BIAN EDITION 2019

A framework for the financial services industry
Other publications by Van Haren Publishing

Van Haren Publishing (VHP) specializes in titles on Best Practices, methods and standards within four domains:
- IT and IT Management
- Architecture (Enterprise and IT)
- Business Management and
- Project Management

Van Haren Publishing is also publishing on behalf of leading organizations and companies: ASLBiSL Foundation, BRMI, CA, Centre Henri Tudor, Gaming Works, IACCM, IAOP, IFDC, Innovation Value Institute, IPMA-NL, ITSqc, NAF, KNVI, PMI-NL, PON, The Open Group, The SOX Institute.

Topics are (per domain):

**IT and IT Management**
- ABC of ICT
- ASL*
- CATS CM*
- CMMI*
- COBIT*
- e-CF
- ISO/IEC 20000
- ISO/IEC 27001/27002
- ISPL
- IT4IT*
- IT-CMF™
- IT Service CMM
- ITIL*
- MOF
- MSF
- SABSA
- SAF
- SIAM™
- TRIM
- VeriSM™

**Enterprise Architecture**
- ArchiMate*
- GEA*
- Novius Architectuur
- Methode
- TOGAF*

**Business Management**
- BABOK* Guide
- BiSL* and BiSL* Next
- BRMBOK™
- BTF
- EFQM
- eSCM
- IACCM
- ISA-95
- ISO 9000/9001
- OPBOK
- SixSigma
- SOX
- SqEME*

**Project Management**
- A4-Projectmanagement
- DSDM/Atern
- ICB / NCB
- ISO 21500
- MINCE*
- M_o_R*
- MSP*
- P3O*
- PMBOK* Guide
- PRINCE2*

For the latest information on VHP publications, visit our website: www.vanharen.net.
BIAN Edition 2019

A framework for the financial services industry

Contributing authors:
Guy Rackham
Hans Tesselaar
Klaas de Groot
Foreword

Why this book?

It’s been over 10 years now that some influential players in the financial services industry bundled their forces to stop the ever growing cost for IT integration. The Banking Industry Architecture Network was born.

So now after 10 years of hard work of all members in our community we have packaged all the knowledge and insights in this book.

There’s never been a more exciting time to be part of the financial services industry. The pace of the change has never been greater, the competitive landscape continues to expand beyond traditional players and emerging technologies are opening doors that allow us to find new ways to differentiate ourselves and explore the art of the possible. But none of this will be possible using traditional approaches.

At BIAN we believe that the Banking industry wastes over a billion dollars each year due to the complexity of our core technologies and integration approaches that only ignore the problem, if not add to the dilemma. This has become one of the primary reason Banks are not getting anticipated benefits from their digital transformations. We must rid ourselves of the anchor that is slowing us down which is proprietary core banking solutions that are today’s legacy technologies only to be tomorrows if we do not change. We need to stop trying to predict the future but as an industry start taking responsibility to define a much more efficient and effective approach. This is where BIAN steps in. We are enabling a unique opportunity to migrate away from existing outdated core systems and move into a fully digital new world supported by Industry Standards. An open standard that establishes a utility for the industry. Virtually eliminating integration costs, leveling the playing field for anyone who develops against the standard and unleashing the power of the Cloud by giving Banks the freedom to have a choice to buy interchangeable micro services regardless who develops them.
This book covers all aspects of Architecture for the financial services industry. It should support all involved to help their organizations to enter a truly digital world.

Besides our original Service Oriented view, the authors also included our latest insight on Enterprise Architecture and give you guidance in the fast evolving API arena.

I’ll hope you will find what you need to perform your architecture role at its peak.

Enjoy reading.

Steve Van Wyk
Executive Vice President, Head of Technology and Operations, PNC Financial Services Group and Chairman of the BIAN board
# Contents

## Part I

### 1 INTRODUCTION

1.1 Who this book is intended for ........................................ 3  
1.2 How to use this book ................................................ 3  
1.3 BIAN, the Banking Industry Architecture Network ................. 4  
1.4 The BIAN Service Landscape, an overview ........................... 5  

## Part II

### 2 BIAN’S PRIMARY PURPOSE AND APPROACH

2.1 Introduction ...................................................................... 7  
2.2 A different approach to a well-established problem ................... 8  
2.2.1 BIAN’s capability view versus a traditional process view .......... 8  
2.3 BIAN in the context of other standard efforts in the industry .......... 10  
2.3.1 Standardization in the financial services industry .................. 11  
2.3.2 Support for industry standards ...................................... 12  

