMN sub-processes to WebSphere Business Modeler
- Print and convert processes to PDF

Moving from High-Level Process Mapping to Robust Modeling and Simulation

BPM BlueWorks



WebSphere Business Modeler



Both BPM BlueWorks and WebSphere Business Modeler provide the capability to model and extend high-level BPMN processes. WebSphere Business Modeler provides advanced modeling capabilities and other features including:

- Process simulation and analysis
- Forms design and storyboarding
- Interactive process design to directly deploy process models to a managed test environment

Clients can export process models from BlueWorks into Business Modeler to extend and test them using these additional capabilities.

Share Strategies & Processes Securely
In a Zero-Cost Cloud Environment

Secure Cloud Provisioning

- Users' content and process assets are protected
- BPM assets are only available to authorized users of a space
- Users still reap the cost benefits of working in the cloud

Secure Registration

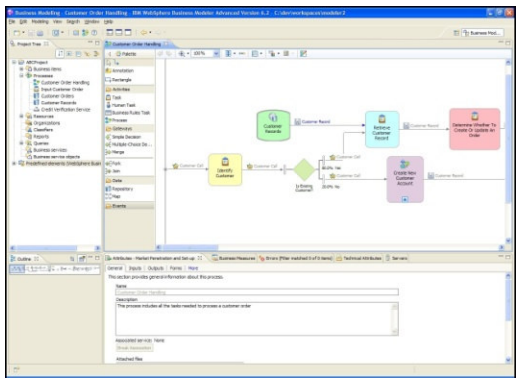
- The first registered user for a company / group becomes the BlueWorks administrator for the company
- Subsequent registrations for the company / group are approved by the administrator
- Registered users can belong to more than 1 space

On Ramp to IBM BPM Suite

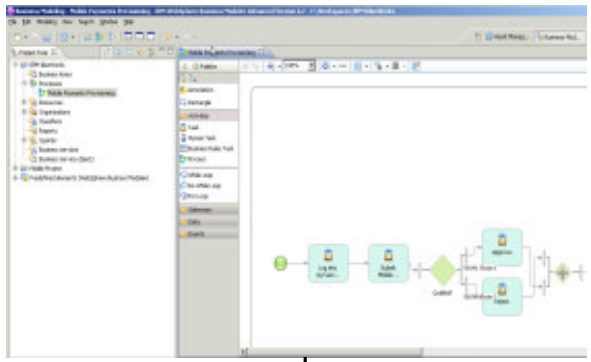
BPM BlueWorks provides an easy on-ramp to the IBM BPM Suite allowing a seamless move from business strategy to process execution. ***Users cannot deploy and run processes in the cloud***

1. Export Modeler artifact to RAM or local file system

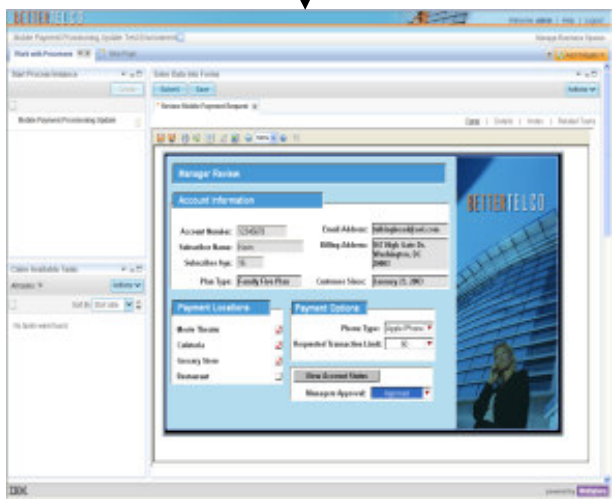
2a. Start WebSphere Business Modeler



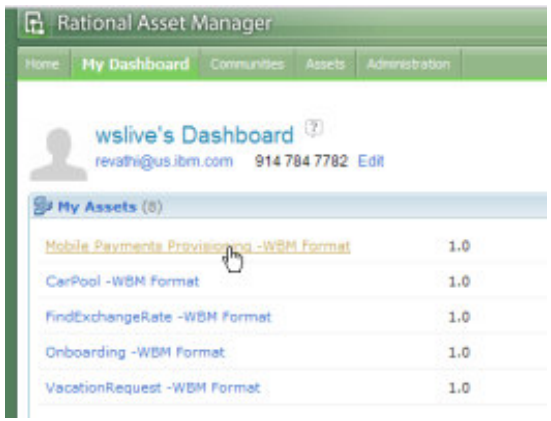
3. Import artifacts into WebSphere Business Modeler



4. Test the process using interactive process design capabilities



RAM Dashboard



BPM BlueWorks and the IBM BPM Suite

Improve BPM success with process discovery driven from business intent, and a robust platform to continuously optimize business processes

Capture business intent
Collaborate around business design

Prove value quickly
Access tools and process assets

Deploy, Execute, and Change

Model and Simulate

Monitor, Analyze, Predict and Act

Governance and Compliance

Get Started with BPM BlueWorks
Sign-up on NOW!!

Register Your Company

- The first registered user for a company / group becomes the BlueWorks administrator for the company
- Subsequent registrations for the company / group are approved by the administrator

Learn More

To learn more about BPM BlueWorks, please visit:

www.ibm.com/software/solutions/smartwork/blueworks

or contact your IBM representative

SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

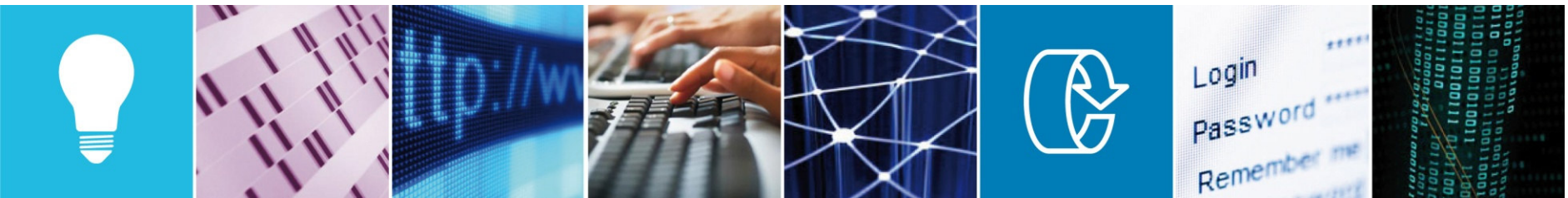
SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

IBM

How to tackle Security in a SOA world?

Martin Borrett
Lead Security Architect NE Europe, WW Tivoli Tiger Team



Abstract

- **In this session I will highlight the progress that our clients have made in understand the security challenges of SOA. Secondly I will discuss the technology and standards available today to solve these issues. Finally I will share successful customer examples of bringing these two aspects together to deliver business value and mitigate these risks.**

Agenda

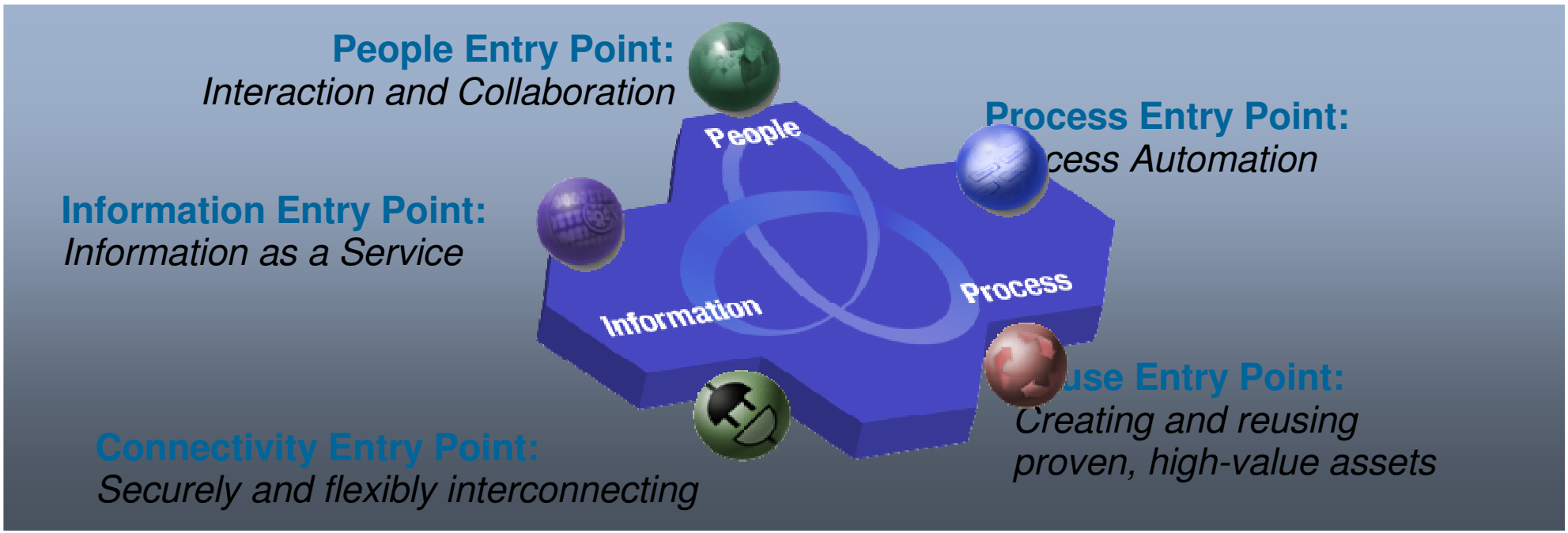
- **SOA Security Considerations**
- **SOA Entry Points and Security**
- **SOA Security Architecture Approaches**
- **Security Standards**
- **Technology and solutions**
  - TFIM
  - TSPM
- **Customer Case studies**

