Building the future of banking services

Banking industry Architecture Network (BIAN)

Presented by:
Parimah Mohammadpour (New Product Development (NPD) Department Manager at Sadad Informatics Corporation)

3rd National Conference on Advances in Enterprise Architecture
Sharif University of Technology
November 12th, 2019
Contents:

- A Review of the Prospects for the Future of Banking Industry
- Some of the important Standards in Banking Industry
- Moving towards Enterprise Architecture with a Service-Oriented Approach
- BIAN Introduction
- Design Principles & Techniques in BIAN
- BIAN's Organization and Content Development Approaches
- Applying the BIAN Standard in Banking Industry
- Overview of BIAN Digital Repository
- Moving Towards Service-Oriented Enterprise with BIAN Approach
A Review of the Prospects for the Future of Banking Industry
Driving Forces and Current Challenges In Banking

**FORCES SHAPING BANKING**
- Legacy bank systems
- New regulations
- CARL (compliance, audit, risk, legal)
- Cost of technology spend

**CHALLENGES TO BANKING**
- Rapidly evolving expectations
- Retail digital customer experience shapes expectations
- New competition (e.g., from nonbanks)
- Unlevel regulatory playing field
# The Profound Darwinian Shakeout

<table>
<thead>
<tr>
<th>2013-2016</th>
<th>2017-2021</th>
<th>2022-2025</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Digital adoption in consumer applications raises expectations for convenience everywhere</td>
<td>- Nonbank and fintech providers move upmarket to wholesale</td>
<td>- Digital banking is mainstream</td>
</tr>
<tr>
<td>- Rise of fintechs and nonbanks</td>
<td>- Products increasingly purchased from specialists due to value and convenience</td>
<td>- Transformation of leading banks' cultures to &quot;tech companies with a banking license&quot;</td>
</tr>
<tr>
<td>- Regulatory playing field not level - CARL (compliance, audit, risk, legal) issue</td>
<td>- Economic choppiness removes many online lenders; fraud sinks less secure payment providers</td>
<td>- Specialized knowledge of clients' industries translated into distinctive advice; capabilities remains a differentiator</td>
</tr>
<tr>
<td>- Large banks invest in digital capabilities primarily for retail</td>
<td>- Increasing regulation of nonbanks begins to level playing field</td>
<td>- Subscale banks can't compete, valuations impaired</td>
</tr>
<tr>
<td>- Most banks feel &quot;analog&quot; in an increasingly digital world</td>
<td>- Trust and digital security increasingly important</td>
<td>- Major banks continue to acquire bank and nonbanks to fill in product, technology or geographic gaps</td>
</tr>
</tbody>
</table>

---

Greenwich Associates

Sadad Informatics Corporation

New Product Development Group
The Profound Darwinian Shakeout

### 2013–2016: Rise of Fintech and Regulatory Inconsistency
- Digital adoption in consumer applications raises expectations for convenience everywhere
- Rise of fintechs and nonbanks
- Regulatory playing field not level – CARL (compliance, audit, risk, legal) issue
- Large banks invest in digital capabilities primarily for retail
- Most banks feel “analog” in an increasingly digital world

### 2017–2021: Rise of Specialists and Great Digital Divide
- Nonbank and fintech providers move upmarket to wholesale
- Products increasingly purchased from specialists due to value and convenience
- Economic choppiness removes many online lenders, fraud sinks less secure payment providers
- Increasing regulation of nonbanks begins to level playing field
- Trust and digital security increasingly important
- Massive investment as leading banks build and acquire digital capabilities, others left behind

### 2022–2025: Rise of Digital Banking Superstores
- Digital banking is mainstream
- Transformation of leading banks’ cultures to “tech companies with a banking license”
- Specialized knowledge of clients’ industries translated into distinctive advice; capabilities remains a differentiator
- Subscale banks can’t compete, valuations impaired
- Major banks continue to acquire bank and nonbanks to fill in product, technology or geographic gaps
Rise of Specialist Providers

As banks lag on digital, convenience and value, specialists seize the opportunity to fill the gaps
## The Profound Darwinian Shakeout

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital adoption in consumer applications raises expectations for convenience everywhere</td>
<td>Nonbank and fintech providers move upmarket to wholesale</td>
<td>Digital banking is mainstream</td>
</tr>
<tr>
<td>Rise of fintechs and nonbanks</td>
<td>Products increasingly purchased from specialists due to value and convenience</td>
<td>Transformation of leading banks’ cultures to “tech companies with a banking license”</td>
</tr>
<tr>
<td>Regulatory playing field not level – CARL (compliance, audit, risk, legal) issue</td>
<td>Economic choppiness removes many online lenders, fraud sinks less secure payment providers</td>
<td>Specialized knowledge of clients’ industries translated into distinctive advice; capabilities remains a differentiator</td>
</tr>
<tr>
<td>Large banks invest in digital capabilities primarily for retail</td>
<td>Increasing regulation of nonbanks begins to level playing field</td>
<td>Subscale banks can’t compete, valuations impaired</td>
</tr>
<tr>
<td>Most banks feel “analog” in an increasingly digital world</td>
<td>Trust and digital security increasingly important</td>
<td>Major banks continue to acquire bank and nonbanks to fill in product, technology or geographic gaps</td>
</tr>
</tbody>
</table>
Leading Banks Become Digital Financial Superstores

Consumers will soon have access to all of their financial needs from one central location.
Banking is Entering a Profound Period of Change

Clients’ expectations of what they want from their banking provider are shifting

“The only thing standing between banks and giants like Google, Apple and the Telecom providers will be a banking license”
Transformation Towards Digital Banking

- **Banking Clients 2025**
  - The banking client 2025 is empowered by digital bots and digitally assisted client advisors.

- **Banking Operating Models 2025**
  - Banking operating models 2025 will be characterized by intelligent automation, cooperation, and industrialization.