### 3 UNDERSTANDING THE THEORY

3.1 Introduction ...................................................................... 17  
3.2 Some key terms/concepts ................................................ 18  
3.3 Business capability partitions ........................................... 19  
3.4 Modeling real world behaviors .......................................... 20  
3.5 The BIAN standard can be interpreted in different situations ...... 21  
3.6 How to combine a static and a dynamic view in your model ........ 22
5.5.1 Translating at the business architecture level .......................... 60
5.5.2 Translating at the information architecture level ....................... 60
5.5.3 The Control Record can be modeled ...................................... 63
5.5.4 Translating at the application level ....................................... 64
5.5.5 Translating at the infrastructure level .................................... 65
5.5.6 Translating summary ........................................................ 67

5.6 Applying BIAN Service Domains in different environments .......... 68
5.6.1 Using BIAN specifications as a high-level implementation design .. 68
5.6.2 Service-oriented architectures and the benefits of
‘externalization’ ........................................................................ 69
5.6.3 Defining BIAN’s concept of ‘externalization’ ............................. 70
5.6.4 Externalization in business application design ......................... 73
5.6.5 Business to technical architecture – mapping Service Domains ... 74
5.6.6 Business architecture versus systems architecture views of a
Service Domain ....................................................................... 76
5.6.7 Service Domain clusters ....................................................... 77
5.6.8 Mapping implementation level functionality to a
Service Domain ....................................................................... 79
5.6.9 Possible Service Domain functional specializations ................... 81
5.6.10 Extending the functional definition of the Service Domain ........ 81
5.6.11 Mapping Service Operations to messages ............................... 82

5.7 Using the BIAN models to define (open) APIs ............................ 88
5.7.1 Semantic APIs .................................................................. 88
5.8 Service-based access ............................................................... 91

5.9 Applying BIAN in different technical architectures .................... 95
5.9.1 Level 1 - Conventional (legacy/core) system rationalization ....... 96
5.9.2 Level 2 - Host renewal/ESB integration and application/system
assembly ............................................................................. 100
5.9.2.1 Host alignment ............................................................ 101
5.9.2.2 Multiple candidate hosts .............................................. 103
5.9.3 Level 3 - Loose coupled distributed/cloud systems ................. 105
5.9.3.1 Service information precision ...................................... 107

5.10 Support for emerging industry approaches .............................. 109
5.10.1 Application Program Interfaces (APIs) ................................ 109
5.10.2 Micro-services ................................................................. 110

5.11 Using BIAN Service Domain partitions to define APIs ............... 111
5.11.1 Cross-technical platform solutions ....................................... 113
5.11.1.1 Specifying point solution requirements – accelerator
pacs ...................................................................................... 113
5.11.2 Business case development ............................................... 115
5.11.3 Select and amend Business Scenario(s) ............................... 115
5.11.3.1 Develop a Wireframe model ....................................... 117
5.11.4 Define the implementation requirements .............................. 117
5.11.4.1 Feature checklists ........................................... 117
5.11.4.2 Service Operations ......................................... 118
5.11.4.3 Business Scenarios and Wireframes ....................... 119
5.11.5 Map and assess existing systems/candidate packages ...... 121
  5.11.5.1 Functional coverage ...................................... 121
  5.11.5.2 Service enablement ........................................ 122
5.11.6 Candidate system ‘hygiene factor analysis’ .................. 123
  5.11.6.1 More general considerations when implementing point
        solutions .................................................. 124
5.11.7 Customization/development .................................... 125
5.11.8 Migration planning ............................................. 125
5.12 Support for incremental adoption/migration ..................... 126
  5.12.1 Using BIAN as an API ‘inventory’ ......................... 126
  5.12.2 API inventory ................................................ 129
5.12.3 Three levels of architectural alignment ...................... 131
  5.12.3.1 Direct to core .......................................... 133
  5.12.3.2 Wrapped host .......................................... 134
  5.12.3.3 Micro-service architecture ............................... 136
5.13 Limitations ......................................................... 137

Part IV ....................................................................... 139

