Organization

Authors

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Paul Bonnie</td>
<td>ING</td>
</tr>
<tr>
<td>Author</td>
<td>Gerard Peters</td>
<td>Capgemini</td>
</tr>
<tr>
<td>Author</td>
<td>Pierre-Philippe Delmarcelle</td>
<td>Callataï en Wouters</td>
</tr>
<tr>
<td>Author</td>
<td>Thomas Obitz</td>
<td>KPMG LLP (UK)</td>
</tr>
</tbody>
</table>

Reference Documents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Type of Document</th>
<th>Reference to Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOGAF 9</td>
<td></td>
<td><a href="http://www.opengroup.org/togaf">www.opengroup.org/togaf</a></td>
</tr>
<tr>
<td>BIAN Service</td>
<td></td>
<td><a href="http://www.bian.org/content/bian_results/index_en.html">www.bian.org/content/bian_results/index_en.html</a></td>
</tr>
<tr>
<td>Landscape 1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAN Metamodel</td>
<td></td>
<td><a href="http://www.bian.org/content/bian_results/index_en.html">www.bian.org/content/bian_results/index_en.html</a></td>
</tr>
</tbody>
</table>

Status

<table>
<thead>
<tr>
<th>Status</th>
<th>Date</th>
<th>Actor</th>
<th>Comment/Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitted</td>
<td>7 November 2011</td>
<td>The Open Group</td>
<td>Submitted for review by The Open Group Architecture Forum and BIAN</td>
</tr>
<tr>
<td>Review Closure</td>
<td>8 December 2011</td>
<td>Working Group</td>
<td>Telecom for Closure of Review Comments; changes agreed</td>
</tr>
<tr>
<td>Approved</td>
<td></td>
<td></td>
<td>Paper approved by Architecture Forum</td>
</tr>
<tr>
<td>Approved</td>
<td></td>
<td></td>
<td>Paper approved by BIAN</td>
</tr>
</tbody>
</table>

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date of Change</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>7 December 2011</td>
<td>Paul Bonnie, Gerard Peters, Pierre-Philippe Delmarcelle, Thomas Obitz</td>
</tr>
</tbody>
</table>
Copyright

© 2012 The Open Group and BIAN eV. All Rights Reserved.

THIS DOCUMENT IS PROVIDED "AS IS," AND BIAN AND THE OPEN GROUP AND THEIR MEMBERS MAKE NO
REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE
CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH
CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS, OR OTHER RIGHTS.

NEITHER BIAN AND THE OPEN GROUP NOR THEIR MEMBERS WILL BE LIABLE FOR ANY DIRECT, INDIRECT,
SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR
DISTRIBUTION OF THIS DOCUMENT UNLESS SUCH DAMAGES ARE CAUSED BY WILFUL MISCONDUCT OR GROSS
NEGLIGENCE.

THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT
REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS TO BIAN AND THE OPEN GROUP AND OTHER
MEMBERS IN CERTAIN WRITTEN POLICIES OF BIAN AND THE OPEN GROUP.

Boundaryless Information Flow™ is a trademark and ArchiMate®, Jericho Forum®, Motif®, Making Standards Work®, OSF/1®,
The Open Group®, TOGAF®, UNIX®, and the “X” device are registered trademarks of The Open Group in the United States
and other countries. All other brands, company, and product names are used for identification purposes only and may be
trademarks that are the sole property of their respective owners.
# Table of Contents

1 Introduction...............................................................................................................................7
  1.1 Architecting for an Emergent Industry...............................................................................7
  1.2 TOGAF and BIAN Initiatives.............................................................................................7
  1.3 White Paper Objectives ....................................................................................................8
  1.4 White Paper Structure ......................................................................................................8
  1.5 Future Evolution of this White Paper ...............................................................................8

2 Drivers in the Banking Industry................................................................................................9
  2.1 External Forces and Market Drivers ..................................................................................9
  2.2 Responses of Banks .........................................................................................................9
  2.3 Impact on the IT Landscape of Banks .............................................................................10
  2.4 Implied Change in IT ......................................................................................................10

3 Enterprise Architecture.........................................................................................................12
  3.1 Development of the Architecture Discipline ....................................................................12
  3.2 Enterprise Architecture in Banking ................................................................................12
  3.3 Challenges for Enterprise Architects in a Banking Environment ....................................12

4 BIAN – A Business Service Model for the Banking Industry..............................................14
  4.1 BIAN Vision and Strategy ...............................................................................................14
  4.2 BIAN Deliverables .........................................................................................................14
  4.3 BIAN Service Landscape ................................................................................................15
  4.4 Overview of the BIAN Metamodel ..................................................................................17
    4.4.1 Metamodel Viewpoint: The Service Landscape ......................................................17
    4.4.2 Metamodel Viewpoint: Capabilities, Responsibilities, and Delegation ......17
    4.4.3 Metamodel Viewpoint: Business Objects ...............................................................17
    4.4.4 Metamodel Viewpoint: Message .........................................................................17
  4.5 BIAN Service Domain Definitions ..................................................................................17

5 TOGAF – The Leading Framework for Enterprise Architecture......................................19
  5.1 The Open Group ............................................................................................................19
  5.2 TOGAF as an Industry-Independent Framework for Enterprise Architecture ............19
  5.3 Overview of TOGAF Components ..................................................................................19
    5.3.1 TOGAF Architecture Development Method (Part II) .............................................21
    5.3.2 Architecture Content Framework (Part IV) ..............................................................21
    5.3.3 TOGAF Enterprise Continuum (Part V) .................................................................22

6 Leveraging the BIAN Deliverables with TOGAF.................................................................23
  6.1 Introduction ......................................................................................................................23
  6.2 Relating BIAN to the Phases of the ADM (Part II) ..........................................................23
    6.2.1 Preliminary Phase (Chapter 6) ...............................................................................23
    6.2.2 Architecture Vision (Phase A, Chapter 7) ...............................................................24
    6.2.3 Business Architecture (Phase B, Chapter 8) ............................................................24
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.4</td>
<td>Information Systems Architecture (Phase C, Chapter 9)</td>
<td>24</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Technology Architecture (Phase D, Chapter 12)</td>
<td>25</td>
</tr>
<tr>
<td>6.2.6</td>
<td>Opportunities and Solutions (Phase E, Chapter 13)</td>
<td>25</td>
</tr>
<tr>
<td>6.2.7</td>
<td>Migration Planning (Phase F, Chapter 14)</td>
<td>25</td>
</tr>
<tr>
<td>6.2.8</td>
<td>Implementation Governance (Phase G, Chapter 15)</td>
<td>25</td>
</tr>
<tr>
<td>6.2.9</td>
<td>Architecture Change Management (Phase H, Chapter 16)</td>
<td>25</td>
</tr>
<tr>
<td>6.2.10</td>
<td>Requirements Management (Chapter 17)</td>
<td>26</td>
</tr>
<tr>
<td>6.3</td>
<td>Relating BIAN to TOGAF Guidelines and Techniques (Part III)</td>
<td>26</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Applying the ADM at Different Enterprise Levels (Chapter 20)</td>
<td>26</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Using TOGAF to Define and Govern SOAs (Chapter 22)</td>
<td>27</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Architecture Principles (Chapter 23)</td>
<td>27</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Architecture Patterns (Chapter 25)</td>
<td>27</td>
</tr>
<tr>
<td>6.3.5</td>
<td>Interoperability Requirements (Chapter 29)</td>
<td>27</td>
</tr>
<tr>
<td>6.4</td>
<td>BIAN and the TOGAF Architecture Content Framework (Part IV)</td>
<td>28</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Deliverables, Artifacts, and Building Blocks</td>
<td>28</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Mapping the BIAN Deliverables to the TOGAF Content Metamodel</td>
<td>28</td>
</tr>
<tr>
<td>6.5</td>
<td>BIAN and the TOGAF Enterprise Continuum (Part V)</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>Other Relevant Standards for the Banking Industry</td>
<td>33</td>
</tr>
<tr>
<td>7.1</td>
<td>Alignment of BIAN with Other Technical, Semantic, and Messaging Standards</td>
<td>33</td>
</tr>
<tr>
<td>7.2</td>
<td>APQC as a Process Standard</td>
<td>33</td>
</tr>
<tr>
<td>8</td>
<td>Glossary</td>
<td>35</td>
</tr>
<tr>
<td>9</td>
<td>APPENDIX: The BIAN Metamodel</td>
<td>37</td>
</tr>
<tr>
<td>9.1</td>
<td>Service Landscape Viewpoint</td>
<td>37</td>
</tr>
<tr>
<td>9.2</td>
<td>Capabilities-Responsibilities-Delegation Viewpoint</td>
<td>39</td>
</tr>
<tr>
<td>9.3</td>
<td>Business Object Viewpoint</td>
<td>40</td>
</tr>
<tr>
<td>9.4</td>
<td>Message Viewpoint</td>
<td>41</td>
</tr>
</tbody>
</table>
Management Summary

Financial institutions are facing significant changes of the environment in which they operate. For example, the financial crisis, changing customer behavior, increased risk awareness, focus on cost reduction, and the entrance of new financial players in the market-place have structurally changed the financial industry.