- **Banking Revenue Models 2025**
  - In 2025 we will see entirely new revenue models in banking.

- **Digital Banking Platforms 2025**
  - 2025 banking platforms will be open and interoperable and designed from front to back.

- **Data-driven Banking 2025**
  - Data-driven banking will allow banks to develop entirely new business models and products and optimize their processes.

- **Banking Value Chain 2025**
  - There will be no isolated banking value chain in 2025; instead we will see new cross-industry ecosystems.
Digital Banking Models 2025

Disruptive

Digital Blockchain Bank 2025

Digital Ecosystem Bank 2025

Back Office

Innovation

Area

Front Office

Digital Custody and Transaction Bank 2025

Digital Advisory Bank 2025

Incremental
What is a digital bank? (Temenos’s Definition)

At Temenos, we believe that being truly digital means enabling “experience driven banking”. This needs to cover both the customer experience and the execution experience. A digital bank offers customers contextualized, seamless experiences that transform the customer journey. And becoming a digital bank means delivering a compelling and relevant customer and execution experience through an open, integrated and flexible architecture. True digital banking can be condensed into two key and distinct factors:

- **Customer Experience** The sum total experience that enables customers to self-serve, in real time, via multiple devices, with environmental context that results in a personal and relevant experience. This requires online access to all products and services as well as the real-time customer intelligence to be able to provider relevant, contextualized and personalized content and offers at the right time and on the right device.

- **Execution Experience** The sum total experience that enables organizations to deliver on-demand services with minimal human involvement via straight-through-processing whilst enabling internal bank users to serve clients via offline channels and continuously improve products and processes. This requires an end to end digital platform and architecture.

The fundamental digital equation: \( A + B + C = D \)

- **A** = Anytime, anyplace, any channel – this is what customers expect.
- **B** = Better banking, beyond the traditional banking service – using customers’ data to become a virtual advisor
- **C** = Contextual – the service, communication, rewards and products you offer to meet customers’ expectations, needs to be driven by data and analytics and personalized to their requirements.
- **D** = Digital banking
Four Pillars of Digital Banking

- **Omnichannel Banking.** The streamlining and integration of channels to ensure a positive and seamless customer journey across all potential touch points.
- **Modular Banking.** A systems architecture that has interchangeable components that can react to market and institutional changes quickly.
- **Open Banking.** The ability to use open APIs to connect internal and external capabilities, building experiences that may extend beyond banking services.
- **Smart Banking.** The use of advanced analytics to leverage data for personalized engagement and experiences.
Major Financial Trends Impacting Banking

- According to researches from Atos, the four most transformational challenges and opportunities for the future of banking through the next 5 years include:

  - **Response to customer needs**: Ranked as the most important trend in each of the last 4 years in research done by the Digital Banking Report financial institutions need to shift from physical interactions to digital engagement. For banks and credit unions that **digitize customer journeys**, there can be a significant benefit in revenues, cost reductions and customer satisfaction.

  - **Optimization of costs**: Because of the efficiencies of digital-only competition, banks and credit unions will need to consider divesting from non-core operations and leveraging intelligent automation. In addition, organizations will need to reinvent back office processes and replace aging infrastructure.

  - **Creation of new revenue streams**: Open banking and the use of APIs will open new opportunities for both cost reduction and revenue growth. As the banking ecosystem expands beyond traditional banking services, new products will be developed and segments served that will provide differentiated offerings and monetization opportunities.

  - **Development of security and compliance systems**: With customer data becoming a ‘product’ for many financial institutions, the need for enhanced security and advanced insights (AI) will become a differentiator from both a compliance and customer trust perspective. This can lead to reduced costs and potential business growth.
10 Technologies That Will Disrupt Financial Services In The Next 5 Years

- As opposed to technology taking a secondary position, supporting only the processing of transactions, future technologies will be more customer-centric and efficient, and provide more targeted, secure and intelligent solutions. With technology as the driving force in the future, organizations will be able to redefine themselves to be more competitive and responsive to marketplace needs.

- **Atos** developed a very helpful Global Banking Technology Radar that provides a perspective on the technologies anticipated over the next five years, the business impact of the technologies and the timing of integration.
Banks are Responding in three different ways, Which one(s) are you adopting?
Factors to Consider When Choosing a Digital Strategy

**Front end only**
- Sufficient technology to build out a self-service model for customers, regardless of channel
- People and processes to create consistent customer maps across all channels and maintaining digital interfaces

**Wrap and digitize**
- APIs allowing interfacing between all IT assets and a data management solution
- A legacy core that is substantial enough to warrant additional investment and personnel to support
- Incorporation of customer analytics
- Ability to create a consistent omni-channel experience for the customer
- Expertise in transformation, integration, and automation projects

**Go digital native**
- Digital native core with IT trained to support
- Partnerships with third parties to help integrate all of the needed tech
- Training focused on the digital changes
- Digital marketing and deep analytics
- Determining which IT capabilities to build, buy, and natively own
- Farm out those capabilities that cannot be developed in house if cost prohibitive
Some of the Important Standards in Banking Industry
Some of The Important Standards in Banking Industry

- Banking Major Standard Category:
  - ISO
  - BASEL
  - IFRS
  - FSAP
  - CFT

- Some of The Important Samples are:
  - BIAN (Banking Industry Architecture Network)
  - AMQP (Advanced Message Queuing Protocol)
  - Banking ISO Standards: 20022, 9564, 7816, 7810, 8583
  - XBRL (Extensible Business Reporting Language)
  - FIBO (Financial Industry Business Ontology)
  - FIX (Financial Information eXchange Protocol)
  - LEI (Legal Entity Identifier)
  - FPLM (Financial products Markup Language)
  - MDDL (Market Data Definition Language)
  - EMV
  - PCI DSS (Payment Card Industry Data Security Standard)
  - ........
Moving Towards Enterprise Architecture with a Service-Oriented Approach
Banking Architecture

What is Architecture?