6 ASSEMBLING A REPRESENTATIVE ENTERPRISE BLUEPRINT .......... 141
  6.1 Building the enterprise blueprint for a bank ..................... 143
  6.1.1 Select Service Domains that match the enterprise activity 144
  6.1.2 Adapt the general BIAN specifications as necessary .......... 145
  6.1.3 Assemble Service Domains in a structure matching
        the enterprise .............................................. 145
  6.1.4 Matching the enterprise segmentation approach ................ 147
  6.2 Case study .......................................................... 148

7 AN ENTERPRISE BLUEPRINT IS A FRAMEWORK FOR ANALYSIS .... 151
  7.1 The BIAN specifications can be augmented ...................... 152
  7.1.1 Feature attribution ............................................ 153
  7.2 Track business and technical performance ...................... 156
  7.3 Overlay resources to identify shortfalls ........................ 156
  7.4 Analysis supported by the enterprise blueprint .................. 157
  7.5 Linking between business and technical assessments .......... 158
Part V

APPENDIX 1: SERVICE DOMAIN DESCRIPTIONS (JANUARY XX8)  ........................................... 161

APPENDIX 2: BIAN AND TOGAF’S ADM PHASES ................................................................. 181

A2.1 Relating BIAN to the phases of the ADM ................................................................. 181
   A2.1.1 Preliminary phase ............................................................................................... 182
   A2.1.2 Architecture vision ......................................................................................... 182
   A2.1.3 Business architecture .................................................................................... 182
   A2.1.4 Information systems architecture ................................................................ 183
   A2.1.5 Technology architecture .............................................................................. 183
   A2.1.6 Opportunities and solutions ......................................................................... 184
   A2.1.7 Migration planning ....................................................................................... 184
   A2.1.8 Implementation governance ......................................................................... 184
   A2.1.9 Architecture change management ............................................................... 184
A2.2 Requirements management .................................................................................. 185
A2.3 Relating BIAN to TOGAF guidelines and techniques .......................................... 185
   A2.3.1 Applying the ADM at different enterprise levels ........................................... 185
   A2.3.2 Using TOGAF to define and govern SOAs .................................................... 185
   A2.3.3 Architecture principles .................................................................................. 186
   A2.3.4 Architecture patterns .................................................................................... 187
   A2.3.5 Interoperability requirements ....................................................................... 187
A2.4 BIAN and the TOGAF Architecture Content Framework .................................. 187
   A2.4.1 Deliverables, artifacts and building blocks ..................................................... 188
   A2.4.2 Mapping the BIAN deliverables to the TOGAF Content Metamodel .............. 188