Security Considerations for SOA

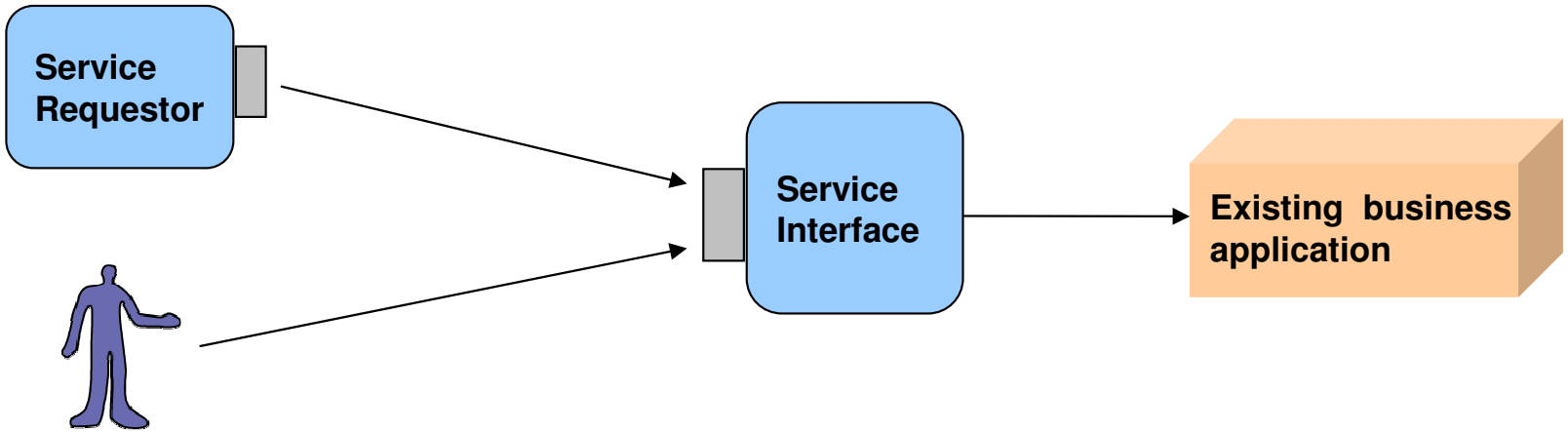
- **Organizational/enterprise boundaries**
  - Perimeter is obscure
  - Identities are managed across boundaries
  - Trust relationships are established across boundaries
- **Composite applications**
  - Ensuring proper security controls are enacted for each service and when used in combination
- **Entities/Identities – users, services**
  - Services have identities
  - Identities and/or credentials are propagated across services
  - Users and services are now subject to the same security controls
- **Greater focus on data/information**
  - Protecting data at transit and at rest
  - Apply consistent protection measures
  - Access to data by applications and services
- **Governance, Risk, and Compliance**
  - Auditing ie. entity identification to specific transactions

The SOA Entry Points

- *When selecting SOA projects, focus on solving **specific business problems** as part of an evolving enterprise architecture*
- *IBM has a variety of assets and best practices around the SOA entry points, based on our **extensive experience with customers***