Numerous initiatives were taken to adapt the financial industry to this changing and highly competitive environment, resulting in more efficient and effective organizations. Not only the operations, but also the supporting IT required a challenging transformational change. Given the close inter-dependency of operations and IT in financial institutions, a carefully planned and guided integrated approach to change is required. Enterprise architecture is key to enabling such change initiatives.

This White Paper aims at supporting Enterprise Architects within the banking industry, reaping the synergies of two complementary industry frameworks:

- TOGAF, an Open Group Standard, is a proven enterprise architecture methodology and framework used by the world’s leading organizations to improve business efficiency.
- BIAN, the Banking Industry Architecture Network, delivers an overall framework and set of IT Service definitions specific to the banking industry.

There are clear synergies in using BIAN in combination with TOGAF. When applying TOGAF in a banking environment, the BIAN content will contribute to speed up the work by providing banking industry-specific architecture content. On the other hand, TOGAF facilitates architecture development work by providing a structured approach and a complete structure of relevant artifacts. Hence, TOGAF adds value to the BIAN deliverables especially for the project approach and capability to perform.

In the White Paper, the concepts and core elements of both frameworks are described in order to reach a common understanding.

With respect to the TOGAF Enterprise Continuum, the BIAN content is positioned as an industry-specific architecture within the Architecture Continuum, addressing banking application-to-application integration issues.

With respect to the TOGAF Content Metamodel, the main BIAN content (Business Objects and the Service Landscape) is positioned in the Information System Architecture.

In the heart of the White Paper, both TOGAF and BIAN are mapped to each other. The leverage of the BIAN Service Landscape in the context of the TOGAF Architecture Development Method (ADM) is further elaborated. For each step in an architecture development process, the integration of BIAN deliverables, especially the BIAN Service Landscape, is described.

In applying both industry frameworks in an integrated way, Enterprise Architects will be able to speed up their work and improve quality and consistency of the architecture products they deliver.
1 Introduction

1.1 Architecting for an Emergent Industry

Financial institutions are facing significant changes of the environment in which they operate. Although the financial crisis starting in 2007 has been a catalyst for many of these, most of them – like industrialization of operations, changing customer behavior, and increasing risk awareness – have emerged since the late 1990s.

Industry maturity is an essential driver of banking transformation. Significant parts of both the services and the product portfolio are getting commoditized, resulting in margin pressures, but on the other side presented as cost-saving opportunities. Changing customer behavior and non-traditional providers of banking services have a deep impact, particularly in the retail banking industry. Emerging economies offer opportunities for growth. Banks have to manage risks carefully, and report transparently and consistently on their activities.

Banks react by rationalizing their product and services portfolios. Automation reduces the cost of manual labor. Organizations distribute processes around the globe, or outsource them; specialized service providers emerge to take them up. White-labeled products can be sold in the supermarket. Social networks offer new opportunities to engage an empowered consumer.

These developments result in increasing tension between the needs of financial institutions and the capabilities of their IT landscapes. Bespoke systems hardwire the operating model of the past. Monolithic platforms make it difficult to facilitate carving out organizational units and processes. Core systems with limited flexibility have been patched up by layers and layers of supporting systems. Functionality duplicated by business silos and channel stovepipes duplicate maintenance effort, and hampers information consistency.

Remediation requires transformational change – which promises huge dividends. A recent study suggests that banks replacing their core banking systems experience an acceleration of their pre-tax profit growth rate by 30%.

However, the transformation is challenging. It needs careful planning. Given the close inter-dependency of operations and IT in financial institutions, an integrated approach to change is required.

Enterprise architecture is crucial to enabling such initiatives. Today, we find an overwhelming interest in this technique and there is virtually no large corporation which does not have an enterprise architecture team. Given the need for integration among market participants, banks have always been at the forefront of standardization and architectural maturity.

1.2 TOGAF and BIAN Initiatives

Two initiatives have gained significant interest in the financial industry – TOGAF and BIAN:

- TOGAF, an Open Group Standard, is a proven enterprise architecture methodology and framework used by the world’s leading organizations to improve business efficiency. It is the most prominent and reliable enterprise architecture standard, ensuring consistent standards, methods, and communication among enterprise architecture professionals. Enterprise architecture professionals fluent in TOGAF standards enjoy greater industry credibility, job effectiveness, and career opportunities.

- BIAN, the Banking Industry Architecture Network, is also gaining significant traction in the banking industry. Its success mirrors the increasing importance of standard software in a maturing industry – the BIAN Service Landscape is a blueprint for the logical components of a bank’s IT environment. Leveraging this blueprint can significantly accelerate architecture initiatives – be it in the planning of change initiatives, in the procurement of components, or the benchmarking of an existing landscape against best practices.

---

1.3 White Paper Objectives

Both TOGAF and BIAN are frameworks which are not only powerful, but also exhaustive in their domains. This White Paper aims at supporting architects within the banking industry, reaping the synergies of both frameworks. It provides an overview of both frameworks and an overall mapping of their content in order to help architects accelerate their work.

1.4 White Paper Structure

This White Paper is organized into the following sections:

- Section 2 analyzes the current situation in the banking industry and its IT environment.
- Chapter 3 describes the specifics of enterprise architecture in this industry.
- Chapter 4 provides an introduction to BIAN and Chapter 5 provides an introduction to TOGAF. Readers familiar with either framework may skip the respective material.
- Chapter 6 is the heart of this White Paper: it maps both frameworks, explains how to leverage the BIAN Service Landscape in the context of the TOGAF Architecture Development Method (ADM), how it integrates with the TOGAF Content Framework, and how it helps to build an architecture capability.
- Chapter 7 describes other frameworks, which an enterprise architect in a bank might find useful.
- Chapter 8 is the glossary, explaining the most important terminology used in this White Paper.
- Chapter 9 details the most important elements of the BIAN Metamodel.

1.5 Future Evolution of this White Paper

This White Paper is the product of a joint effort of a working group of BIAN and The Open Group Architecture Forum, which brought together the experience of banks, banking software vendors, and consulting organizations.

The group endeavors to continue updating and enhancing the document as appropriate. On the BIAN side, it currently relates to the Service Landscape Version 1.5; enhancements might be appropriate with the evolution to Version 2.0, and when embracing a broader SOA approach encompassing not only IT services, but also those provided by the business itself. On the TOGAF side, it is based on Version 9.

The authors hope that their effort will help you – the reader – to succeed in today’s complex banking industry environment.
2 Drivers in the Banking Industry

The banking industry is currently undergoing significant transformation. It is driven by changes in the marketplace, customer behavior, and regulatory environment; at the same time, competitors are raising the bar. Although some of these changes were prompted by the credit crisis, most of them have been ongoing for several years.

2.1 External Forces and Market Drivers

Since 2007, the banking industry not only has had to cope with a major crisis of the financial system, but also with changes in the marketplace which have influenced the industry for several years.

During the recent crisis, a significant part of the banking sector – including several systemically important financial institutions – have come to face not only a shortage of liquidity, but even of capital to cover the losses induced by sudden revaluation or illiquidity of considerable parts of their asset portfolio. They became dependent on government bailout for sheer survival; significant funding had to be injected by the public sector.

As a result, public scrutiny tightened on financial services. Regulatory regime and accounting standards (including SOX, Basel I – III, Dodd-Frank, US-FATCA, and IFRS) will continuously change the environment in which banks are operating.

In response, financial institutions shifted focus towards risk management, compliance, and transparency. Their appetite and ability to take risk lowered considerably. Limitations on leverage ratios and higher requirements of regulatory capital constrained their ability to take up new business.

As per the trend of previous years, customer behavior continued to change, and customer loyalty decreased. More and more banking services and products – and most of the retail products, like payments, savings, and loans – were increasingly commoditized.

Finally, new players (like Google, PayPal, Zopa, etc.) entered the financial market, claiming a (niche) part of the market without the burden of a legacy full-service product offering.

2.2 Responses of Banks

Recovering their losses and regaining profitability in this environment imposed a challenge to many organizations. Some players realized that they would not survive as independent entities; others sold parts of their operations to provide capital to their core business; a few had to close down, typically selling off profitable parts. A consolidation and rationalization process started in the banking industry. This resulted in a wave of mergers, acquisitions, and spin-offs, increasing the demand for integration amongst organizations.

Today, the surviving players have to control costs tightly and still find new, profitable areas of business. At the same time, they have to accommodate the need to manage risk and ensure adequate reporting.