- The fundamental organisation of a system embodied in its components, the relationship between these components and the environment and the principles governing its design and evolution.

What is Banking Architecture?

- The fundamental organisation of a BANKING SYSTEM, embodied in its components, the relationship between these components and the environment and the principles governing its design and evolution.

So what are these components?

- Business Architecture
- Information Architecture
- Application Architecture
- Technology Architecture
What is Enterprise Architecture and it’s Layers?

- Processes
- Information Needs
- Application Components
- Networks
- Application
- Information
- Business
- Goals, Strategies
- Org. Structure
- Databases
- Hardware
BIAN Architecture Layers

**Business Architecture**
- Service Domain Definition
- Service Landscape
- Business Scenario
- Business Capability Model
- Wireframe Model

**Information Architecture**
- Business Object Model
- Asset Decomposition Model
- Control Record

**Application Architecture**
- Application Capabilities
- Logical System
- Message Format
- Open API

**Technology Architecture**
- Service Protocol Standards
Moving towards Enterprise Architecture in Banking with a Service-Oriented Approach

Enterprise Architecture is the central tool for enabling the transformation process of both Business and IT

Challenges

- Lack of an overall, consolidated and accepted perception of current and target state;
- Actual impact of architecture on decision-making at various levels of the organization;
- A significant set-up of “custom built” assets within the architecture function;
- Limited adoption of standards in some areas.

Mitigations

- In terms of architectural work, TOGAF can establish a shared approach and vocabulary
- In terms of content, BIAN can serve as reference in a banking environment
Why BIAN & TOGAF?

Significant challenges in the banking Industry

Transformational changes are inevitable

Enterprise Architecture is key in enabling change

Bring together two industry frameworks for content and methodology to speed up work and improve quality and consistency
BIAN – A Business Service Model for the Banking Industry

Vision

Industry-wide consensus regarding SOA for banking systems, aligned with the business objectives of agility and reduced cost

BIAN Service Landscape

BIAN Meta Model

BIAN Service Domain Definitions

Community of banks, service providers and software vendors
Leveraging BIAN Deliverables with TOGAF

- Consider BIAN as framework
- (part of) Overall BIAN Service Landscape
- Service Landscape (Business areas & Business domains)
- Service Domains (Reusable building blocks related to application components and data entities)
- Service Landscape to assess compliance of external packages
BIAN Introduction
BIAN Introduction

- **Introduction**
  - BIAN Introduction
  - BIAN Organisation
  - The Problem BIAN Seeks to Solve
  - Process Model Vs Capability Model

- **BIAN Artifacts**
  - How to Guide Series
  - Digital Repository
  - BIAN Release
  - BIAN Overview
Introduction

BIAN

- BIAN is a global, open, independent and unique community where banks, software providers and system integrators are collaborating to define a common yet exceedingly flexible SOA framework for the banking industry with the goal of establishing a common language.
Introduction

BIAN Organisation

- Founded in 2008, the Banking Industry Architecture Network (BIAN) is a global, not-for-profit organization that seeks to develop standard Service Landscape and Semantic IT Service (A2A) Definitions for the Banking industry. BIAN will enable the next generation of banking industry solutions developed either in-house or commercially:

  - By leading banks sharing their requirements for banking services
  - By leading software and services vendors to implement them based on standard semantics
Introduction

BIAN Is a Member Driven Organization

Financial Institutions

![List of financial institutions]

Software Vendors / Service Providers

![List of software vendors and service providers]
The Problem BIAN Seeks to Solve

Design combines two perspectives – what is it made of, how does it behave…

<table>
<thead>
<tr>
<th>‘INGREDIENTS’</th>
<th>‘BEHAVIORS’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of Buildings</td>
<td>Work &amp; Education</td>
</tr>
<tr>
<td>Parks &amp; Squares</td>
<td>Entertainment &amp; Sports</td>
</tr>
<tr>
<td>Transportation</td>
<td>Shopping</td>
</tr>
<tr>
<td>Utilities</td>
<td>Health and Services</td>
</tr>
<tr>
<td>etc…</td>
<td>etc…</td>
</tr>
</tbody>
</table>

Architects select the required ingredients and configure them to support anticipated behaviors to create a ‘town plan’

Business architects do not have the organizing equivalent of the town plan

Business Processes

With no easy way to capture and associate the ingredients with the anticipated behaviors there is no equivalent ‘enterprise plan’
The Problem BIAN Seeks to Solve

Without a coordinated view of ingredients & behaviors, business design suffers

- Every project was justified to meet a compelling and immediate need, but without the big picture to guide development, over time isolated development leads to chaos

  *A city where new construction is not coordinated with a town plan...*

  *An enterprise where application development is not coordinated with an enterprise plan...*

- Many Bank’s application portfolios suffer from such fragmentation. They contain a high level of redundancy. Worse, as most connections are unique, they present highly complex maintenance and enhancement challenges
The Problem BIAN Seeks to Solve

Once a plan has been developed its adoption can be incremental

- Advances in design techniques, technology and awareness mean that the migration to well architected solutions can be incremental

- An organization can migrate towards a well partitioned application portfolio
The Problem BIAN Seeks to Solve

With good design, the migration to service based solutions is incremental

- At the outset, the applications in a conventional ‘process based’ application portfolio can be drawn as a criss-cross network of overlapping processes.