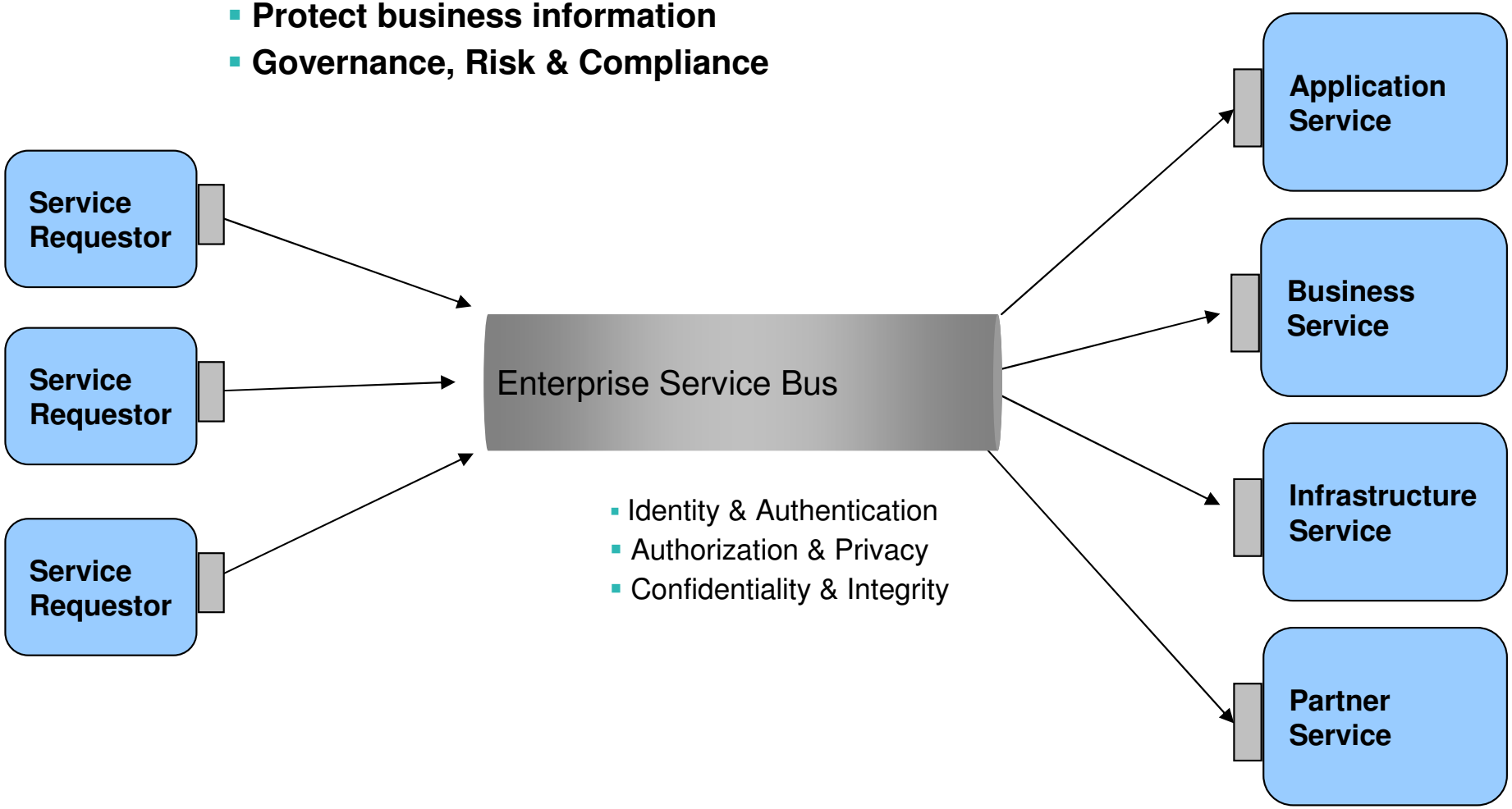
Reuse - Service Creation



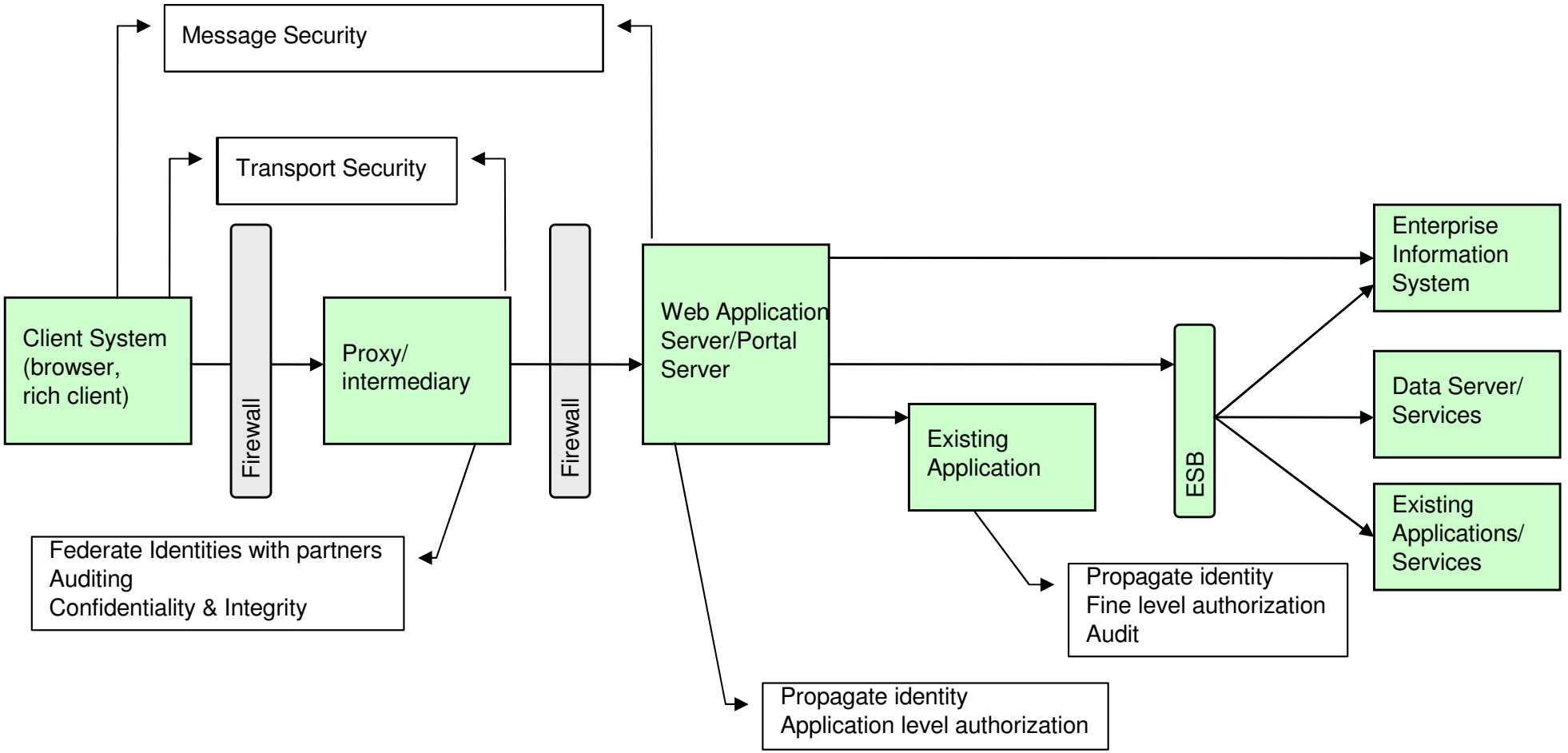
- **Establish identity: Authentication**
- **Protect messages: Confidentiality & Integrity**
- **Accountability: Audit access to service**
- **User experience: SSO, Privacy**

Connectivity - Service Integration

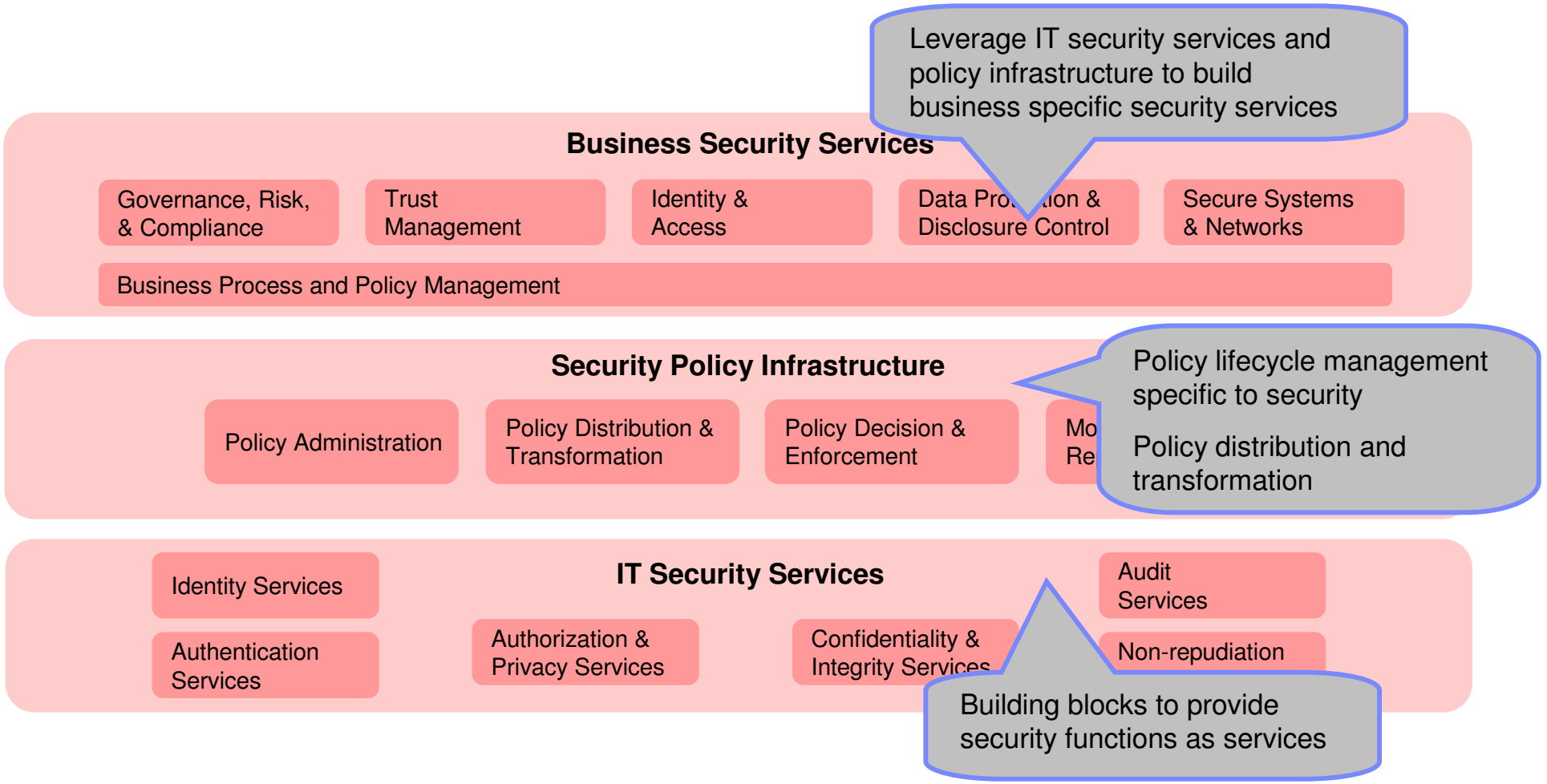
- **Propagate identity: Cross domain/realm identity mapping and token transformation**
- **Reflect business relationships: Trust Management (for data, identity, etc)**
- **Protect business information**
- **Governance, Risk & Compliance**



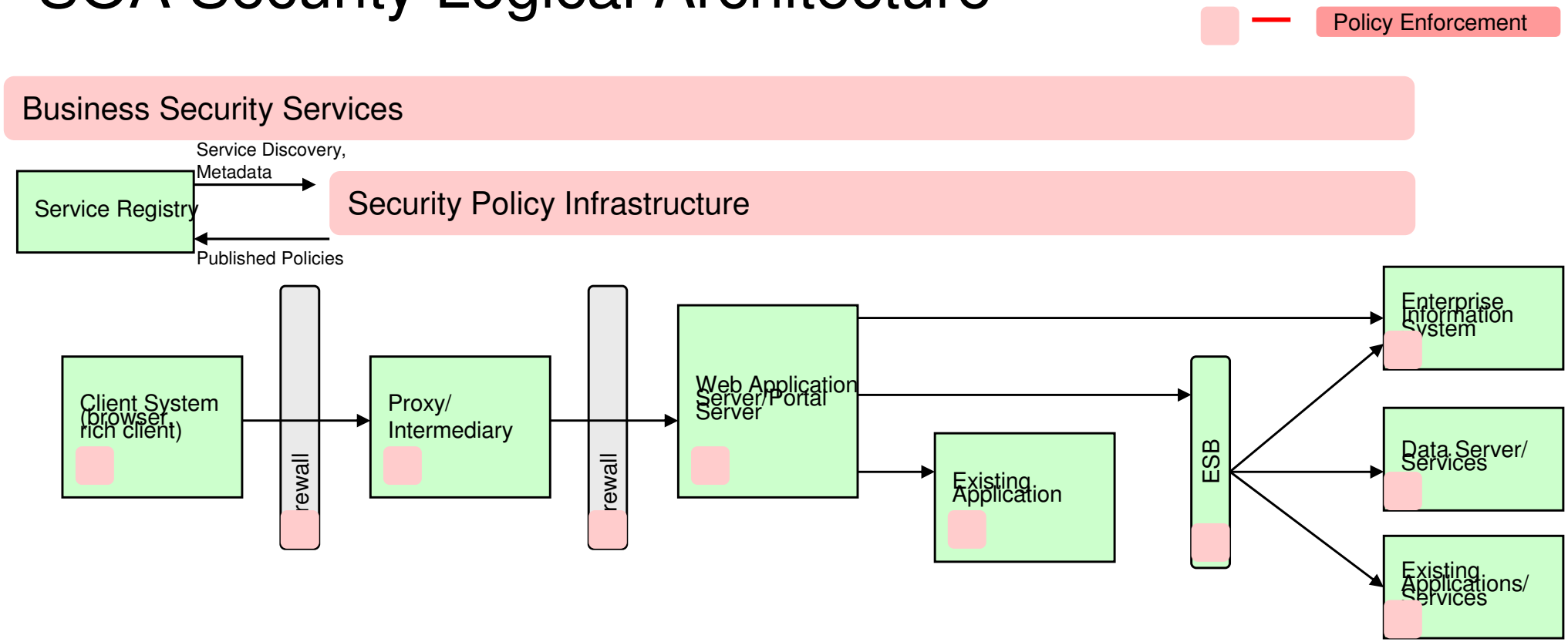
Security in a Typical Deployment Architecture