Many cost levers are adapted from the production industry. In a quest for cost reduction and operational excellence, internal processes are streamlined and optimized by techniques such as Lean and Six Sigma; activities are automated and manual interventions are minimized; Straight Through Processing (STP) increases throughput and reduces errors; product factories replace the traditional branch back-office. As a result, volumes can be scaled at sub-linear cost.

Large organizations are distributing processes across multiple operational centers around the globe. Specialized service providers emerge in areas such as payments, securities processing, and loans, driven by the quest for scale and operational excellence.

Product portfolios are harmonized and optimized. This process typically results not only in fewer and simpler products and services, but also in a modular product design allowing parameterization of existing products, and to combine new products from existing components. White-labeling allows institutions on the one hand to procure parts of their product portfolio without managing additional processing complexity; on the other hand, it enables them to widen the market reach for their own products, even using non-traditional channels like supermarkets and online merchants.
Other ways of expanding the market of an organization are by improving access to its service offerings across channels. Banks are adopting a flexible and more personalized “on-demand” approach, with products available anywhere (via the “channel of choice”) and anytime at the lowest cost. Today, customers expect their bank to support new online real-time channels through mobile devices (smart-phones, tablets) in a way which is integrated and consistent with the traditional branch and call-center channels.

### 2.3 Impact on the IT Landscape of Banks

Almost all assets managed by a financial institution are completely virtualized – represented as booking entries or, more generally, as information. Hence, information processing and its enabling technology are of tremendous relevance to the banking industry. Its IT spend in 2011 is estimated to reach 8.7% of revenues – almost three times the cross-industry average.\(^2\)

Today, banks are running an IT platform of enormous complexity. It is impacting their ability to react quickly to a changing market, affecting their cost/income ratio, and driving the risk associated with any further change. Action is required.

Most banks have a long history in IT. Starting from the replacement of physical index files in manual administrative processes, to creating large and complex legacy IT systems to automate processes, supporting numerous market segments, customer groups, products, and services.

Many core systems were developed in-house several decades ago; in particular, the current account and lending platforms of commercial and retail banks often were developed in the 1960s and 1970s. Over the years, functionality for new products is added; the migration from the batch to a near-time processing paradigm required deep re-engineering. Support for new channels is added around the core platforms; so were systems for risk management, compliance, increasing security requirements, and regulatory reporting. Parallel development in different lines of business resulted in functional duplication. Mergers and acquisitions, as well as in-sourcing and outsourcing (also to the “Cloud”), added further complexity to the IT landscape. The need for integration resulted in a hairball of system dependencies which made change tedious at best.

While banks have proven very successful in standardizing external communication, internal integration often has been achieved in a proprietary way. Many firms have implemented a common integration platform – using EAI (Enterprise Application Integration) or ESB (Enterprise Service Bus) approaches – but use completely custom data formats with it. Not only do these proprietary formats create maintenance challenges and vendor dependencies, they result in off-the-shelf solutions, which typically support a range of integration standards that cannot easily be plugged into a standards-based landscape. They require complex mappings, which keep integration cost high on an ongoing basis.

As landscapes were built over decades, the development of IT during this time-span has manifested itself in a technology stack which grew in a way not dissimilar to sediment depositions. Hardware platforms, programming languages, transaction monitors, databases, application servers, standard software platforms, and integration technologies of about 50 years have manifested themselves, forcing banks to keep a skill set of enormous diversity and raising concerns about operational risk related to technical obsolescence.

### 2.4 Implied Change in IT

As a consequence of the above, many banks find that their IT landscape is complex, inflexible, and silo-based, impacting their ability to thrive in a dynamic and competitive marketplace:

- Running IT is becoming more costly – the Total Cost of Ownership (TCO) is perceived to be very high. “Keeping the lights on” requires an increasing part of IT budgets. Funding for new initiatives, supporting the introduction of new products, or transforming the bank’s operations becomes challenging, impacting competitiveness. Key resources are locked up in maintenance activities.

Due to the complexity of the IT landscape, the time-to-market for introducing new or amending existing products and services is too long; often, the cost of change is too high to achieve a positive business case of initiatives. Hence, IT complexity stifles business agility.

Additionally, integrating new products in the current IT landscape is not without risk and may add further complexity.

All this puts significant pressure on the IT function. Numerous measures are taken to mitigate the pain. IT rationalization, active life-cycle management, the adoption of market and industry standards to reduce integration costs, the utilization of Service Oriented Architecture (SOA), and the selection and implementation of readily available software packages and solutions (like COTS and SaaS) are just a few to mention.

In this context, more and more institutions are approaching the replacement of custom-built core platforms. In the past, both their deep integration with the existing landscape, and the business rules hidden in historic COBOL code, made banks wary about changing them. In the meantime, both an improved capability of managing such large-scale initiatives, typically in an incremental way rather than a “big bang”, and the state of technology has helped kick off a wave of core banking replacement initiatives. Despite the perceived risk, the rewards are high: A recent study\(^3\) suggests that banks replacing their core banking systems experience an acceleration of their pre-tax profit growth rate by 30%.

In many cases, however, changes are perceived as IT-driven. A lack of traceability against the overall objectives of the organization results in limited support and commitment of IT changes by other non-IT business functions. A holistic approach to transformation is required.

---

3 Enterprise Architecture

In this situation, enterprise architecture becomes a central tool for enabling the transformation process of both IT and the overall business.

3.1 Development of the Architecture Discipline

Enterprise architecture still is a young discipline. It has evolved over the last 20 years in terms of both its scope and its approaches. Established in the 1980s and 1990s as an approach for standardizing IT landscapes, it has become a crucial tool for aligning an IT landscape to the needs of the overall organization.

Today, many organizations find that architectural approaches are applicable not only to the transformation of IT. As parts of processes are sourced externally and distributed across organizations, and as parts of the value chain are delegated to partners, architecture becomes an indispensable planning tool.

To distinguish these stages in the evolution of enterprise architecture, specific terms are coined. Enterprise Technology Architecture (ETA) is used for an approach primarily focused on technology; Enterprise IT Architecture (EITA) aims at providing the right IT landscape for an organization. For the integrated transformation of an organization using architectural approaches, the term Enterprise Architecture (EA) is commonly used. Still, given the swift development of the discipline, there is significant ambiguity in terminology.

This White Paper does not judge what “real” enterprise architecture is; we will use the term both for approaches targeted primarily towards IT transformation, as well as those aiming at an integrated organizational development.

3.2 Enterprise Architecture in Banking

As discussed before, banks are amidst transformational change in the market-place, in their way of doing business, and in their IT landscapes. Given the close inter-dependency of people, processes, and supporting technology, most banks are well aware of the need to manage their architecture in a coherent way, in alignment with their strategy. Enterprise architecture is a tool for achieving this objective.

Many financial institutions have set up enterprise architecture functions early, and they have evolved into a crucial tool for driving projects in alignment of business and IT. They are also in the process of manifesting the close linkage of business processes and their supporting IT assets in the organizational set-up. Merrill Lynch is a well-known, but in no way a unique example of a bank consolidating operations and IT in a common unit.4

Hence, enterprise architecture in the banking industry appears to be in a stronger position than in many other industries. It appears to be well-positioned to influence the organization for converging to a consistent and strategy-aligned Target Architecture. However, there are also some challenges specific to this industry.

3.3 Challenges for Enterprise Architects in a Banking Environment

Enterprise architects in the banking industry – like in any other industry – face a number of challenges.

Three observable challenges in many organizations are mentioned here. A typical challenge is a lack of an overall, consolidated, and accepted perception of the current and target state. A second common challenge relates to the actual impact of architecture on decision-making at various levels of the organization. A third one relates to the early start of enterprise architecture, which resulted in a significant set-up of “custom built” assets within the architecture function, and limited adoption of standards in some areas.

In most institutions, there is no overall (documented) description of business processes and their supporting IT assets. This might indicate that there is still a gap in the Business/IT-alignment. In general, banks have not

---

4 Refer to: www.mckinseyquarterly.com/Focusing_on_the_customer_An_interview_with_the_head_of_Merrill_Lynchs_operations_and_IT_2021.
documented their overall Business Architecture, with some parts of the landscape missing or under-developed. Consequently, many banks do not have a full and complete insight into their services, demonstrated by a complete, up-to-date, and widely known service catalog. All of this is reflected in a fragmented IT landscape. Often, the question is raised whether this landscape is suitable to support business dynamics in a predictable way.

The influence of enterprise architecture concerning decision-making both on a strategic level and in actual projects is questionable at times. This is actually not specific to the banking industry as such; the governance of enterprise architecture is a common challenge. Communication tends to be a key focus area.

As mentioned above, existing custom solutions and integration approaches drive implementation and maintenance cost; still, making the case for replacing these established assets is daunting. Internal to the enterprise architecture function, non-standardized architecture processes make it difficult to work with representatives of multiple vendors in the architecture team, and hinder flexible augmentation of resources.