**KEY:**
- Legacy systems capabilities

---

Sadad Informatics Corporation                New Product Development Group
The Problem BIAN Seeks to Solve

With good design, the migration to service based solutions is incremental

- Analysis of the BIAN Service Domains can be used to target those areas that are the main bottlenecks…
The Problem BIAN Seeks to Solve

With good design, the migration to service based solutions is incremental

- As the early Service Domains are stabilized behind more complete service boundaries, additional areas can be addressed
The Problem BIAN Seeks to Solve

With good design, the migration to service based solutions is incremental

- Over time more of the traffic is handled between the rationalized and stable service centers and more of the legacy is wrapped, replaced or rendered obsolete/replaced
The Problem BIAN Seeks to Solve

With good design, the migration to service based solutions is incremental

- Eventually the migration to a service based architecture replaces the significant majority of the portfolio. Operational capability re-use enables new business needs to be addressed with incremental changes rather than completely new stand alone solutions...
The Problem BIAN Seeks to Solve

Process models are well established for application development

- Process based views of business activity have been used very effectively

Five established ways to leverage a process perspective

- Eliminate redundant steps
- Automate manual tasks
- Reduce number of variations
- Do more things in parallel
- Leverage shared services

But the very flexibility of process modeling limits how it can be used to define canonical standards for service operations – the allowed variations mean only the most common procedures or commodity type activities can have a standard process definition
The Problem BIAN Seeks to Solve

Anything that can be modeled as a process can also be modeled using service centers

- For the monthly credit card billing cycle the process is broken down until the steps are fine grained enough for coding

**Monthly card billing process:** at month end a card holder’s outstanding balance is evaluated and a minimum payment and outstanding amount calculated. A billing statement is printed off and sent to the registered billing address. A period later the payment is either received in time or the account is placed into *past 30 day payment* state and further use of the card restricted

- Process analysis automates a tightly coupled sequence of “Input/Process/Output” (IPO) activities that can be implemented as software

Processing steps at the lowest level could be implemented as software elements/modules. They can be assembled to create a stand-alone/monolithic billing application

Choreography implemented as a dependent procedural flow of tightly coupled service couplets that ‘exist’ for the life of the transaction itself
The Problem BIAN Seeks to Solve

The same actions can be associated with specialised service centers

- The same activity remodeled using five collaborating service centered structures:

- Service center design removes the dependent ‘Input/Process/Output’ sequencing allowing for more flexible patterns of collaboration…

Choreography implemented as an asynchronous series of loose coupled service interactions between persistent Service Domains
The ‘sequence diagram’ perspective hides the highly ‘networked’ interactions

- When modeled this way the activity is one instance of a pattern of collaboration between the Service Centers (that are stable/static capabilities). The service centers connect as and when they need to in a loosely coupled service network.

- Many business activities are not well represented as a repeatable, sequence of predefined activities as implied by a process model. Some analysts have estimated as little as 5% of business behavior is well characterised using conventional process model designs.
The Problem BIAN Seeks to Solve

The content of the service operations and messages is greatly simplified

- Because each Service Domain performs a unique and discrete business role, and it ‘encapsulates’ its specific knowhow – access service operations are narrowly focussed and can be semantic. Consider the traffic in the earlier worked example:

Consider the make-up of the exchanges…

Choreography implemented as a dependent procedural flow of tightly coupled service couplets that ‘exist’ for the life of the transaction itself

Compare to the exchanges along a process ‘factory or workflow’ production line…
BIAN Artifacts

Service Landscape

Business Scenarios

Service Operations

How To Guides & Training Presentations

Reference Formats

Metamodel Guide

UML Based Repository & Tooling ISO20022 based

HTML Service Landscape & Business Scenarios

Web-tool Business Scenarios & Wireframes

Excel Model Extract

Being piloted:
BIAN Overview

Relating the BIAN business Architecture to underlying application/systems Architectures

BIAN Architecture Layers

- Business Architecture
  - Service Domain Definition
  - Service Landscape
  - Business Scenario
  - Business Capability Model
  - Wireframe Model

- Information Architecture
  - Business Object Model
  - Asset Decomposition Model
  - Control Record

- Application Architecture
  - Application Capabilities
  - Logical System
  - Message Format
  - Open API

- Technology Architecture
  - Service Protocol
  - Standards

BIAN - the link between business and technical architectures
Design Principles & Techniques in BIAN
Design Principles & Techniques in BIAN

- Service Domain
- Service Operation
- Service Landscape
- Business Scenario
- Wireframe
- Clustering
The building block of the BIAN standards is the Service Domain

- A Service Domain is a discrete and ‘elemental’ business capability that exacts or creates value by “doing something to something”

The Bank is made up of resources or “objects” that it can use…

...and functions it performs on those resources/objects

Exacting value through object use, or by maintaining/enhancing the object to increase its value creating potential
Service Domain

- Techniques for Defining a Service Domain
  - Asset Types
  - Functional Pattern
  - Generic Artifact
  - Behavior Qualifiers Types
  - Control Record
  - Features of Service Domain
  - Service Operations
Asset types

- A high level decomposition of the Objects. Any bank has a collection assets that it can own or have some influence over e.g. a customer relationship, cash, or a payment facility. The asset needs to have an associated use or purpose.
Unique business context example building Vs business architecture

The ‘threshold of decomposition’ in business and building architecture
Functional Pattern

- BIAN has identified a number (18) of generic commercial behaviors that are applied to different asset types in the execution of business. Each Service Domain has a dominant Functional Pattern that defines its specific role.