SOA Security – Reference Model



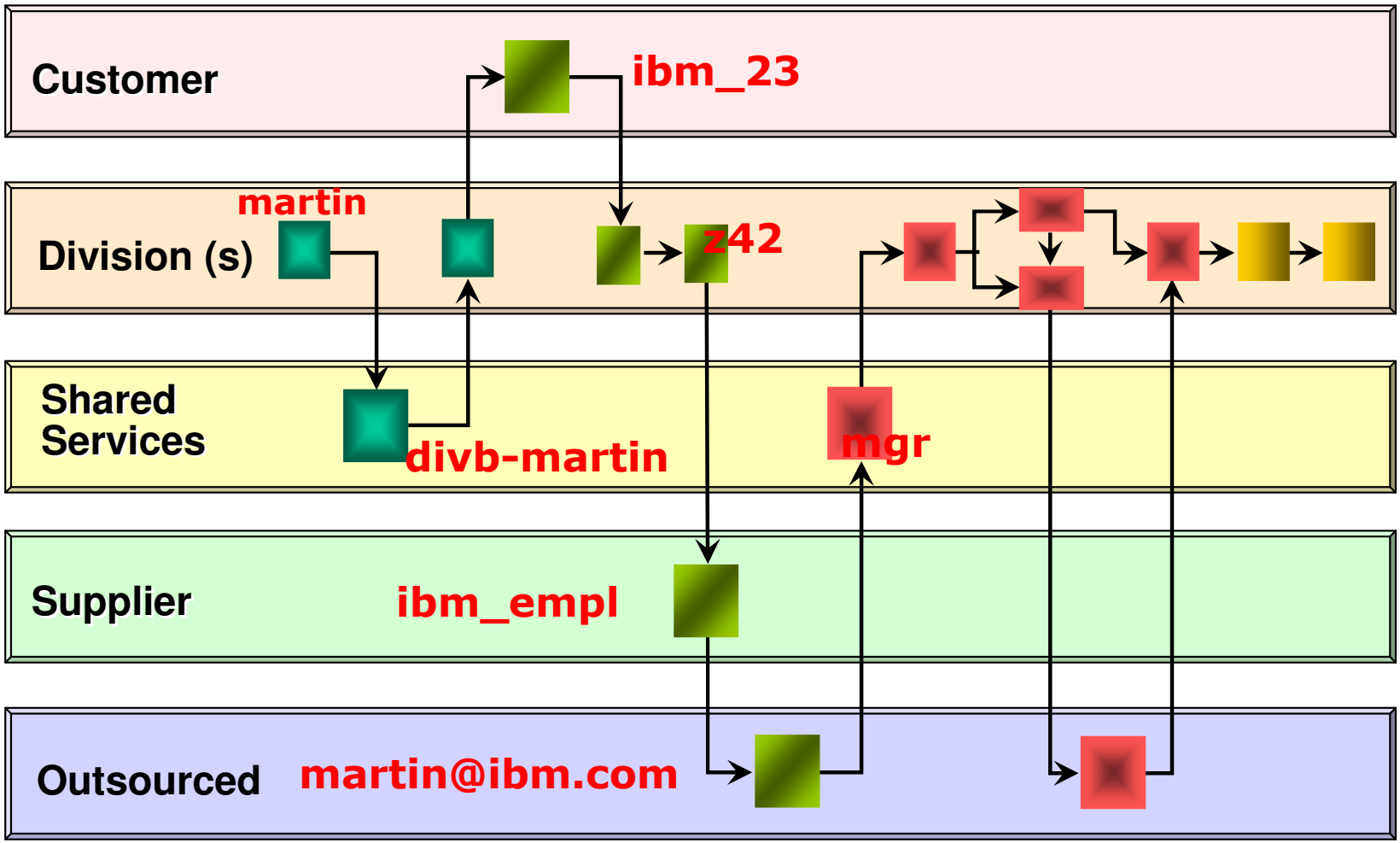
SOA Security Logical Architecture



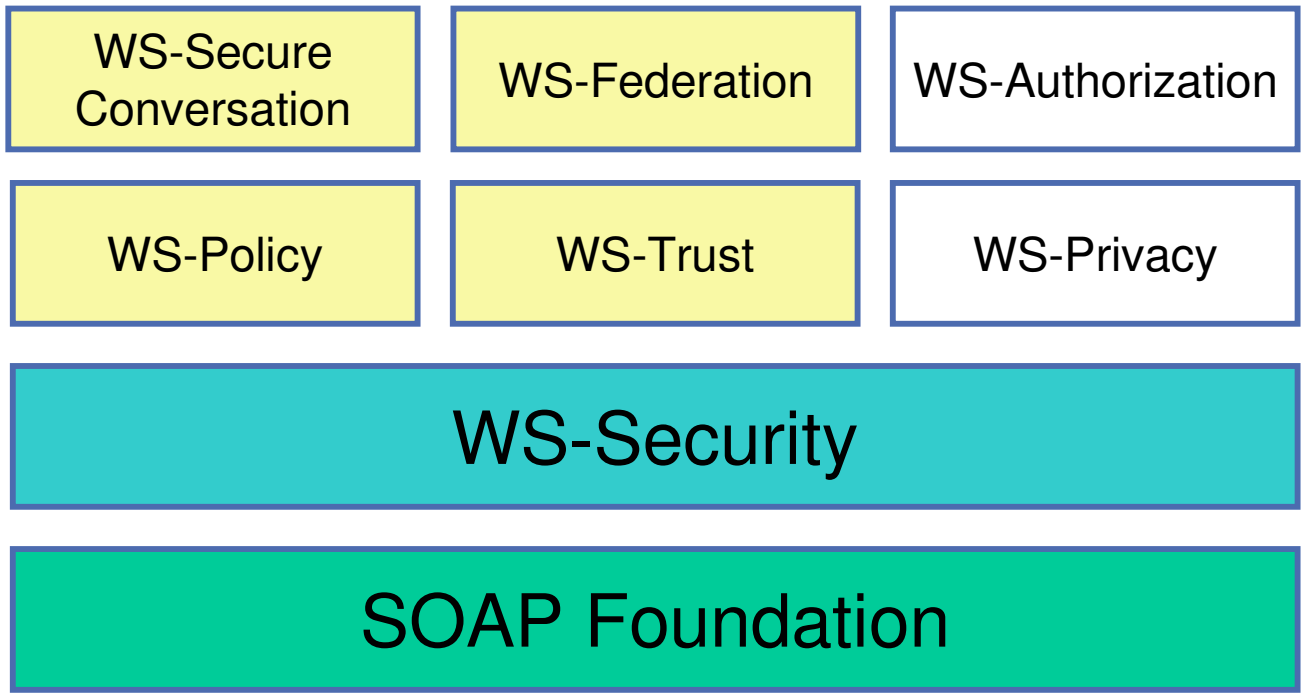
Policies are distributed to not only to Security Services but only to different enforcement points.
The Enforcement points can leverage local capabilities or access centralized security services to enforce policies.

Identity Flow in a Service Oriented Architecture

How does Identity flow between services ?



Web Services Security Roadmap



Web services zone page:
http://www-106.ibm.com/developerworks/webservices/

WS-Security : SOAP Message Security

- **WS-Security : SOAP Message Security**
  - defines "…a standard set of SOAP extensions that can be used when building secure Web services to implement integrity and confidentiality."
- **Allows:**
  - sending Security Tokens to authenticate requests
  - signing Data to ensure data integrity and verify sender
  - encrypting Data to ensure privacy of data
- **Goal:**
  - "End-to-end message content security…"

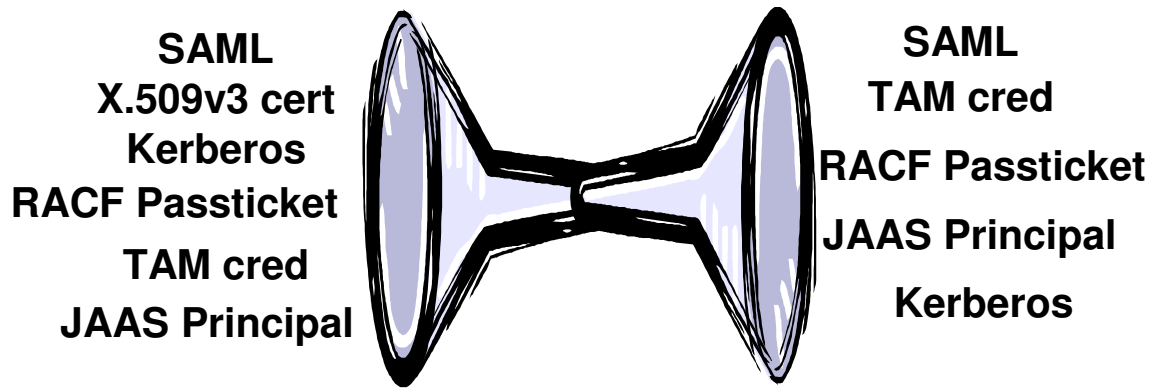
WS-Trust : Overview

- **WS-Trust defines mechanism for:**
  - "…security token exchange to enable the issuance and dissemination of credentials within different trust domains"
- **Defines the *Security Token Service (STS):***
  - Request security tokens
  - Validate security tokens
  - Exchange security tokens
- **IBM Tivoli Federated Identity Manager (TFIM) implements a STS which provides:**
  - Token mediation (validation, mapping, issuance)
  - Identity mediation
  - Authorization (via TAM)
  - Auditing (to CARS)