Both TOGAF and BIAN may have a positive impact on these challenges: In a banking environment, the Service Landscape and the business scenarios of BIAN (as explained in Section 4.2) can serve as a reference to benchmark architectural documentation, and for identifying gaps. In addition, the interfaces of the BIAN services can be a blueprint for integrating components.

In terms of architectural work, TOGAF establishes a shared approach and vocabulary. The definition of the architecture capability in TOGAF helps to identify processes relevant for setting up appropriate architecture governance.
4 BIAN – A Business Service Model for the Banking Industry

4.1 BIAN Vision and Strategy

As mentioned above, significant transformative forces have influenced the banking industry over the years. It became more and more apparent that significant renewal of existing platforms was required, and that standard software would play a significant role in this process. The service-oriented paradigm appeared to be an approach for orchestrating business capabilities based on interoperable components throughout the organization. However, to plan this transformation, to ensure interoperability, and to contain integration cost, a common semantic framework would be highly advantageous. BIAN aims to address this need.

The Banking Industry Architecture Network (BIAN) founded in 2008 is a not-for-profit standards organization. Members of BIAN are relevant players in the banking industry consisting of banks, service providers, and software vendors. BIAN is a global, open, and independent community in which the members openly exchange banking IT requirements concerning Service-Oriented Architecture (SOA).

BIAN’s vision of an industry-wide consensus regarding SOA for banking systems is closely aligned to the business objectives of agility and reduced cost. This vision is based on the concept of achieving the flexibility required by implementing:

- Standards-based semantic interoperability of in-house built and commercially productized banking IT services
- Banking IT services which are independent of the technology platform

BIAN has the strategy to bring banks, service providers, and software vendors together in a community to achieve synergies. This synergy is achieved by:

- Collaborating on a consensus understanding of the requirements for “banking enterprise IT services” within an SOA framework
- Formally describing the services required to orchestrate a banking IT landscape
- Close alignment with other standardization bodies like OMG, ISO, and IFX to prevent “development of similar wheels” (see also Chapter 7)
- Encouraging the adoption of existing or new industry semantic standards, including descriptions of service boundaries and responsibilities, service operations, and the messages that serve as input and output to the service operations

More information on this can be found on the BIAN website: www.BIAN.org.

4.2 BIAN Deliverables

To address its strategy, BIAN has produced the following main deliverables which are further described in the following sections:

- Service Landscape: The set of BIAN service definitions, organized within a functional hierarchy (see Section 4.3) and set up via an overall, systematic approach to IT service identification and definition.
- BIAN Metamodel: Definitions of the concepts used to define BIAN services.
- Service Domain Definitions: Semantic definitions of banking-specific IT services.

5 Note that this vision is currently under analysis to enlarge the current SOA approach (focusing on IT services) to a broader SOA approach encompassing business service boundaries (Service Landscape better covering all banking activities, use of business scenarios, …).
A number of further deliverables are currently under construction. Some of these are business scenarios, which describe collaborations among the BIAN services to achieve a specific business objective, such as executing a payment. As of now, these scenarios are not yet publicly available. In due time they will be published, rather as examples than as canonical standards against which BIAN compliance will be measured.

4.3 BIAN Service Landscape

The ultimate goal of BIAN is to arrive at a set of standardized IT services for banking. However, these IT services need to be positioned in a framework which is common to the banking industry.

BIAN is heading towards reaching consensus on a common logical functional landscape, consisting of coherent sets of logical capabilities/functionabilities. It is called the Service Landscape (and not “IT function landscape”) to emphasize its role in describing the set-up of the BIAN Service Operations.

The approach followed by BIAN is a combination of top-down and bottom-up. It draws heavily upon experience from the BIAN members for identifying candidate Service Operations bottom-up, and uses a top-down approach mainly for checking consistency and completeness, and for detecting missing services. However, there is no check on overlapping functionality derived from detailing the different services.

The Service Landscape is a powerful instrument for many reasons:

- It acts as an “index” to the service catalog, offering an overview and the basis for facilities for discovery and look-up.
- It organizes the services, locating them within a functional hierarchy of Business Areas and Business Domains.
- It is the basic instrument for service portfolio management; i.e., managing the complete set of services as a whole, allowing reporting, monitoring, impact analysis, trend surveys, usage statistics, completeness and consistency checks, etc.
- It serves as a reference framework for migration roadmaps (by projecting the existing and target application landscape on the Service Landscape) or for gap analysis.

The BIAN Service Landscape Version 1.5 depicted below was issued in July 2011. This version is an interim release containing a high degree of exploratory design and is published along with full disclosure of the underlying design techniques and examples of the anticipated uses and benefits.
## BIAN Services Landscape v1.5

The BIAN Services Landscape v1.5 provides a high-level view of the services and capabilities within the financial services industry. It categorizes services into various domains such as business support, operations & execution, analytics & risk, and sales & service. Each domain is further divided into specific service areas, reflecting the operational and functional aspects of financial services organizations.

### Business Support
- Human Resource Management
- IP Management
- Knowledge of IP Management
- Corporate Knowledge
- Business Decisions
- Business Solution
- Role & Authorization

### Operations & Execution
- Reference Data Business Support
- Sales & Service
- Cross Product
- Corporate Products
- Corporate Products
- IT Management
- Systems Administration
- Investment Management
- Business Administration
- Investment Portfolio Planning

### Analytics & Risk
- Channel Specific
- Cross Channel
- Banking Products
- Corporate Products
- Payment
- Fraud Detection
- Corporate Credit Facility
- Business Domain
- Key:

### Sales & Service
- Consumer Loans & Deposits
- Credit
- Consumer Services
- Market Operations
- Trade Finance Services
- Cards
- Operational Services
- Accounts Receivable
- Consumption Finance Services

### Reference Data
- Trade Finance
- Business Data
- External Agency
- Model Management
- Service Data Management
- Offer Management
- Case Management

### Key
- Business Area
- Business Domain
- Service Domain
- Reference Point
- End Point

*Charters are Special Mixed Service Domains (UMS) uniquely established for business processes or when the business case encompasses a specific business function.*

The BIAN Services Landscape v1.5 is a high-level view that helps identify and understand the various services and capabilities required to support the operational and functional aspects of financial services organizations.
4.4 Overview of the BIAN Metamodel

The BIAN Metamodel describes the structure of the BIAN deliverables.

In the following paragraphs, the most relevant views of the BIAN Metamodel are briefly described. A detailed explanation is provided in an appendix (see Chapter 9).

4.4.1 Metamodel Viewpoint: The Service Landscape

The key constituents of the BIAN Service Landscape are:

- The **Business Area**, representing a coarse-grained top-level functional structure of a financial institution; e.g., “operations & execution”, “reference data”, “sales & service”.

- The **Business Domain**, which is the next level of decomposition, representing certain skills and knowledge, which are clearly identifiable in the banking business (e.g., “payments” as a domain within “operations & execution”).

- The **Service Domain** as the smallest unit of functionality or business capability which can be service-enabled independently (e.g., “payments execution” is a Service Domain within the “payments” Business Domain); it owns a specific business object, its focus object (see below).

The Service Domains are the canonical elements of the Service Landscape; their grouping into Business Domains and Business Areas is not. It is not possible to provide a common hierarchy for structuring the business capabilities which would be applicable to all banks. Hence, different organizations may structure the Service Domains into different Business Area/Business Domain hierarchies.

- A **Service Group** is an interface within a Service Domain described in business rather than technical terms (e.g., “manage payment order”).

- A **Service Operation** is an individual interaction with the Service Domain in terms of business semantics (e.g., “execute payment order” or “update payment order” are service operations of the service group “manage payment order”).

4.4.2 Metamodel Viewpoint: Capabilities, Responsibilities, and Delegation

A BIAN Service Domain is responsible for supporting specific aspects of the business. This responsibility is supported either by a capability of the Service Domain itself, or by delegation to another Service Domain.

4.4.3 Metamodel Viewpoint: Business Objects

A **Business Object** is an abstraction of the objects managed within the Service Landscape. The object owned by a Service Domain is called its **Focus Object**.

4.4.4 Metamodel Viewpoint: Message

**Messages** play a fundamental role in the definition of ServiceOperations, as a ServiceOperation may have an input message, may have an output message, and must have a fault message.

4.5 BIAN Service Domain Definitions

The Service Domains define the industry standard semantic services that BIAN seeks to establish for the Banking Industry. Each Service Domain – including its Service Groups and their respective Service Operations – is assigned to a specific Business Domain within a Business Area of the Service Landscape.

Service Domains are described at multiple levels:

- The BIAN Business Scenario: A BIAN Business Scenario is a simple visual representation of how a collection of Service Domains might react to or collaborate for a specific business event or opportunity. Its purpose is to clarify the use of the Service Domain by example.
- BIAN Level 1 Service Domain Definition: Defines the functional content and boundary of a BIAN Service Domain. Includes the Service Domain’s Responsibilities, Capabilities, Delegations, and Focus Objects.