<table>
<thead>
<tr>
<th>Functional Pattern</th>
<th>Description</th>
<th>Example(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECT</td>
<td>Define the policies, goals &amp; objectives and strategies for an organizational entity or unit.</td>
<td>Direct a business division of the enterprise</td>
</tr>
<tr>
<td>MANAGE</td>
<td>Oversee the working of a business unit, assign work, manage against a plan and troubleshoot issues.</td>
<td>Manage the day-to-day activities at a bank branch location</td>
</tr>
<tr>
<td>ADMINISTER</td>
<td>Handle and assign the day-to-day activities, capture time worked, costs and income for an operational unit.</td>
<td>Administer the time reporting and billing for the specialist sales support team.</td>
</tr>
<tr>
<td>OPERATE</td>
<td>Operate equipment and/or a largely automated facility.</td>
<td>Operate the bank's internal internet facility</td>
</tr>
<tr>
<td>PROCESS</td>
<td>Complete work tasks following a defined procedure in support of general office activities and product and service delivery functions.</td>
<td>Process the evaluation and completion of customer offers</td>
</tr>
<tr>
<td>REGISTER</td>
<td>Capture and maintain reference information about some type of entity.</td>
<td>Register customer reference details in a directory</td>
</tr>
<tr>
<td>DESIGN</td>
<td>Create and maintain a design for a procedure, product/service model or other such entity.</td>
<td>Create and maintain product designs and analytical models</td>
</tr>
<tr>
<td>DEVELOP</td>
<td>To build or enhance something, typically an IT production system. Includes development, assessment and deployment activities.</td>
<td>Build, enhance, test and deploy a major enhancement to a production processing system</td>
</tr>
<tr>
<td>ASSESS</td>
<td>To test or assess an entity, possibly against some formal qualification or certification requirement.</td>
<td>Perform regulatory tests on a proposed financial transaction; check a new offer conforms to an existing contractual agreement</td>
</tr>
<tr>
<td>MAINTAIN</td>
<td>Provide a maintenance service and repair devices/equipment as necessary.</td>
<td>Establish a maintenance and repair program covering the PC technology used in the central offices</td>
</tr>
<tr>
<td>TRACK</td>
<td>Maintain a log of transactions or activity, typically a financial account or a log of activity to support behavioral analysis.</td>
<td>Maintain a financial journal of transactions processed for a product; maintain a log of customer events and activity for subsequent analysis</td>
</tr>
<tr>
<td>ANALYSE</td>
<td>To analyze the performance or behavior of some on-going activity or entity.</td>
<td>Provide behavioral insights and analysis into customer behavior; analyze financial market activity in order to identify opportunities</td>
</tr>
<tr>
<td>MONITOR</td>
<td>To monitor and define the statusing of some entity.</td>
<td>Monitor the status and key indicators of a customer to influence on-line interaction; track the status of issued cards for security control</td>
</tr>
<tr>
<td>AGREE-TERMS</td>
<td>Maintain the terms and conditions that apply to a commercial relationship.</td>
<td>Define and maintain the terms governing the contextual relationship with a customer</td>
</tr>
<tr>
<td>ENROLL</td>
<td>Maintain an enrollment for some group or related collection of parties.</td>
<td>Administer the membership status of a syndicate of executors</td>
</tr>
<tr>
<td>ALLOCATE</td>
<td>Maintain an inventory or holding of some resources and make assignments/allocations as requested.</td>
<td>Track the inventory and administer the distribution of central cash holdings throughout the bank branch &amp; ATM network</td>
</tr>
<tr>
<td>FULFILL</td>
<td>Fulfill any scheduled and ad-hoc obligations under a service management, most typically for a financial product or facility.</td>
<td>Perform the scheduled (e.g. statements, standing orders) and ad-hoc fulfillment tasks (e.g. fund transfers) for a current account facility</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>Execute a well bounded financial transaction/task, typically involving largely automated/structured fulfillment processing.</td>
<td>Execute a payment transaction</td>
</tr>
</tbody>
</table>
Asset decomposition Excel extract

<table>
<thead>
<tr>
<th>Object</th>
<th>Sub-Type</th>
<th>Sub-Type</th>
<th>Sub-Type</th>
<th>Sub-Type</th>
<th>Mapped Functional Patterns</th>
<th>Mapped Service Domains</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>322</td>
<td>Intellectual Property</td>
<td>Knowhow</td>
<td>Product/Service</td>
<td></td>
<td>INVENTORY</td>
<td>Intellectual Property Portfolio</td>
<td></td>
</tr>
<tr>
<td>323</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FULFILL</td>
<td>Knowledge Exchange</td>
<td></td>
</tr>
<tr>
<td>325</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>326</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Product/Service**
- Product: DESIGN | Product/Service Design
- Product Bundle: DESIGN | Product/Service Combination Design

**Model**
- Financial: DESIGN | Quantitative Analysis
- Market Risk: DESIGN | Market Risk Models
- Instrument Value: DESIGN | Financial Instrument Valuation Models
- Credit Risk: DESIGN | Credit Risk Models
- Liquidity Risk: DESIGN | Liquidity Risk Models
- Business: DESIGN | Business Risk Models
- Behavioral: DESIGN | Customer Behavior Models
- Operational: DESIGN | Production Risk Models
- Rating: DESIGN | Operational Risk Models

**Procedures**
- Operational: MANAGEMENT

**Application**
- Technique/Method: PLAN | IT Architecture Standards & Guidelines
- Business Architecture: DESIGN | Business Architecture

**General Market**
- Market Insights: ANALYSE | Market Research
- Competitor insights: ANALYSE | Competitor Analysis
- Location: REGISTER | Location Data Management
Generic Artifact

➢ As functional patterns describe a behavior they typically take the verb form. The generic artifact for a functional pattern simply describes some form of tangible record or document that can be associated with the execution of the functional pattern.
Behavior Qualifiers Types

- Based on the Functional Pattern a ‘behavior qualifier type’ is defined and this is used to list behavior qualifiers specific to the Service Domain. The definition of behavior qualifiers is then used as necessary to clarify the working of the Service Domain and its offered services to provide more precision to their purpose.
End to end state for the functional patterns

These states only define the fairly simple externally visible states that the Service Domain may pass through. There will typically be far more detailed internal states for individual control records.