SOA Credential Translation – using TFIM

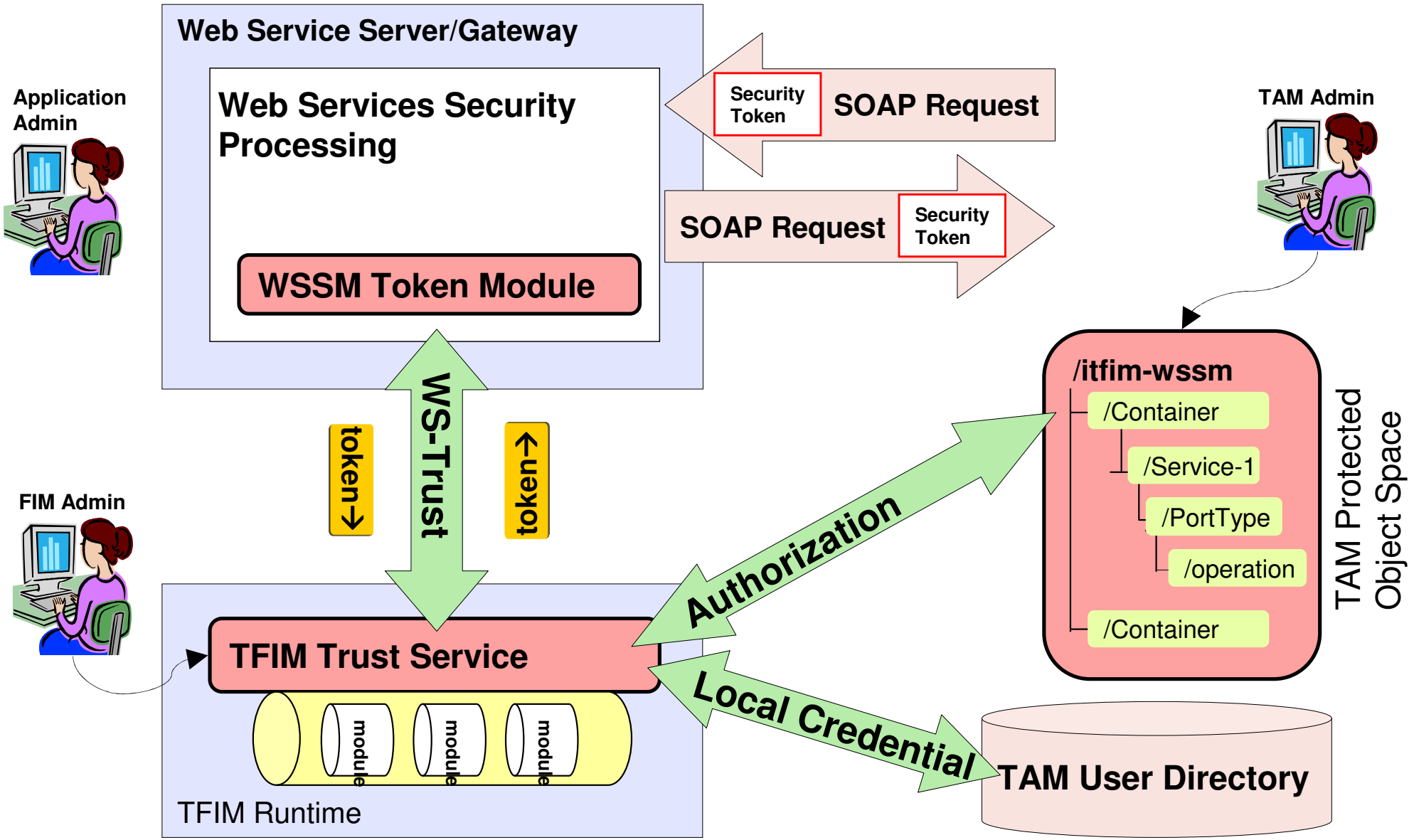
- **Integrate identities for web services / SOA environment**
- **Implement centralized identity mediation & token mapping across multiple, diverse enforcement points**
  - DataPower Gateway
  - WebSphere Application Server (WAS)
  - WebSphere Portal
  - Enterprise Service Bus (ESB)
  - WebSphere Message Broker (WMB)
  - .NET environment
  - Java2 Connector (i.e JDBC)
  - CICS protected by RACF
  - SAP Integration
  - InfoCard for consumer identities
  - ITCAM for SOA for identity-based monitoring

Secure Token Service
(Tivoli Federated Identity Manager)



End-to-End Identity Propagation in a SOA environment

TFIM – Generic Design Overview



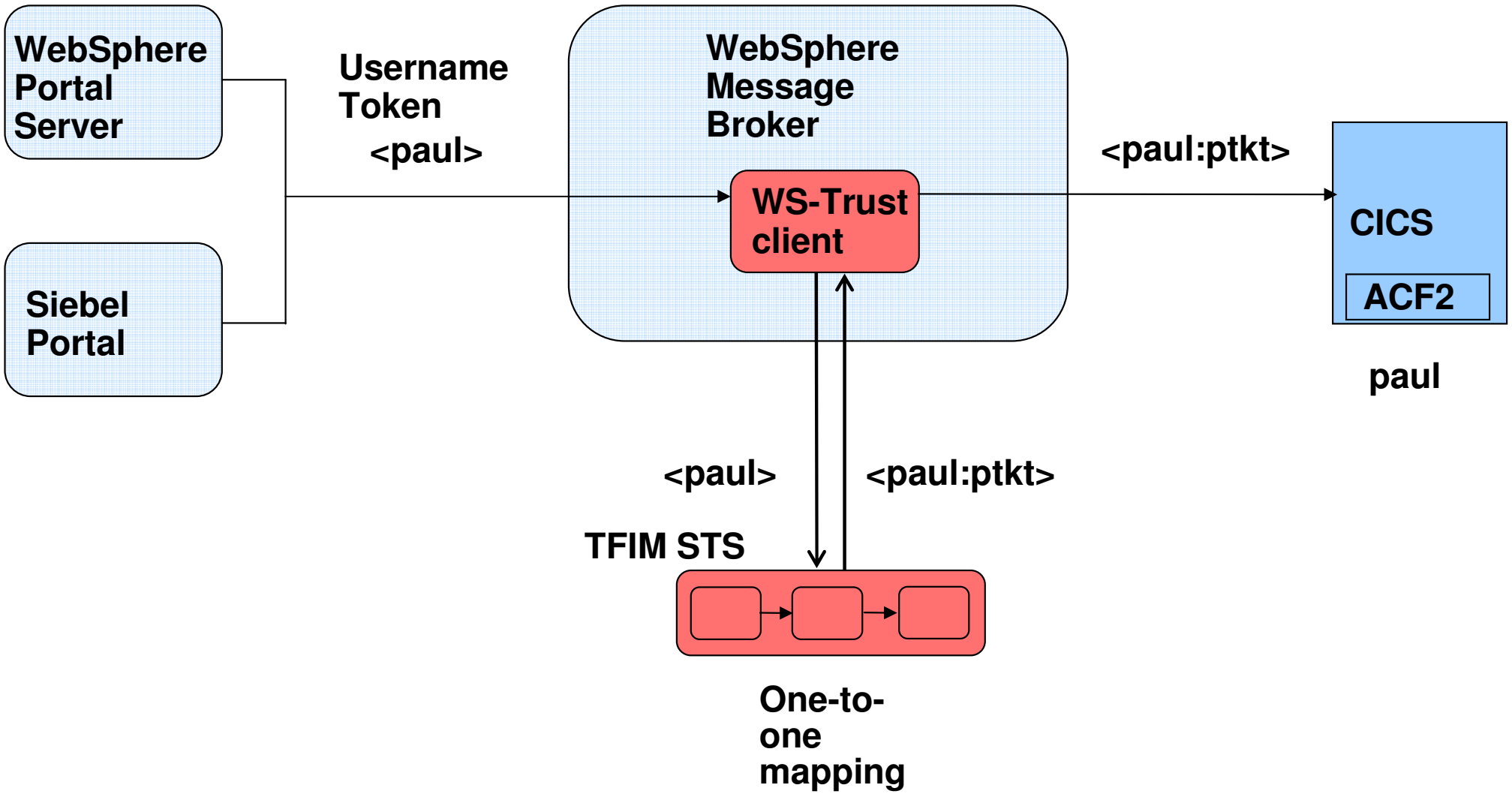
Customer Example

- **Client: An Immigration Agency**
- **Country: Asia Pacific**
- **Industry: Government**
- **TFIM Use Case: Identity Service**
- **Current state of deployment, details of timelines (production/pilot/PoC):**
  In production since early 2007

Business Requirement

- **The business requirement is to create a new front end for the immigration agency staff.**
- **Instead of accessing a variety of legacy mainframe applications using terminal style interfaces, the staff should have a consolidated view via a Portal interface.**
- **In the current deployment the users are all employees (6000+). Later, Internet users (millions) may be added.**
- **The web service "enabling" of the legacy systems requires a new approach for propagating and mapping identities.**

Architecture (brief description + picture)



Today's Challenge: *How to apply entitlements consistently?*

The business *requirement* is to protect access and disclosure of client and customer PII
(e.g. Client Transaction, Patient Records, Financial Results)

Security Architect

Internal Tool

Translates it as need to encrypt that information in all services using message security *policy*

Security Officer

Corporate Intranet

App Owner

Eclipse

Translates it as application specific data entitlement *policy*

IT Operations

IAM Console

Translates it as configurations and tool-specific access *policy*

How can customers demonstrate compliance back to the business?