- BIAN Level 2 Service Domain Definition: Defines the Service Domain’s Service Groups and their Service Operations. Also defines the Messages that Service Operations receive as input and return as output, but the Messages are defined at the semantic rather than the technical level.

- BIAN Level 3 Definition: Defines a Service Domain at the technical level. BIAN Level 3 definitions are developed for proof-of-concept only and are not published as canonical standards against which BIAN compliance is measured. However, a Level 2 Service Domain definition can be mapped to standardized technical messages such as IFX and ISO 20022\(^6\) messages.

\(^6\) ISO 20022: Financial Services -- Universal Financial Industry Message Scheme.
5 TOGAF – The Leading Framework for Enterprise Architecture

This chapter aims to provide an overview of TOGAF and the organization behind it, The Open Group. Its key focus areas are the Architecture Development Method (ADM), the TOGAF metamodel, and the TOGAF Content Framework. These areas are also those where leveraging BIAN yields the largest benefit.

5.1 The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry’s premier certification service

Further information on The Open Group is available at www.opengroup.org.

5.2 TOGAF as an Industry-Independent Framework for Enterprise Architecture

TOGAF Version 9 is an open, industry consensus framework and method for enterprise architecture. TOGAF has been developed through the collaborative efforts of more than 300 Architecture Forum member companies from some of the world’s leading companies and organizations. Using TOGAF results in enterprise architecture that is consistent, reflects the needs of stakeholders, employs best practice, and gives due consideration both to current requirements and to the perceived future needs of the business. TOGAF is focused on today’s business needs and expectations as it:

- Is rooted in best practices
- Is a generic framework for developing architectures to meet different business needs
- Is customizable, not a “one-size-fits-all” framework
- Considers most aspects of running an architecture practice
- Incorporates business/IT alignment
- Is an open standard that is vendor-, tool-, and technology-independent

The TOGAF standard is industry-independent; it does not contain any “vertical”, industry-specific content. While this ensures wide applicability, it imposes a challenge to its users, as they will not find any abstractions specific to the business they are supporting.

5.3 Overview of TOGAF Components

There are seven main parts in the TOGAF Version 9 document. Next to the introduction (Part I) the document covers the six main components of the TOGAF framework:

- The Architecture Development Method (ADM – described in Part II) is the core of TOGAF. It is a step-by-step approach to developing and managing enterprise architecture.
The ADM Guidelines & Techniques (Part III) is a collection of approaches which support the implementation of TOGAF and the application of the TOGAF ADM.

The Architecture Content Framework (Part IV) describes a framework for content used in an enterprise architecture initiative, including a structured metamodel for architectural artifacts, the use of re-usable architecture building blocks, and an overview of typical architecture deliverables.

The Enterprise Continuum (Part V) is a taxonomy to categorize architectural artifacts within an enterprise. It forms the basis for defining an Architecture Repository.

TOGAF Reference Models (Part VI) is a selection of technical architecture reference models, including the TOGAF Foundation Architecture, and the Integrated Information Infrastructure Reference Model (III-RM).

The TOGAF Architecture Capability Framework (Part VII) discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise.
5.3.1 TOGAF Architecture Development Method (Part II)

The TOGAF ADM provides a tested and repeatable process for developing architectures. The ADM includes establishing an architecture framework, developing an architecture for the organization, transitioning, and governing the realization of architectures. All of these activities are carried out within an iterative cycle of continuous architecture definition and realization that allows organizations to transform their enterprises in a controlled manner in response to business goals and opportunities.

The ADM is meant to be applied iteratively and in a non-linear fashion. Architecture is refined through multiple passes of the ADM cycle. Within an iteration, it may often be required to go back and forth between the phases, to elaborate further on deliverables sketched out earlier.

In executing the ADM, the architecture team not only develops a snapshot of the enterprise at particular points in time, it also populates the organization's Architecture Repository (see Section 5.3.3) with all the architectural assets identified and leveraged along the way. This not only comprises enterprise-specific architecture, but also artifacts used in the process (like the BIAN Service Landscape). Hence, over time, the architect gradually adds more and more content to the organization's Architecture Repository.

In fact, the first execution of the ADM will often be the hardest, since the architecture assets available for re-use will be relatively scarce. BIAN provides both service definitions for the application landscape and a tentative structure of business, which can accelerate development significantly.

5.3.2 Architecture Content Framework (Part IV)

Architects executing the ADM will produce a number of outputs such as process flows, architectural requirements, project plans, project compliance assessments, etc. The Content Framework provides a structured model for architectural content that allows the major work products that an architect creates to be consistently defined, structured, and presented.

This Content Framework may be extended or even replaced by another Content Framework; for example, mappings exist to use TOGAF for creating Frameworx/NGOSS-based architectures in the telecommunications industry, or for generating the DoDAF deliverables in the US Defense environment.

The integration with BIAN does not actually reach that far. The BIAN deliverables – in particular, the Service Landscape and the BIAN business objects – provide industry-specific reference models as specific starting points for elements of the TOGAF Content Framework, in particular for the logical application and data components within the Information Systems Architecture. A detailed explanation is provided in Section 6.4.
5.3.3 TOGAF Enterprise Continuum (Part V)

Performing enterprise architecture not only generates but also consumes a significant number of artifacts and deliverables – not only architecture and solution documents, but also standards and architecture management documents. These are maintained in an Architecture Repository.

To provide a better overview of these assets, TOGAF suggests the Enterprise Continuum as a framework for structuring the content of this repository. It has two major dimensions of classification:

- The Specialization dimension describes how generic or specific an architecture or a solution is. On the left of the Solutions Continuum, it collects industry-neutral components (Foundation Architectures). Moving right, it combines these generic building blocks into certain patterns to address specific generic challenges in an IT landscape (Common System Architectures), like a security or an operations architecture. Industry architectures address specific needs of vertical industries, and organization-specific architectures are re-usable artifacts with a scope and applicability of a specific organization.

- The Architecture – Solution dimension differentiates between abstract architecture building blocks available for the organization and their instantiation as solution building blocks.

In this context, BIAN provides an Industry Architecture for the banking industry; i.e., assets which are bound to the banking industry, but applicable to multiple organizations, and at architecture level; i.e., which have to be instantiated to become a solution (Section 6.5).
6 Leveraging the BIAN Deliverables with TOGAF

6.1 Introduction

This chapter will elaborate on the synergies of using BIAN in combination with TOGAF. To do so, it looks at TOGAF as the overall framework for architecture work and at BIAN as a pool of industry-specific architecture deliverables, set up with a specific goal. Based on that, it will analyze how the BIAN deliverables are related to the TOGAF framework. As to be expected, the BIAN deliverables have much ground in common with the TOGAF framework. In general, BIAN provides content in a specific structure. When applying TOGAF in a banking environment, BIAN content will contribute to speed up the work and improve quality. On the other hand, TOGAF facilitates architecture development work by providing a structured approach and a complete structure of relevant artifacts. Hence, TOGAF adds value to the BIAN deliverables especially for the project approach and capability to perform as well as by providing generic technology architectures.

Section 6.2 will analyze how BIAN deliverables can be used as input for the different ADM phases and how they may influence the way of working within each phase. Then, in Section 6.3 we will relate the BIAN deliverables and philosophy to some specific guidelines and techniques that may be relevant when applying the ADM cycle. Section 6.4 maps the BIAN deliverables on the TOGAF Content Framework and thus provides insight into which areas BIAN adds architecture content. Finally, Section 6.5 will position BIAN deliverables in the Enterprise Continuum to better understand the nature of BIAN deliverables in this respect.

Part VI (TOGAF Reference Models) and Part VII (Architecture Capability Framework) are not covered here: Part VI only relates to technical architectures, whereas BIAN only provides Business and Information Systems Architecture deliverables. With respect to Part VII, BIAN does not provide deliverables related to establishing and operating an architecture function within an enterprise.

6.2 Relating BIAN to the Phases of the ADM (Part II)

In this section we describe where the availability and use of BIAN deliverables may influence the way the TOGAF ADM phases are performed. Each chapter refers to the related chapter of the TOGAF Version 9 document.

For each TOGAF ADM phase, we only include those BIAN deliverables that are direct input for that ADM phase. Since the impact of those BIAN deliverables is fully integrated in the output of that ADM phase, the influence on later ADM phases is fully covered.

Applying the TOGAF ADM will produce and enrich a Service Landscape specific to the organization. The results could further be consolidated back to the BIAN community to extend and enrich the BIAN service landscape.

6.2.1 Preliminary Phase (Chapter 6)

This Preliminary Phase is about defining “where, what, why, who, and how we do architecture” in the enterprise concerned.