<table>
<thead>
<tr>
<th>Main Life-cycle States for Functional Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANAGE</td>
</tr>
<tr>
<td>ADMINISTER</td>
</tr>
<tr>
<td>OPERATE</td>
</tr>
<tr>
<td>PROCESS</td>
</tr>
<tr>
<td>DESIGN</td>
</tr>
<tr>
<td>ASSESS</td>
</tr>
<tr>
<td>TRACK</td>
</tr>
<tr>
<td>ANALYSE</td>
</tr>
<tr>
<td>MONITOR</td>
</tr>
<tr>
<td>ENROLL</td>
</tr>
<tr>
<td>FULFILL</td>
</tr>
<tr>
<td>TRANSACT</td>
</tr>
</tbody>
</table>
Control Record

- The execution of the Service Domain’s role for the full life cycle is tracked/managed using an instance of its ‘control record’. A Service Domain applies one pattern of behavior (functional pattern) to one asset type. Its control record combines the functional pattern’s generic artifact with the asset type.
Service Domain

The discrete business functional capacity partition represented by a Service Domain can be considered as being broadly equivalent to an organizational unit of the enterprise that combines the ‘people, process and technology’.
Features of a Service Domain

Some defining Service Domain characteristics:

- A unique business purpose
- It is elemental
- Collectively comprehensive
- Has a ‘Control Record’
- Full Life-Cycle support
- Single or Multiple Instances
- Short or Long Life-Span
- Service Based
Service Domain Broken into a Functional Core and Service Wrapper

Differentiating between the core functions and the service ‘boundary’
Rescoping a BIAN Service Domain

Service Domains Too Specific
- European Equities
- US Equities

Service Domains Too General
- Options
- SWAPS
- Derivatives

Pick the level

Shared Solutions

Enterprise Alignment

Business Ownership

Agent
Broker
Custodian
Correspondent
Regulator

External Party
Alliance Partner
Customer
Retail
Corporate
Institution

Card
Branch
Finance
HNW

SMB
National
Multi-Nat
MM
Equities

European Equities
US Equities
Options
SWAPS
Derivatives
With the recent introduction of the behavior qualifier type and Service Domain specific behavior qualifier definitions, the service operations and their information content can be defined to a finer level of detail.

Designing Service Operations Features:
- Allowed types of exchange
- Standard service operation parameter types
- Service operation standard action terms
- Service operations select from a standard information profile
- Checklist Information is referenced in the BIAN Vocabulary
Service Operation Action Terms

The primary purpose for each service operation call is reflected in its action term. BIAN has identified a standard set of action terms to select from and each service operation uses one of these action terms.

<table>
<thead>
<tr>
<th>Action Terms</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initiate</strong></td>
<td>Begin an action including any required initialization tasks</td>
<td>A payment transaction is initiated</td>
</tr>
<tr>
<td><strong>Create</strong></td>
<td>Manufacture and distribute an item</td>
<td>A new analytical model design is created</td>
</tr>
<tr>
<td><strong>Activate</strong></td>
<td>Commerce/ reopen an operational or administrative service</td>
<td>The ATM network operation is activated</td>
</tr>
<tr>
<td><strong>Configure</strong></td>
<td>Change the operating parameters for an ongoing service/capability</td>
<td>The on-line ATM’s in the network are changed to take machine out of service</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>Change the value of some (control record) properties</td>
<td>A customer’s reference details are updated with a change of address</td>
</tr>
<tr>
<td><strong>Register</strong></td>
<td>Recode the details of a newly identified entity</td>
<td>A new customer’s details are captured</td>
</tr>
<tr>
<td><strong>Record</strong></td>
<td>Capture transaction or event details associated with a life cycle step</td>
<td>An employee logs time spent working on a project against the plan</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Execute a task or action on an established facility</td>
<td>A payment is applied to a charge card</td>
</tr>
<tr>
<td><strong>Evaluate</strong></td>
<td>Perform a check, trial or evaluation</td>
<td>The eligibility to sell a product is checked against the customer’s existing agreement</td>
</tr>
<tr>
<td><strong>Provide</strong></td>
<td>Assign or allocate resources or facilities</td>
<td>A branch requests an allocation of cash for its tellers</td>
</tr>
<tr>
<td><strong>Authorize</strong></td>
<td>Allow the execution of a transaction/ activity</td>
<td>Regulatory compliance authorities a product design feature</td>
</tr>
<tr>
<td><strong>Request</strong></td>
<td>Request the provision of some service</td>
<td>A customer requests that a standing order is set up on the current account</td>
</tr>
<tr>
<td><strong>Terminate</strong></td>
<td>Conclude, complete activity</td>
<td>The use of a product version is terminated</td>
</tr>
</tbody>
</table>

Delegation — no new action terms apply as they are called Service Domains offer the same Origination/Invocation & Reporting options described here.

<table>
<thead>
<tr>
<th>Action Terms</th>
<th>Delegation - no new action terms apply as called Service Domains offer the same Origination/Invocation &amp; Reporting options described here</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notify</strong></td>
<td>Provide details against a predefined notification agreement</td>
</tr>
<tr>
<td><strong>Retrieve</strong></td>
<td>Return information/report as requested</td>
</tr>
</tbody>
</table>
## Action Terms mapped to Functional Patterns

<table>
<thead>
<tr>
<th>Action Terms</th>
<th>DIRECT</th>
<th>MANAGE</th>
<th>ADMINISTER</th>
<th>OPERATE</th>
<th>PROCESS</th>
<th>REGISTER</th>
<th>DESIGN</th>
<th>DEVELOP</th>
<th>ASSESS</th>
<th>MAINTAIN</th>
<th>TRACK</th>
<th>ANALYSE</th>
<th>MONITOR</th>
<th>AGREE TERMS</th>
<th>ENROLL</th>
<th>ALLOCATE</th>
<th>FULFILL</th>
<th>TRANSACT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initiate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Create</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Activate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Configure</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Register</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Record</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Evaluate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Provide</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authorise</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Request</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Terminate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Notify</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Retrieve</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default Service Operations mapped to the Functional Patterns
Design Foundation

A Service Domain combines an Asset Type with a Functional Behavior

A Service Domain is accessed through its offered service operations. A selection from a standard set of service operations is matched to the specific functional pattern.
BIAN Service Domain Building Blocks