IBM Tivoli Security Policy Manager (TSPM) provides the ability consistently define, manage and enforce entitlements across the enterprise

The business *requirement* is to protect access and disclosure of client and customer PII
(e.g. Client Transaction, Patient Records, Financial Results)

Security Architect

TSPM Policy Design Tool

Security Officer

Corporate Intranet

App Owner

TSPM Policy Server / Console

IT Operations

TSPM
TSPM
TSPM Runtime Security Service

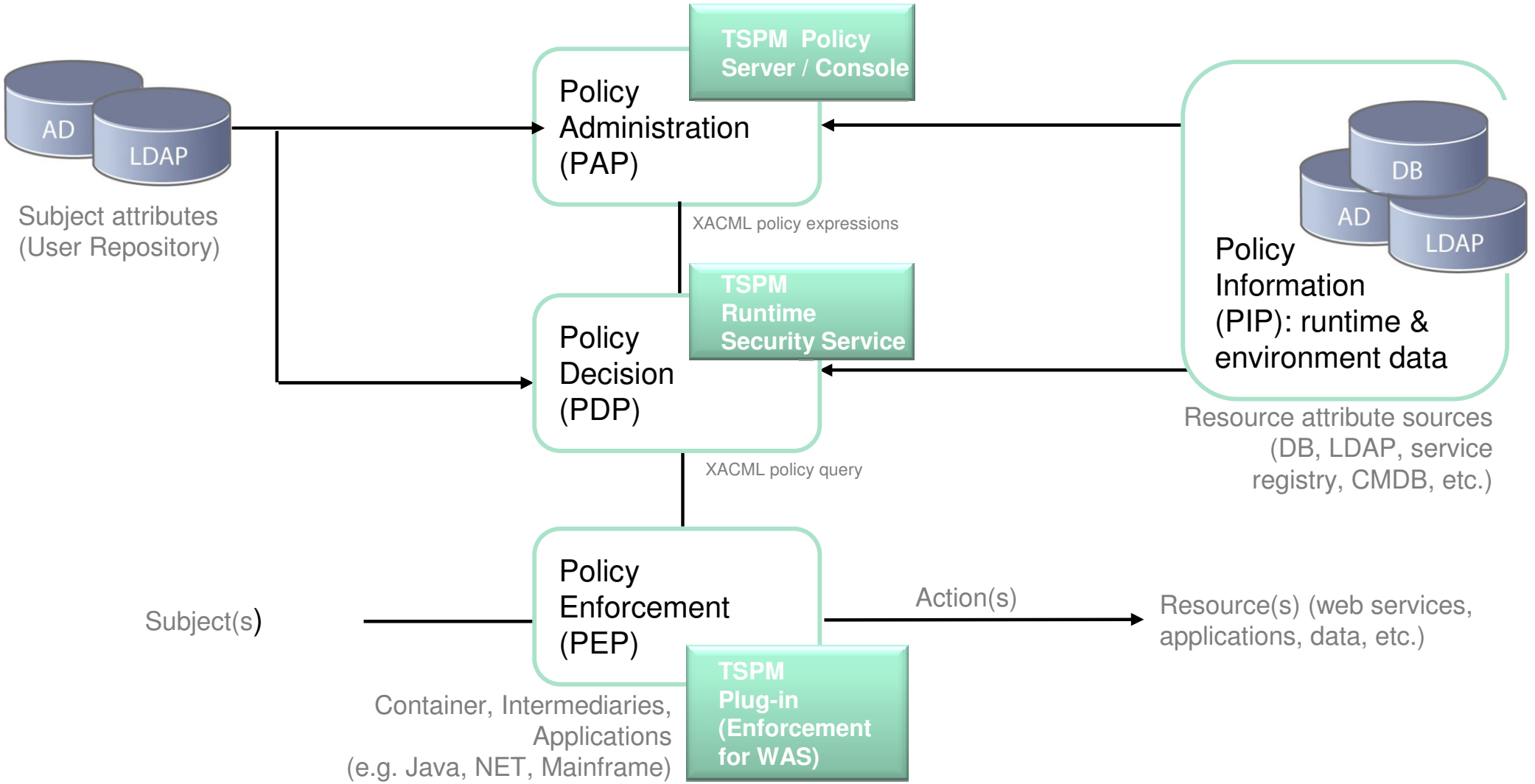
Demonstrate Compliance and Enable Identity Governance

Sample Policy: Who can Access to Approve a Funds transfer?

TSPM provides ability to author entitlement based on one or more conditions

- **Role**
  - The user must be in the `tfr_approver` role
- **Service attribute**
  - The transfer amount must be less than the maximum transfer limit for the type of transfer
- **User attribute**
  - The transfer amount must be less than the maximum transfer limit for the user
- **Relationship**
  - The user must have been assigned responsibility for the source account.
- **Environment**
  - The transfer must be made during business hours and from the corporate network
- **Request/Session Context**
  - The user must have authenticated using 2-factor authentication
- **Other Decision Engines**
  - The transaction must pass the criteria checked by the Fraud Detection system

TSPM Enables Application Owners to Easily Implement Entitlements for New Applications

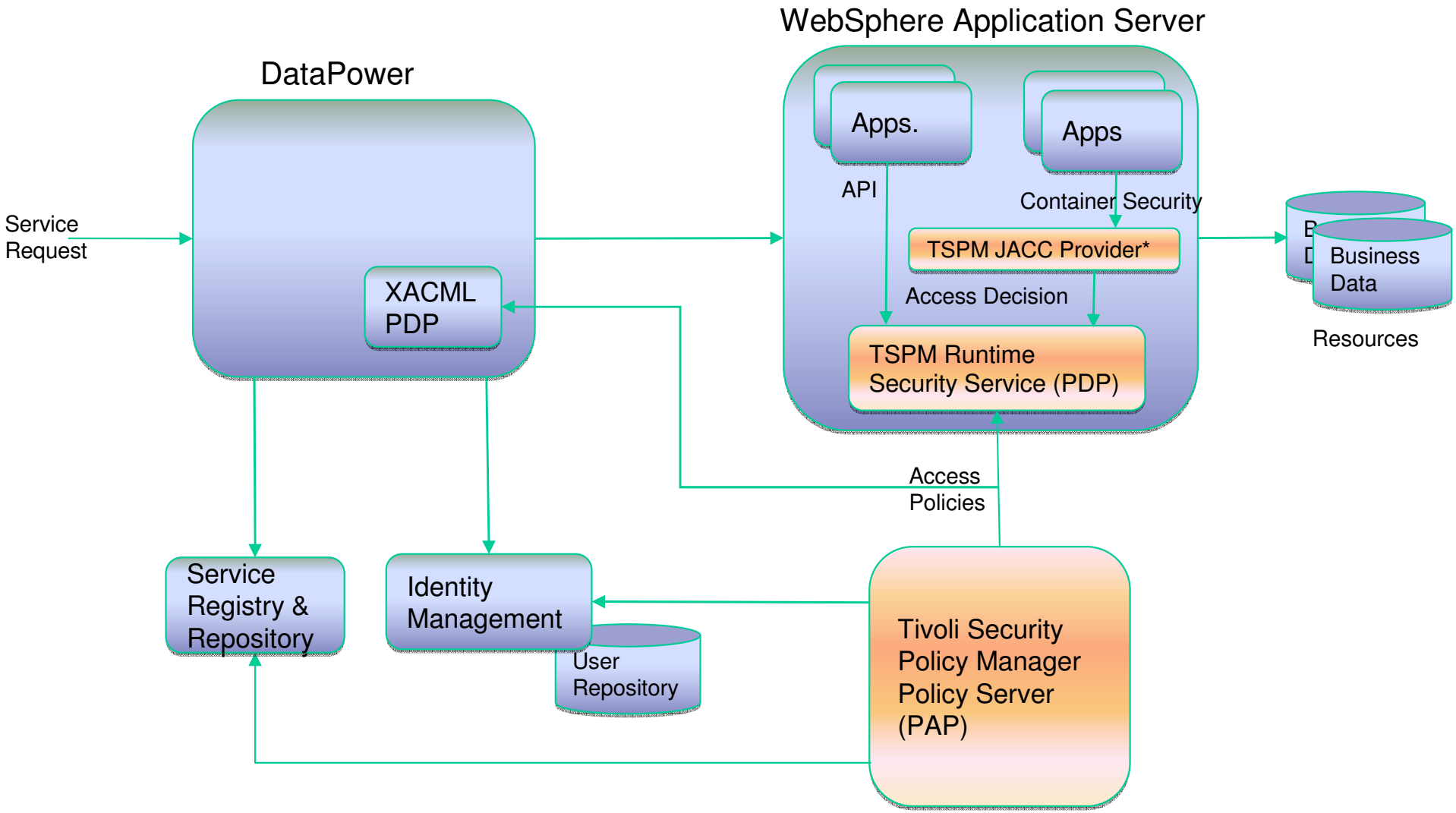


Customer Example

- **Background**
  - Government agency
  - New call center application deployment
  - Primarily Java environment including WAS
  - Existing IAM solution with Tivoli Identity and Access Manager
  - DataPower for XML firewall and web services security