In this phase we should consider:

- That the existence of the BIAN network contributes to the awareness and acceptance of an architectural approach and in that sense can be used to create sponsorship and general commitment for the approach
- That the use of the BIAN framework and related deliverables may be prescribed by the architecture
- Whether or not to use BIAN input and principles during the architecture work
6.2.2 Architecture Vision (Phase A, Chapter 7)

This phase is about validating starting points, defining the scope and approach of the architecture development cycle, and recognizing key success factors.7

- Relate the architecture development cycle to the use of BIAN deliverables. In first instance, a decision has to be made regarding the relevance and fitness-for-purpose of the BIAN deliverables for this architecture engagement.

- The part of the BIAN Service Landscape that is relevant for the project can be used to identify possible stakeholders.

- The BIAN Service Landscape can also be used to identify relevant related other architecture developments.

- When using BIAN, certain re-use requirements may be applicable.

6.2.3 Business Architecture (Phase B, Chapter 8)

The main objectives of Phase B are to describe the Baseline Business Architecture, to develop a Target Business Architecture, and to identify the gaps.

- The BIAN business scenarios8 can be used as a starting point in defining the Baseline and Target Business Architecture.

- The BIAN Service Landscape is structured according to a common reference hierarchy: a business breakdown in Business Areas and Business Domains. Although not a Target Business Architecture9 in itself, it can be used as a starting point (or at least be used as a source of inspiration) for the set-up of the Target Business Architecture.

- As the BIAN Service Landscape is directly derived from the business breakdown, it should be clear where the Target Business Architecture deviates from this breakdown. This insight is needed to apply the BIAN Service Landscape in Phase C.

6.2.4 Information Systems Architecture (Phase C, Chapter 9)

The objective of Phase C is to develop Target Architectures for the data and/or the application systems domains, and identify the gaps between the Baseline and the Target Architecture. Information Systems Architecture focuses on identifying and defining the applications and data considerations that support an enterprise’s Business Architecture. It does so by defining views that relate to information, knowledge, application services, etc.

- BIAN provides a Service Landscape specific to the banking industry, constructed of re-usable building blocks related to application components and data entities. As such, this Service Landscape can be used as a reference point for defining or assessing the Target Application Architecture of the organization. Its breakdown into Business Areas, Business Domains, and Service Domains can be applied to structure the application landscape. The principles applied in constructing the Service Landscape can be translated into application and data principles.

---

7 In the Architecture Vision phase, TOGAF suggests to “evaluate business capabilities”. Although BIAN also uses the term capability, the underlying concept of a capability in BIAN (the ability to fulfil a responsibility that supports a specific aspect of business) is quite different in scope and nature from the capability term in TOGAF (as derived from a capability-based planning perspective). Hence, due to their granularity, BIAN “capabilities” are typically not an outcome of Phase A, but rather of Phase B or C.

8 As soon as they are publically available – compare Section 4.2.

9 A Target Business Architecture Framework will be part of the next release of the BIAN Service Landscape in 2012.
• A core activity will be to relate the identified BIAN service operations on the target application level to the Target Business Architecture (developed in Phase B), to recognize the relationship between business processes and applications, and to analyze the relation between business objects and data. The mapping of BIAN deliverables on the TOGAF Content Metamodel (see Section 6.4) supports the execution of this activity.

• The other way around, the BIAN initiative could benefit from this phase by updating the BIAN Service Landscape with the output of this analysis; e.g., the split-up or repositioning of a certain Service.

6.2.5 Technology Architecture (Phase D, Chapter 12)

The Technology Architecture phase seeks to map application components defined in the Information Systems Architecture phase onto a set of technology components.

• Since the BIAN service definitions are implementation-independent, BIAN does not contribute to this phase directly. However, the result of Phase C will certainly contain specific technical requirements coming from the BIAN service interfaces. (Note that BIAN service operations will be related to other standards such as ISO, IFX, etc., which may imply specific technology components.)

6.2.6 Opportunities and Solutions (Phase E, Chapter 13)

A goal of this phase is to derive a series of Transition Architectures that deliver continuous business value (i.e., capability increments) through leveraging opportunities to leverage the building blocks.

• The BIAN Service Landscape and other BIAN deliverables are created in co-operation with software vendors serving the banking industry. Hence it is fair to assume that BIAN services leveraged in previous ADM phases will also be physically available in the market as COTS software solutions or application components.

• The BIAN Service Landscape can also be used in vendor and package selections, assessing the compliance of vendors and products with the BIAN Service Landscape. Compliance of products to the BIAN Service Landscape will increase the seamless integration of the software products in an existing (BIAN-compliant) target Application Architecture and thus reduce integration cost.

6.2.7 Migration Planning (Phase F, Chapter 14)

The objectives of Phase F are to finalize a detailed Implementation and Migration Plan.

• There are no specific BIAN deliverables that are direct input for this ADM phase. All relevant BIAN input is included in the output of previous ADM phases.

6.2.8 Implementation Governance (Phase G, Chapter 15)

The goal of this phase is to govern and manage the implementation and deployment process.

• There are no specific BIAN deliverables that are direct input for this ADM phase. All relevant BIAN input is included in the output of previous ADM phases.

6.2.9 Architecture Change Management (Phase H, Chapter 16)

The goal of this phase is to establish an architecture change management process for the new enterprise architecture baseline.

• There are no specific BIAN deliverables that are direct input for this ADM phase. All relevant BIAN input is included in the output of previous ADM phases. However, the tracking of BIAN changes and initiatives should be incorporated in the architecture change management.
6.2.10 Requirements Management (Chapter 17)

The goal of requirements management is to define a process whereby requirements for enterprise architecture are identified, stored, and fed into and out of the relevant ADM phases.

- Leveraging BIAN deliverables in the ADM would guide and structure the capturing of requirements, in particular along the Service Landscape (including its Business Areas and Business Domains).

6.3 Relating BIAN to TOGAF Guidelines and Techniques (Part III)

This section focuses on specific guidelines and techniques related to the ADM cycle that are relevant when using BIAN deliverables in an architectural engagement. The chapters mentioned refer to the related chapters in the TOGAF Version 9 document.

6.3.1 Applying the ADM at Different Enterprise Levels (Chapter 20)

The ADM is intended to be used as a model to support the definition and implementation of architecture at multiple levels within an enterprise.

In general, it is not possible to develop a single architecture that addresses all needs of all stakeholders. As can be seen from the graph below, an enterprise can be partitioned into different areas, according to “Subject Matter”, “Time Period”, and “Level of Detail”.

The BIAN Service Landscape and other BIAN deliverables are especially useful in:

- The definition and/or assessment of the Target Architectures, especially Information Systems Architectures (Application and Data) and the related definition of change from baseline to target
- The assessment and selection of COTS solutions in the market, aimed at achieving compliance with BIAN deliverables and thus reducing integration cost

As BIAN does not focus on a comprehensive view of Business Architecture, it may be appropriate to apply an iterative approach to develop the relevant capability architectures at different levels of the enterprises in multiple passes of the ADM cycle.

The BIAN Service Landscape is based on SOA principles. It uses a hierarchical decomposition of the business in terms of Business Areas and underlying Business Domains. Business Domains might serve as a source of inspiration to define “segments”. Per Business Domain, the relevant Service Domains are recognized and
described. This is more related to the “capability architecture” level; however, BIAN does not provide the overall architectural insight at all levels of an organization (Company – Business Unit – Division – Department – …).

6.3.2 Using TOGAF to Define and Govern SOAs (Chapter 22)

This chapter discusses the factors related to the adoption and deployment of SOA within the enterprise.

- The BIAN Service Landscape is based on SOA principles. Hence, the architect should be aware whether and how SOA is leveraged within the organization. Where necessary, the architecture engagement should include activities required to apply SOA principles in the organization.
- To be effective, it is necessary to align TOGAF and BIAN terminology beforehand.

6.3.3 Architecture Principles (Chapter 23)

This chapter describes principles for use in the development of an enterprise architecture.

The definition of a principle is: “a qualitative statement of intent that should be met by the architecture”. This applies to principles:

- That govern the architecture process
- Which are affecting the development, maintenance, and use of the enterprise architecture
- That govern the implementation of the architecture

BIAN has developed various design rationales and supporting design techniques, mainly in order to define a Service Domain. Examples are:

- Operational properties: Clearly bounded business purpose, focus object record, and full life-cycle handling, exclusively service-based.
- Design techniques: Right-sizing the focus objects; assigning a single functional pattern; confirming the structure of its capabilities; confirming the make-up of its service boundary.

Not all BIAN design principles are canonical and fully and explicitly described in the BIAN Metamodel. As the construction of the BIAN Service Landscape is based on these specific design principles, it is important to make them explicit during the execution of the different ADM phases.

6.3.4 Architecture Patterns (Chapter 25)

In TOGAF a “pattern” is defined as: “an idea that has been useful in one practical context and will probably be useful in others”.

- Patterns describe how, when, and why building blocks can be applied, and which trade-offs have to be made in doing so. In that sense, BIAN can be used as an additional source for best practices. In particular, the business scenarios describe typical patterns of service usage and service interaction. Additionally, the how-to guide associated with the Service Landscape provides patterns for service design.