BIAN service domains are
- Elemental in scope
- Define an Business Function as combination of a type of action or use applied to a type of asset or entity.
- BIAN has
  - identified a standard list of uses (called functional patterns)
  - developed a hierarchical decomposition of the assets or entities (tangible and intangible) that may make up any bank.
- combines a single primary functional pattern (for example: ‘fulfill’) with an asset or entity type (for example ‘Current Account’)

Legend:
- Focus Object
- Core Data
- Service Domain
- Functional Pattern
- Generic Object
- Core Functions & Logic
- Core Data
- Service Operation Organization (Pattern)
- Focus Object State Management
- Services = Core Functions & Logic = Service Operation
BIAN SOA Framework

Service Domain Service Operations

- Service Operation Responsibility Category and Service Type
- Service Operation Name – a formal structure is used
- Service Interaction Type – likely types of operational exchange – these have been discontinued but are included here for reference
- Service Domain - pre & post states
- Input & Output - descriptions of the four main service operation parameter types
First Order Interactions

First Order Interactions Are simple business scenarios that detail the specific service exchanges from the perspective of a ‘primary’ Service Domain in response to a business event.

The First Order Interactions Connected Three Elements:

- Service Domain’s name
- Category of the Business Event
- Descriptive name of the Business Event
A well-formed business scenario has a clear business start and end point and a business goal or purpose that can have some form of associated performance or value measurement. This more complete definition can usually be related to the concept of a business capability as described.
Wireframes

- A collection of Service Domains with their associated service operation connections for some area of activity or domain of the business can be represented using a ‘wireframe’ diagram.
- Business activity can then be overlain on the wireframe as a flow through the network of service operation connections involved for any particular business event/scenario.
Mapping Business Scenario on Wireframes
Clustering Service Domains

- A Service Domain Cluster describes a grouping of related Service Domains. It can also be used to define a grouping that maps to the functional scope of a business application or production system.

- However, for a business application cluster of Service Domains it is necessary to define ‘roles’ that define how the individual Service Domain designs relate to the broader application portfolio.

**Core Banking Business Application Cluster**

[Diagram showing Core/Utility/Proxy Roles]
BIAN's Organization and Content Development Approaches
BIAN's Organization and Content Development Approaches

- The BIAN Organization
- The BIAN SOA Framework
- Content Development
BIAN Organization

- BIAN Organization
  - Architecture Framework & Foundation Team (AF&F)
  - Service Landscape Team (SL)
  - Architecture Committee (AC)
  - Central BIAN Resources
  - The BIAN Board

Executive Director at BIAN
Hans Tesselaar
Content Development is Supported by Tools & Facilities

➢ The design elements of the BIAN SOA Design Framework are supported and enabled by some key standard specifications, guides and tooling. These include:

1. The BIAN Metamodel
2. The BIAN UML Repository
3. The BIAN Business Vocabulary
4. The BIAN Business Object Model (an Extended version of the ISO20022 Business Model)
5. The BIAN Business Scenario Generation and Browsing Tool
6. BIAN How-to Guides and other training materials
7. The BIAN Wiki and Working Group tools and facilities.
Content Development is Supported by Tools & Facilities

Way of Working in line with ISO20022

1. Focus on Service Domain
2. Create BIAN BOM
3. Adapt to ISO20022 BCM
Content Development

- Working Group Assignments
- Building Content in the Working Groups
- Semantic API Initiative
Content Development

Building Content in the Working Groups

Service Domain Specification Cycle

- Define Provisional Service Operations
- Model First Order Interactions
- Service Domains
- Develop Provisional Service Domain Specifications
- Quality assurance design exchange
- Proposed service operation descriptions
- Matched and approved service operations
- Proposed Service Domain Draft

Business Scenario Specification Cycle

- Develop Scenarios & Wireframes
- Refine Scenarios & Wireframes
- Business Scenarios
- Review & Accept Ratified Content
- Notify Service Operation In Context

Copyright BIAN 2011 | Banking Industry Architecture Network
Content Development

Building Content in the Working Groups

1. Identify Candidate Service Domains
2. Develop provisional specifications
3. Model events & first order interactions
4. Draft provisional service operations
5. Assemble elements into scenarios
6. Define/refine scenarios and wireframes
7. Ratify/refine service operation content
8. Quality Assure & Publish

Service Domain Specification Cycle
- Proposed Service Domain Draft
- Matched and approved service operations
- Service Domain Specifications
- Service Operations

Business Scenario Specification Cycle
- Develop Scenarios & wireframes
- Business Scenarios
- Business Service Operation description
- Service operation description
- Service Domains
Approaches for Applying the BIAN Standard in Banking Industry
Applying BIAN Content

Projects Split between Point and Enterprise Solutions

- The ‘point solutions “Using the BIAN model as a high level implementation design”.
- The enterprise solutions Building an Enterprise Blueprint” and “Using the Enterprise Blueprint for Planning & Analysis”.

<table>
<thead>
<tr>
<th>Targeted Point Solutions</th>
<th>Enterprise Analysis Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessing or Implementing a Point Solution – A targeted solution for a narrowly scoped aspect of the business as might be supported by a single application and modeled using a collection of representative business scenarios to identify the involved Service Domains.</td>
<td>Application Portfolio Rationalization – Using the enterprise blueprint as a framework to map the application portfolio to reveal gaps, overlaps and mis-aligned applications. Because the Service Domains define discrete, non-overlapping partitions, mapped applications can be compared ‘like-for-like’.</td>
</tr>
<tr>
<td>Product Launch – An initiative to cover the specific activities that need to be coordinated and procedures followed with the development and deployment of a new product or a significant extension to an existing product. This would include development, training, cutover, customer updates.</td>
<td>Mergers &amp; Acquisitions – Mergar activity is similar to application portfolio rationalization with one additional consideration. Attributes (such as a Service Domain’s cost sensitivity, security or competitive level) can be used to help select between competing applications from the merged organizations</td>
</tr>
<tr>
<td>Core Systems Repurposing – An initiative using the BIAN Service Domain and service operation specifications to renew or repurpose an existing application. The would include specifying and service enabling key service operations to support wider access and possibly aspects of ‘externalization’.</td>
<td>Investment Planning – Using an enterprise blueprint assembled from Service Domains to assess existing capabilities, define target capability requirements, operational characteristics and performance goals and to target investment to address identified shortfalls.</td>
</tr>
</tbody>
</table>
| Vendor Solution Alignment – Match and select vendor solutions for an existing or new business requirement. The motivation differs for banks and vendors:  
  For the Bank – define required functions and interfaces & supplier standards alignment  
  For the Vendor – ease of integration and greater re-use through standard interfaces | Outsourcing/Insourcing – BIAN Service Domains define ‘outsourcable’ business capabilities assuming their service dependencies are fully supported. Usually Service Domains will be outsourced in groups rather than individually. An enterprise model can be used for a cross-organization assessment. |
Applying BIAN Content