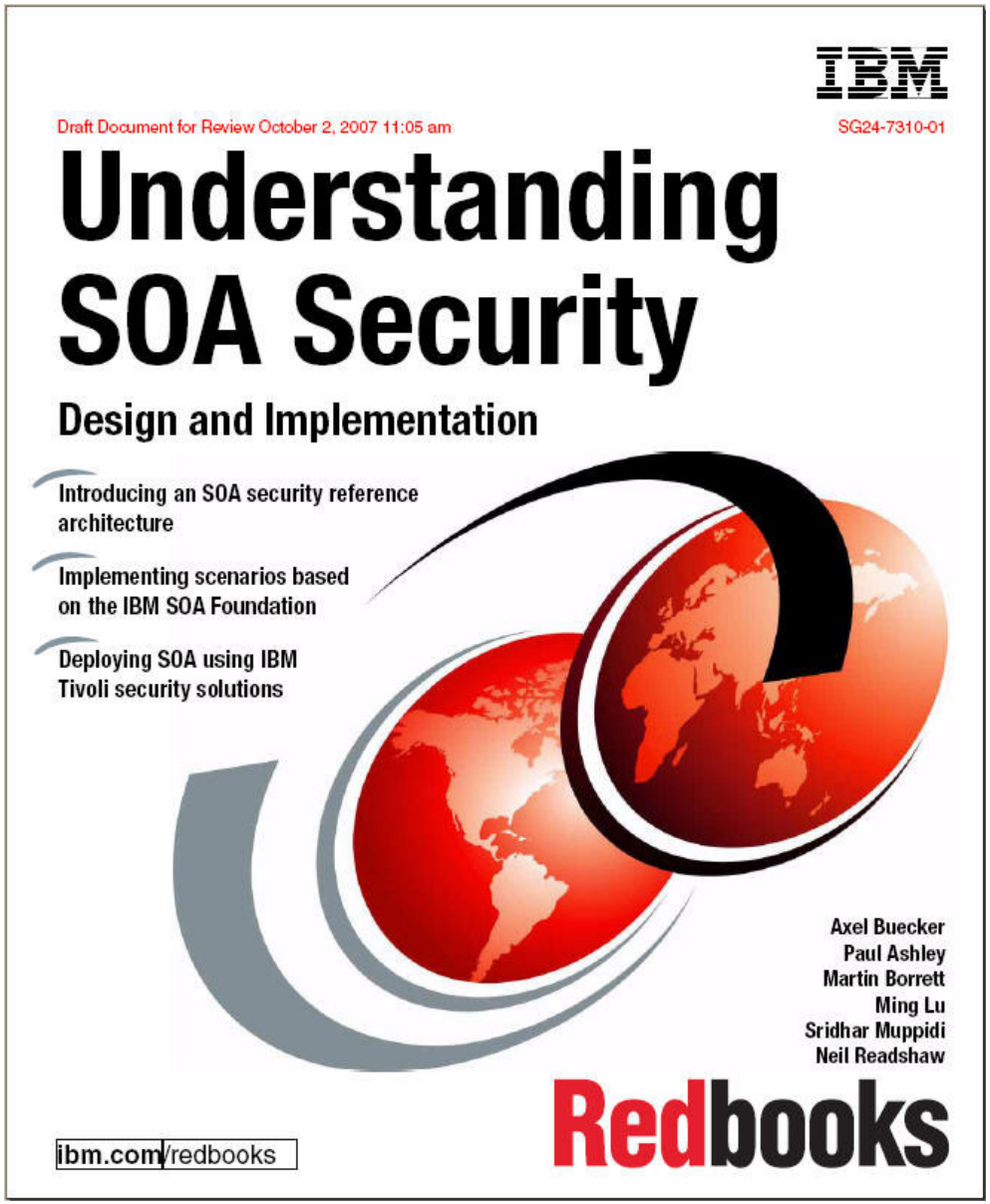
- **Challenges**
  - Need to address compliance concerns and privacy data security
  - How can we provide fine-grained access control including data-level access to employees, contractors and 3rd party partners?

Solution Approach – Government Agency Entitlements



* Currently in beta

http://www.redbooks.ibm.com/redpieces/abstracts/sg247310.html?Open



Summary

- **Issues well understood**
- **Standards and adoption of standards continues to mature**
- **Technology exists today to address security issues**
- **Customers are making progress in meeting business/security requirements**
- **IBM can help**

SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

SOA ARCHITECT SUMMIT

Turn your ideas into practical solutions.

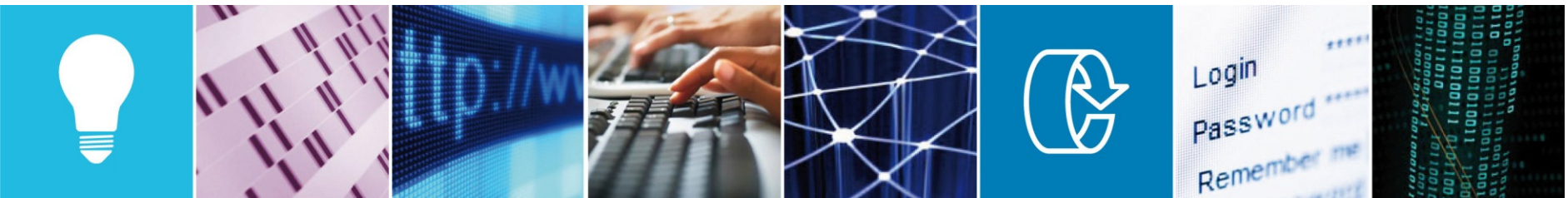
IBM

Recycling SOA Services & Artefacts

9th September 2009

Reviews reuse strategies and opportunities; establishes a rational basis for reuse decisions; recommends an approach and critical factors.

Richard Whyte *CEng,FIET, CITP,FBCS*



Reuse, Recycle: building from existing parts

- **Strategies and Opportunities**
  - Types of reuse
  - Barriers to reuse
  - Anti-patterns
- **A rational basis for recycling**
  - Organisation
  - Building from existing parts
  - Critical success factors
- **Recommendations**

An error cannot be believed sincerely enough to make it a truth.
(Robert Green Ingersoll 1881, foremost orator and political speechmaker of late 19th century America)

Reuse strategies and opportunities

- **Reuse is a tactic to achieve an objective**
  - Technical: *Consistency, Quality, Simplicity*
  - Project: *Risk, Quality, Skills, Duration*
  - Business: *Improve time to value, Reduce cost*

- **It can lead to**
  - Poor quality “patched together” solutions
  - Components & code that don't contribute to the solution
  - High dependency environments
  - Expensive, delayed, complex “future proof” solutions

Types of reuse

- **Reuse is referencing an asset or taking a copy and making changes**
  - "By reference" (Runtime)
    - Changes affect everyone
    - Challenges for impact analysis, testing, upgrades, etc
    - Creates dependencies and version control challenges
  - "By value"
    - Creates a plethora of components with subtle differences
    - Challenges for impact analysis, testing, upgrades, etc

- **A common target architecture reduces change to assets**
  - Alignment of assumptions reduces mismatch
  - Common performance and availability standards

Barriers to reuse

Operational	Leads to	Causing
Inefficient asset catalogue	Finding / understanding	Difficult to add components to register: Duplication
Charge-back and rewards	Rewarding individuals / projects	Complexity and adoption cost
Culture	Value invention over reuse	Ad-hoc approach to reuse
Process	Inefficient harvesting, **indexing**, **documentation**, testing	Added administrative costs
Information	What is available, how is it used. What SLA	Penalty for success! Your load on my server
Organisation & Investment		
Technical	**Leads to**	**Causing**
Design	Inconsistent architecture containing exceptions	Limited “by reference” reuse opportunity
Lack of encapsulation	High cohesion/dependencies	Incomplete interfaces / Key data not available
Different assumptions	Incomplete implementation significant mediation	Additional “hidden” costs of reuse
Inconsistent Standards	Security, Latency, Availability, Accuracy, Retirement date	**High maintenance**
Mismatch of Non-functional		

Service Oriented Architecture delivers

- **Isolation**
  - Reduce impact of change.
  - Avoid “long range” cohesion across the enterprise
  - Provided by published interfaces and routing through a mediator
- **Encapsulation**
  - Reduce impact of build decisions, different technologies and standards
  - Principle of independent implementation
- **Specialisation**
  - Specialist components (Services) with “simple” interfaces are more likely to be reused
  - System “stuff” provided by mediation: audit, security, routing, data management
- **Billing and metrics collection** (Not always included)
  - Simple to add to SOA compared to other styles
  - Provides a way to fund common components and improve shared infrastructure

SOA combined with strong governance provides the basis for removing barriers and fostering a reuse culture

Reuse can be an excuse for excess

- **Building components for all eventualities**
  - Creating many layers “just in case”
  - Adding fields that “might be useful”
  - Insisting on a technology because “it is more useful in the future”
  - ***Fine grained interfaces !***

- **Spending time and effort building for “undefined” reuse**
  - Undermines the current project and reduces the benefits
  - The architecture and governance MUST enforce characteristics for reuse
  - Humans have a poor record at predicting the future
  - Additional features are rarely tested so cannot be relied on