6.3.5 Interoperability Requirements (Chapter 29)

Defining the degree to which the information and services are to be shared.

- This is a very useful architectural requirement, especially in a complex organization and/or extended enterprise and key in BIAN’s focus. BIAN provides guidelines (in terms of structure with the business landscape or Service Domain design principles) that should be met with respect to interoperability.
6.4 BIAN and the TOGAF Architecture Content Framework (Part IV)

The TOGAF Architecture Content Framework consists of the relevant artifacts produced in the ADM cycle. As shown earlier, the use of BIAN is relevant to several phases of the ADM (Section 6.2). These dependencies make it straightforward to map the BIAN deliverables onto the TOGAF Content Metamodel.

It may be noted that we only map the specific BIAN deliverables that are direct input in an ADM phase. This means that not all parts of the TOGAF metamodel are related to specific BIAN deliverables.

6.4.1 Deliverables, Artifacts, and Building Blocks

The figure below shows the relations between the different concepts (Deliverables, Artifacts, and Building Blocks) in TOGAF. Building Blocks refer to descriptions of specific parts of an architecture. They are devised in architecture building blocks and solution building blocks.

BIAN focuses on those IT services relevant to the banking industry. The structure is based on a common understanding of the business and systems in use. The business scenarios are used to validate the completeness of the Service Landscape.

In TOGAF terminology, the BIAN Service Landscape is a structured description of the various architecture building blocks needed to provide the required capabilities in the banking industry.

6.4.2 Mapping the BIAN Deliverables to the TOGAF Content Metamodel

The following picture shows the mapping of BIAN deliverables onto the TOGAF Content Metamodel. They are further detailed in the following sections.

- BIAN Service Domain principles ensure consistency within a Business Domain.
- BIAN business scenarios, currently used to validate the Service Landscape and to ensure completeness, may be used as best practice process templates in the future.
- BIAN focuses on application-to-application integration relevant for the various logical application components.
- BIAN service definitions are canonical, implementation-independent descriptions of logical components; hence, physical and technical layers are out of scope.
- BIAN Services are clustered around focus objects and manage their full life-cycle.
Focus objects are relevant to logical Data Architecture.

Using BIAN deliverables in an architecture engagement may require a TOGAF Content Metamodel extension with respect to the BIAN Service Landscape. The services extension is intended to allow more sophisticated modeling of the service portfolio by creating a concept of IT services in addition to the core concept of business services.

Service Domain and Landscape Design Principles

The BIAN Service Landscape is based on a set of design principles. These design principles map smoothly to the architecture principles allocated in the TOGAF Content Metamodel.

Business Scenarios

BIAN has defined the business scenario as a “pattern of collaboration between a suitable selection of Service Domains”. The business scenario clarifies the role each Service Domain plays in the scenario (executing its own specific capabilities) and the interactions between all involved Service Domains. These interactions occur via the industry standard semantic services.

The business scenarios are not canonical and are not explicitly described in the BIAN Metamodel.

The business scenario may be mapped onto processes where a process represents the flow of control between or within functions and/or services (depending on the granularity of definition). Processes represent a sequence of activities that together achieve a specified outcome, can be decomposed into sub-processes, and can show the operation of a function or service (at the next level of detail).

Business Objects

Business objects play a fundamental role in the definition of BIAN Service Domains, as each Service Domain has one focus object whose instances it uniquely owns and whose life-cycle it manages.

BIAN business objects may be mapped on the following high-level TOGAF Metamodel entities:
- Data Entity: An encapsulation of data that is recognized by a Business Domain expert as “a thing”. Logical data entities can be tied to applications, repositories, and services and may be structured according to implementation considerations.

- Logical Data Component: A boundary zone that encapsulates related data entities to form a logical location to be held; for example, external procurement information.

Because the BIAN business objects are independent of any implementation, they cannot be mapped onto a physical data component.
Service Landscape

BIAN aims to address application-to-application integration issues. The BIAN Service Landscape is a coherent set of IT logical capabilities/functionalities that are recognized by the business.

This corresponds well to the TOGAF definition described in TOGAF Section 34.4.2 (Services Extensions, Purpose): “The services extension is intended to allow more sophisticated modeling of the service portfolio by creating a concept of IS services in addition to the core concept of business services. IS services are directly supported by applications and creating the layer of abstraction relaxes the constraints on business services whilst simultaneously allowing technical stakeholders to put more formality into an IS service catalog in TOGAF”.

This TOGAF service catalog may be considered as a generalization of the catalog of BIAN Service Domains or service operations.

For these reasons, the BIAN Service Landscape elements and the capabilities/responsibilities/delegations can be interpreted as being:

- Logical Application Component: “An encapsulation of application functionality that is independent of a particular implementation. For example, the classification of all purchase request processing applications implemented in an enterprise.”

- Application Service: “The automated elements of a business service. An information system service may deliver or support part or all of one or more business services.”

Because the BIAN IT services are independent of any implementation, BIAN does not provide any physical application component (in the sense of “an application, application module, application service, or other deployable component of functionality”). However, software vendors or software development organizations may provide BIAN-compliant software components which can be implemented in the IT landscape.

6.5 BIAN and the TOGAF Enterprise Continuum (Part V)

Related to the TOGAF Enterprise Continuum, BIAN content can be seen as an industry-specific architecture within the Architecture Continuum (Section 5.3.3); as BIAN Service Landscape and service definitions are implementation-independent, they are not related to the Solution Continuum.10

Looking at the Architecture Repository containing all architectural deliverables (see the figure below), BIAN contributes by providing a banking industry-specific reference model (red circles in the picture).

It contains a Common Systems Architecture as well as data standards and application standards relevant for the architecture. All of these BIAN deliverables can be included in the Architecture Repository or treated as external references and models. In addition, BIAN developed and published a metamodel for the Service Landscape which can be considered an architecture metamodel in the context of the Architecture Repository.

---

10 Still, describing structures like the SOA architecture style is actually defining properties of the Solution Continuum.
7 Other Relevant Standards for the Banking Industry

7.1 Alignment of BIAN with Other Technical, Semantic, and Messaging Standards

BIAN is the first standards body to define semantic service standards for the financial services domain; hitherto the majority of the standardization activity has focused on B2B requirements at a lower “implementation” level. The following diagram shows how BIAN is positioned alongside a number of other standards groups that are active in Financial Services.

It is the policy of BIAN to align and contribute to service standards where they exist, rather than developing competing standards. BIAN has and will continue to test and confirm that its semantic standards can be mapped to existing standards as appropriate. In particular, BIAN seeks to maintain close alignment and integration with the ISO 20022 standard initiative.

- BIAN’s Application-to-Application (A2A) focus differs from others where B2B or company-to-company is typically the priority.
- BIAN’s standards are implementation-independent, semantic service definitions that can be interpreted or mapped to implementation-level standards and proprietary solutions.
- BIAN, IFX, and the OMG Finance Domain Task Force recognize ISO 20022 and the SWIFT administered ISO 20022 Repository as core to alignment of standards initiatives in the financial services industry.
- BIAN employs a service center approach to SOA.

7.2 APQC as a Process Standard

APQC\textsuperscript{11} is a non-profit organization focusing on benchmarking, measurement, and process improvement. In particular, it allows its members to benchmark their processes against other members and best practices without compromising confidentiality.

\textsuperscript{11} Refer to: www.apqc.org.
As a basis for benchmarking, it publishes a Process Classification Framework (PCF) as an open standard for structuring the processes of an institution. This framework has been specialized for the banking industry. This industry-specific PCF can be used on the one hand as a starting point and reference for developing the Business Architecture. On the other hand, aligning the Business Architecture with the PCF allows accessing the benchmark data pool of APQC to measure process effectiveness, and identify hotspots for improvement.

---

8 Glossary

Architecture Content Framework [TOGAF]: Provides a structural model for architectural content that allows major work products to be consistently defined, structured, and presented.

Architecture Repository [TOGAF]: Can be used to store different classes of architectural output at different levels of abstraction, created by the ADM.

ADM/Architecture Development Method [TOGAF]: Forms the core of TOGAF and is designed to address enterprise business and IT needs by providing a set of architecture views, guidelines on tools for architecture development, a set of recommended deliverables, and a method for managing requirements.

BA/Business Area [BIAN]: Formed by a broad set of capabilities and responsibilities and an element at the highest level of the hierarchy when decomposing the functions of financial institutions. This decomposition is primarily driven by the business understanding and complemented by application and information-specific needs.

BD/Business Domain [BIAN]: A coherent set of capabilities and responsibilities. It is an element of the functional decomposition of the banking business functions in the context of the Service Landscape. Business Domains are linked to certain skills and knowledge, which are clearly identifiable in the banking business.