Mapping Service Domain to Business Applications

Different types of Mapping Service Domain to Business Applications:

- One-to-One
- One-to-Many
- Many-to-One
Applying BIAN Content

Mapping Service Domain to Business Applications

- The simplest is where the scope of the Service Domain aligns precisely to the coverage of the supporting application.

Application matched ‘One 2 One’

- Business Application
- Service Domain
- Service Mapping

One to One
Applying BIAN Content

Mapping Service Domain to Business Applications

- Less commonly it may be that the Service Domain defines a business function that typically brings together a collection of business applications.
- This can occur when the function exploits an array of tooling and support that may be implemented by different specialized applications.
By far the most common mapping will be where several Service Domains are contained within a business application.
Applying BIAN Content

Service Landscape with Shared and Common Solution Overlaid
Externalization is an approach used to determine what a Service Domain should do itself and when it should call on or ‘delegate to’ the services of another Service Domain. Externalization ensures that each Service Domain performs a single discrete function and so enforces the encapsulation principle.
Applying BIAN Content

Interpreting Service Domains in Different Technical Environments

- Conventional (legacy/core) System Rationalization
- Host Renewal/ESB Integration and Application/System
- Loose Coupled Distributed/Cloud Systems

Two discrete elements of the Service Domain:
- internal processing
- external service orchestration

In this simple example 8 of 9 Service Domain aligned ‘modules’ are externalized from an example stand-alone legacy application.

In the cloud exchanges are at the semantic level, each element interprets the semantic message its own way.
Applying BIAN Content

Conventional (legacy/core) System Rationalization

- Duplication: The most obvious is where two or more business applications perform the role of the same Service Domain.
- Gaps: It may also be possible to see which legacy applications are the best candidates to expand to cover these requirements.
- Misalignment: The problem can be that an application designed to support one function can become compromised when it tries to support many additional and potentially conflicting operational requirements.
Applying BIAN Content

Host Renewal/ESB Integration and Application/System

- This is done using service enabling technologies to provide access to their established host systems such as an enterprise service bus (ESB). These fine-grained services can improve new business application development by providing re-usable software utilities.

- The Service Domains are highly enduring and non-overlapping meaning that it is usually possible to implement ESB service incrementally and have the services adopted progressively across the overall application portfolio.
Applying BIAN Content

Loose Coupled Distributed/Cloud Systems

- This type of environment can be considered as a progression from the ESB environment to something that is a ‘pure’ service oriented architecture.
- The mapped host systems that were presented through the structured ESB are replaced by freestanding business capability ‘containers’ that are made available to collaborate over the network as autonomous service centers.

Advance Cloud Technology Solutions
Applying BIAN Content

Point Solutions

- Describes the typical steps that can be followed to apply the BIAN designs in the context of a targeted or point solution:
  - Business Case
  - Business Scenarios
  - Wireframe
  - Requirement
  - Solution Mapping
  - Customization/Development
  - Deployment Planning
Applying BIAN Content

Point Solutions - Map and Assess Existing Systems/Candidate Packages

- **Functional Coverage**: A simple assessment is performed to determine whether the requirement is either fully supported, can be supported with limited enhancement work or if the requirement is not supported at all.

<table>
<thead>
<tr>
<th>Correspondence Feature Types</th>
<th>Feature Description</th>
<th>System 1</th>
<th>System 2</th>
<th>System 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Technical Architecture</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Operating Features</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Non-Functional</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mapping Candidate Systems to the Feature List of a Service Domain
Applying BIAN Content

Point Solutions- Map and Assess Existing Systems/Candidate Packages

- **Hygiene Factors**: Hygiene Factor Analysis can be useful to quickly eliminate candidate solutions and simplify the selection process.
- There are many possible factors involved in determining the suitability of a candidate solution (functional, technical and operational) that influence its long term use or strategic alignment.

Example Hygiene Factor Analysis
Applying BIAN Content

Enterprise Blueprint – Assembling a Representative Enterprise Blueprint

**Step 1 – Filter**
Select the required Service Domains (note the ‘value chain’ structure shown is explained elsewhere in the How To Guide)

**Step 2 – Specialize**
Specialise/adapt Service Domains to the enterprise (refine, repeat, combine)

**Step 3 – Organize**
Repeat structures to match the lines of business and the reporting hierarchy of the enterprise

Three steps in developing an Enterprise Blueprint
Applying BIAN Content

Matching the Enterprise Segmentation Approach

Mapping product and customer types to segmentation views
Applying BIAN Content

An Enterprise Blueprint is a Framework for Analysis/Measurement

Service Domain level systems mapping

Shapes show application functional footprint

Target & Current State Capabilities (e.g.):
- Core functionality and service boundary
- System/technology properties
- Operational properties
- Organizational properties
- Business/Financial properties

Many different target and current state properties can be defined in the associated Service Domain specifications.

Performance Measures

Resource Mapping/Assessment
- systems coverage
- organizational coverage

Performance Measures
- actual to plan (budget)
- productivity
- utilization

Feature Attribution
- input source target
- business criticality
- customer influence
Questions?