- **So**
  - Stick to the principles; Isolation, Specialisation, Encapsulation to allow change
  - Build only what is needed for today
  - Remember the objectives: Cost, Time,

Conclusion: SOA creates opportunity to reuse

- Concepts, principles, **experience**, and skills
- Documents, designs, standards, plans, templates
- Test harnesses, test data, and runtime environments
- Business continuity, transaction control, availability

- (Runtime) Services
- (Build time) code libraries, templates, tests, concepts

> **But there are pitfalls ☺**

A rational reuse strategy is based on

- **Clear Objectives**
  - What is to be achieved through reuse
  - How it will be measured in terms of business benefit
- **Consistent Architecture**
  - What characteristics are necessary from the architecture
  - How will harvesting operate
  - Are there clear and appropriate rewards
  - A funding model for shared components
- **Strong Governance**
  - Reuse does not trump structure, requirements, isolation, or specialisation
  - How are decisions made
- **Supportive Culture**
  - Design against known artefacts
  - Clear ownership and open discussion
  - Peer reviewed intentions

Recommendations: Strategy

- **Use SOA to provide a consistent design canvas**
  - A consistent bar for assumptions
  - Disaster recovery, scalability, interface structure, unique keys
- **Centrally control and manage reuse**
  - Products - RUNTIME services and documents provided with support
  - Assemblies - BUILDTIME documents and libraries provided by reference
  - Components - BUILDTIME code and “as-is” artefacts brovided by value
- **Decide how to meaningfully measure success**
- **Change the organisation**
  - Reward reuse and creation of reusable assets
  - Create an owner responsible for support, maintenance and modification of reusable assets
    - Promotes reuse and has an overview
- **Clear and strong governance**

Recommendations: Organisation

- **Central control**
  - Asset repository?
  - Website and support group?
- **Measuring Success**
  - Number of consumers? Website audience, Throughput
  - “Usefulness”: Developer satisfaction
  - Business measures
    - Time to value
    - Cost profile and benchmarking

Recommendation: Governance

- **Standards**
  - A small set of “Mandatory” characteristics
  - Not best practice or style guides
  - Only naming where names are critical to the design
    - Example: All communication through the ESB
- **Best practice**
  - We (the establishment) prefer this approach but its not mandatory
    - Example: All error logs should be circular
- **Style guides**
  - Naming and coding layout preferred styles etc.
    - Example: Indentation

Design Tactics: Isolation, specialisation, encapsulation

- **Architecture and principles foster reuse**
  - Re-factoring is always possible
  - Reuse does not drag a lot of baggage
  - Interfaces can be extended to accommodate new requirements

  – Service orientation
    - "Service" is the unit of reuse, version control, regression test, and documentation
    - "declarative" Interfaces

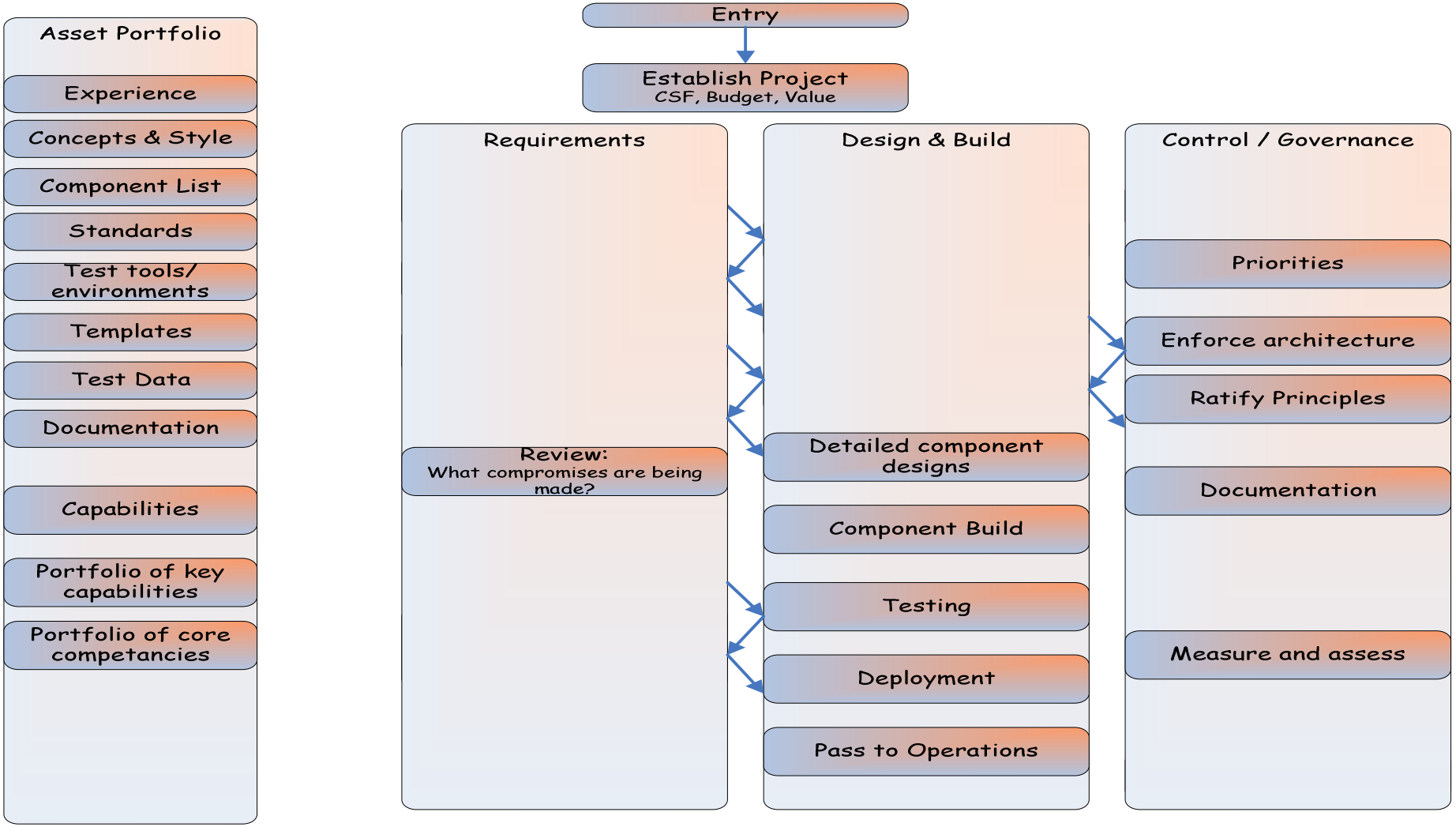
- **Design for the problem not the future**
  – Follow the rules but don't add function for possible future use
  – Responsibility of the architecture to foster reuse; not the programmer
  – Test all functions available. The fewer there are the more cost effective this is.

- **Design interfaces to hide complexity and be extensible**
  – Feature switches
  – Fine grained interfaces leak implementation and complexity

Critical Success Factors

- **Reuse is a tactic to achieve an objective**
  - Be clear about objectives
  - Don't spend time building for "undefined" reuse
- **Be prepared to pay for reuse**
  - Reusable assets need an owner (Maintain ownership?)
  - Keep a core team together; experience is an important reuse.
  - Reuse includes concepts, plans, documentation, etc.
  - Scaling and managing the cost of reuse, maintenance, upgrades
- **Design first then refine for reuse.**
  - Compromising SOA principles reduces reuse opportunities
  - Reuse using reference and value appropriately

A methodical checklist (reusable)



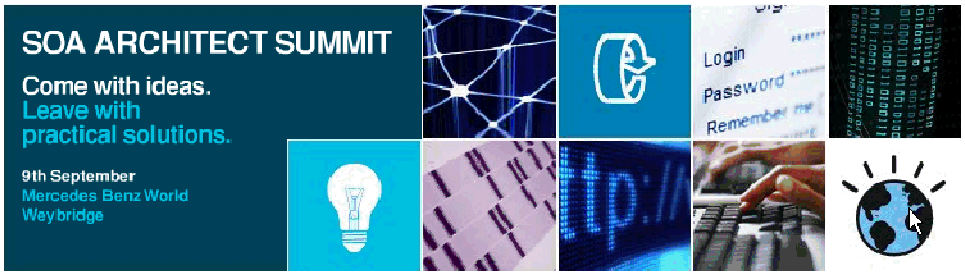
Finally

Reuse without a plan,
on a zero budget,
with no rules,
delivers to expectation

Reuse and recycling have very similar meanings.
This presentation introduces the idea of recycling to convey that parts can be wholly recycled, reused, or refined to make them useful.

SOA Architect Summit Next Steps

- **Summit Presentations at:**
  ibm.com/software/uk/itsolutions/soa/soa-summit



- **Building SOA Solutions**

– An IBM Proof of Technology

6th – 8th October in IBM Hursley

ibm.com/itsolutions/uk/soa/building-soa-solutions-pot/

- **SOA Forum**
  – please join & participate
  soaforumuk.com

- **European WebSphere Technical Conference, Hamburg, Germany**
  European WebSphere Technical Conference

Backup Notes: The cost of reuse

- Must adhere to standards and interoperability needs
- Must maintain backward compatibility
- Regression testing

- Management of repositories
- Maintenance and scaling of shared resources
- Organisation

Reuse artefacts by project phase

- Map reuse by project phase