BIAN/Banking Industry Architecture Network [BIAN]: A not-for-profit organization which seeks to accelerate the adoption of Service-Oriented Architecture (SOA) in the banking industry. It does so by promoting convergence towards a common service landscape, and by providing semantic standards which makes it easier and more cost-effective to integrate such services (www.bian.org).

Capability [BIAN]: The direct fulfillment of a responsibility by a Service Domain. It sets out the main actions supported corresponding to generic stages in the life-cycle of the Service Domain’s focus object (e.g., tasks to set up/update a new instance of the focus object, maintenance tasks, access/invocation tasks, analysis, reporting, and archiving tasks).

COTS/Commercial Off-The-Shelf: Pre-packaged software product, ready to be used “as-is”.

EC/Enterprise Continuum [TOGAF]: A view of the Architecture Repository that provides methods for classifying architecture and solution artifacts as they evolve from generic Foundation Architectures to Organization-Specific Architectures. The Enterprise Continuum comprises two complementary concepts – the Architecture Continuum and the Solutions Continuum.

Focus Object [BIAN]: A type of business object, representing a record (which may be machine-representable), for which an instance of the Service Domain manages the complete life-cycle (initialization, maintenance, execution, reporting, and termination) in fulfilling its business purpose.

IFX (Interactive Financial eXchange forum): A content-rich financial messaging protocol built by financial industry and technology leaders. IFX is a consistent framework incorporating best-of-breed design principles, a common object model, and a service-oriented architecture that accounts for the interactions among those objects and related data definitions.

ISO 20022: Provides the financial industry with a common platform for the development of messages using a UML-based modeling methodology, a central dictionary of business items used in financial communications, and a set of XML design rules to convert the messages described in UML into XML schemas.

SaaS (Software as a Service): Sometimes referred to as “on-demand software,” is a software delivery model in which software and its associated data are hosted centrally (typically in the Cloud) and are typically accessed by users using a thin client, normally using a web browser over the Internet.

SD/Service Domain [BIAN]: Has a unique business purpose and manages an artifact or record of its execution. The Service Domain represents the finest-grained design building block, encapsulating its functionality behind a semantic service boundary.

SOA/Service-Oriented Architecture: An architectural style in which the functionality of an application is exposed (for re-use) via a well-defined interface.
**TCO/Total Cost of Ownership**: A method to determine direct and indirect costs of a product or system throughout the life-cycle.

**TOGAF [TOGAF]**: An industry standard architecture framework that may be used freely by any organization wishing to develop an Information Systems Architecture for use within that organization (www.opengroup.org/togaf).
9 APPENDIX: The BIAN Metamodel

The concepts described in Chapter 4 (Business Area, Business Domain, etc.) as well as their relationships are formalized in the BIAN Metamodel. They are structured in packages and viewpoints. Within the scope of this White Paper, the following viewpoints are the most relevant:

- **Service Landscape Viewpoint**: Highlights key elements of the service landscape.
- **Capabilities-Responsibilities-Delegation Viewpoint**: Shows the definition and relationships of capabilities, responsibilities, and delegations.
- **Business Object Viewpoint**: Describes business objects and their attributes as related to Service Domains.
- **Message Viewpoint**: Messages play a fundamental role in the definition of ServiceOperations, as a ServiceOperation may have an input message, may have an output message, and must have a fault message.

The BIAN Metamodel is an extension of the ISO 20022 metamodel. It also has a generic mechanism to relate Service Landscape content to content in other standards repositories.

### 9.1 Service Landscape Viewpoint

As explained above, BIAN wants to reap the benefits of the agility and interoperability promises of SOA by defining IT services in a manner that is semantics-focused, standards-based, and independent of any technology platform.

In addition, these IT services need to be defined in a consensus mode by the BIAN members and validated by the banking industry. This consensus includes the need for a common logical functional landscape to organize the defined services in a coherent set of logical capabilities/functionalities. This is the so-called BIAN Service Landscape.

The figure below illustrates the metamodel of the BIAN Service Landscape: it is basically a hierarchical structure of different classes of services.

- **BusinessArea**: Formed by a broad set of capabilities and responsibilities. Business Areas are used to decompose the functions of financial institutions according to their main business responsibility, operational behavior, or architectural features; e.g., “operations & execution”, “reference data”, “sales & service”. BusinessAreas are decomposed (i.e., subdivided) into BusinessDomains.

- **BusinessDomain**: An element of the functional decomposition of the banking business functions in the context of the Service Landscape. BusinessDomains are linked to certain skills and knowledge, which are clearly identifiable in the banking business (e.g., “payments” is a BusinessDomain of the BusinessArea “operations & execution”).

- **ServiceDomain**: Represents an “atomic” logical design. Atomic means that a BIAN ServiceDomain represents the smallest practical capability or functional partition that can be service-enabled as a discrete and unique business capability (e.g., “payments execution” is a ServiceDomain of the “payments” BusinessDomain). All BIAN ServiceDomains taken together make up a “peer set” with each performing its own specific business function or purpose. The collection of all BIAN ServiceDomains within a reference hierarchy of Business Areas and Business Domains is called the BIAN Service Landscape.

- **ServiceGroup**: A set of ServiceOperations, and is owned by a ServiceDomain. In essence, it is an interface to the ServiceDomain that is defined in terms of business semantics rather than in technical IT terms. For example, “manage payment order” is a ServiceGroup.

- **ServiceOperation**: Represents a service defined at the level of business semantics, specifying the access to one or more capabilities of a ServiceDomain (e.g., “Execute Payment Order” or “Update
Payment Order” are ServiceOperations of the ServiceGroup “Manage Payment Order”). ServiceOperations have input messages and output messages.

- **BIANConstraint**: ServiceOperations may have constraints such as pre-conditions or post-conditions.

BIAN does not define specific BusinessAreas and BusinessDomains as canonical standards: It is not possible to force a particular hierarchical decomposition of business function on all banks. The BusinessAreas and BusinessDomains that BIAN publishes in its Service Landscape provide a reference model only. The BIAN ServiceDomains, on the other hand, are canonical, and are designed to fit into an arbitrary number of BusinessArea – BusinessDomain hierarchies.
9.2 Capabilities-Responsibilities-Delegation Viewpoint

The concept of a ServiceDomain is central to BIAN’s mission of defining canonical service standards. A key principle behind a BIAN ServiceDomain is that it is responsible for some aspects of the business. In fulfilling its responsibilities, it will implement some of them directly using its own capabilities, and may delegate others to other ServiceDomains through service interactions.

A **Responsibility** is a commitment by a ServiceDomain to ensure the achievement of a business purpose. A ServiceDomain can fulfill a responsibility by means of one of its own capabilities or by delegation to other ServiceDomains.

A **Capability** is the capacity of a ServiceDomain to fulfill a responsibility that it owns without relying on other Service Domains.

A **Delegation** is the fulfillment of a responsibility owned by a ServiceDomain, achieved by enlisting the aid of another ServiceDomain.

As an example, “payments execution” is a ServiceDomain which has responsibilities that it fulfills in itself, such as “enrich payments” and “validate payments”, and responsibilities that it delegates to other ServiceDomains, such as “account status check” and “calculate fees”.

[Diagram of ServiceDomain interactions and responsibilities]

A Delegation is the fulfillment of a responsibility owned by a ServiceDomain, achieved by enlisting the aid of another ServiceDomain.
9.3 Business Object Viewpoint

The diagram below shows the elements of the metamodel that pertain to BusinessObjects. BusinessObjects play a fundamental role in the definition of ServiceDomains, as each ServiceDomain has one focusObject whose instances it owns uniquely.

**BusinessObject:** An individually distinguishable element characterized by a well-defined identity, structure, and behavior. A ServiceDomain also owns a BusinessObject called its focusObject. A BusinessObject is a specialization of the ISO 20022 BusinessComponent.
9.4 Message Viewpoint

This diagram below shows the elements of the metamodel that pertain to Messages. Messages play a fundamental role in the definition of ServiceOperations, as a ServiceOperation may have an input message, may have an output message, and must have a fault message.

Much of the internal structure of a Message is not visible on this diagram because that structure is defined by the ISO 20022 Metamodel, of which the BIAN Metamodel is an extension. The diagram below reveals the following:

- A BIAN Message is a specialization of an ISO 20022 MessageDefinition.
- A BIANMessageComponent is a specialization of an ISO 20022 MessageComponent. MessageComponents have BusinessAttributes and also can be associated with other MessageComponents.
- A BIANMessageAttribute is a specialization of an ISO 20022 MessageAttribute. There is a similar relationship between BIANMessageAssociationEnd and ISO 20022 MessageAssociationEnd.
- BIANMessageBuildingBlock is a specialization of ISO 20022 MessageBuildingBlock. In ISO 20022, a MessageBuildingBlock defines how a MessageComponent or DataType is assembled into a Message.

The relationships between BusinessObjects and Messages are also not visible on this diagram, because those relationships are defined by the ISO 20022 metamodel where it defines the relationships between BusinessComponents and MessageComponents.