APPENDIX 3: THE BIAN ORGANIZATION ................................................................. 191

A3.1 General Assembly ................................................................................................. 191
A3.2 Board of Directors ............................................................................................... 192
A3.3 Secretariat ............................................................................................................ 192
A3.4 Working Groups ................................................................................................... 192
A3.5 BIAN special projects ......................................................................................... 193
A3.6 Communication between a member and BIAN ................................................... 193
A3.7 Official roles of members ..................................................................................... 193
A3.8 BIAN events and Chapter Meetings ................................................................... 194
   A3.8.1 Scope and content ......................................................................................... 194
   A3.8.2 Where should members participate? ............................................................... 194
   A3.8.3 Location and frequency ................................................................................ 195
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Components of the BIAN Service Landscape</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Comparing enterprise and city planning</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Building without a plan – shanty town and application portfolio</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Migrating to a well architected application map</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>BIAN in the context of other standards</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>The central role of ISO 20022</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>Design principles and techniques</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>Static structures and dynamic use</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>The Service Landscape framework</td>
<td>34</td>
</tr>
<tr>
<td>10</td>
<td>The BIAN Service Landscape</td>
<td>36</td>
</tr>
<tr>
<td>11</td>
<td>Periodic table and different BIAN Service Landscape views</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Key properties of BIAN Service Domains</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>Clarifying points for determining the correct scope</td>
<td>40</td>
</tr>
<tr>
<td>14</td>
<td>Simple Business Scenario with rules</td>
<td>46</td>
</tr>
<tr>
<td>15</td>
<td>Example Business Scenario in MagicDraw</td>
<td>48</td>
</tr>
<tr>
<td>16</td>
<td>A payment transaction mapped on a Wireframe view</td>
<td>50</td>
</tr>
<tr>
<td>17</td>
<td>An example of the Service Operation connections for a Service Domain</td>
<td>51</td>
</tr>
<tr>
<td>18</td>
<td>A Wireframe showing the main Service Operations for a collection of Service Domains</td>
<td>51</td>
</tr>
<tr>
<td>19</td>
<td>Levels of completion of Service Domains</td>
<td>52</td>
</tr>
<tr>
<td>20</td>
<td>Other mapping considerations</td>
<td>60</td>
</tr>
<tr>
<td>21</td>
<td>The association between the BIAN standard and prevailing model views</td>
<td>66</td>
</tr>
<tr>
<td>22</td>
<td>Mapping Service Domains down the stack</td>
<td>67</td>
</tr>
<tr>
<td>23</td>
<td>Point solutions environment: Legacy re-alignment</td>
<td>71</td>
</tr>
<tr>
<td>24</td>
<td>Mapping business applications to Service Domains</td>
<td>76</td>
</tr>
<tr>
<td>25</td>
<td>Aligning utility and common solution application modules to Service Domains</td>
<td>77</td>
</tr>
<tr>
<td>26</td>
<td>Mapping Service Landscape with shared and common solutions</td>
<td>78</td>
</tr>
<tr>
<td>27</td>
<td>Example cluster for a retail financial services business application</td>
<td>80</td>
</tr>
<tr>
<td>28</td>
<td>Four types of input and output parameters</td>
<td>87</td>
</tr>
<tr>
<td>29</td>
<td>Semantic API design scheme</td>
<td>90</td>
</tr>
</tbody>
</table>
Figure 30: Design topics included in the API scheme
Figure 31: Design topics selected for four typical types of exchange
Figure 32: BIAN action terms
Figure 33: Default action term by functional pattern
Figure 34: Example of a BIAN API exchange
Figure 35: Service Domain broken into a functional core and a service wrapper
Figure 36: Using BIAN Service Domain partitions for comparisons
Figure 37: Externalizing Service Domains in an application
Figure 38: The use of BIAN Service Domains to define Service Domains to define a service directory for the ESB
Figure 39: ESB solutions integrating host and cloud-based service solutions
Figure 40: Advanced 'loose coupled' development
Figure 41: Advanced cloud technology solutions
Figure 42: Mapping BIAN to a cloud-based environment
Figure 43: BIAN Service Domains related to (micro)-services
Figure 44: Cloud-based services for a relationship management Service Domain
Figure 45: Example Business Scenario with rules
Figure 46: A payment transaction mapped on a Wireframe view
Figure 47: The completed payments area Wireframe (example)
Figure 48: Feature list for a Service Domain - Customer Credit Rating
Figure 49: Mapping candidate systems to the feature list of a Service Domain
Figure 50: Overlaying current systems on a Wireframe model
Figure 51: Example hygiene factor analysis
Figure 52: The BIAN Service Landscape – First API Inventory
Figure 53: Wave 1, Service Landscape coverage
Figure 54: Offer Management – scoping statement
Figure 55: Offer Management Wireframe
Figure 56: Summary of the API sophistication levels
Figure 57: Level 1 Layout
Figure 58: Level 2 layout
Figure 59: Level 3 Layout
Figure 60: The scope of BIAN’s M4 Bank model
Figure 61: From the conventional Service Landscape to the value chain layout.
Figure 62: Three steps in developing an enterprise blueprint
Figure 63: Two value chain elements representing different lines of business
Figure 64: Two lines of business connected to a regional operation
Figure 65: M4Bank with local units, regional and head office reporting
Figure 66: Mapping product and customer types to segmentation views
Figure 67: Enterprise analysis: a measurement framework
Figure 68: Enterprise analysis: a measurement framework for cost of staff
Figure 69: Attribution quadrant with an attributed value chain element
Figure 70: Example approaches associated with an attribution
Figure 71: Systems and operational cost and performance measures
Figure 72: Overlay of systems on an enterprise blueprint revealing shortfalls 157
Figure 73: BIAN designs applied to point and enterprise solution 157
Figure 74: Using the enterprise blueprint for planning & analysis 158
Figure 75: BIAN designs help bridge between point solutions and enterprise viewpoints 181
Figure 76: Relating BIAN to the phases of the ADM 186
Figure 77: Different areas of an enterprise 188
Figure 78: Deliverables, artifacts and building blocks 189
Figure 79: Mapping BIAN deliverables onto the TOGAF Content Metamodel 191
PART I
1.1 WHO THIS BOOK IS INTENDED FOR

This book is intended for those enterprise, business and solution architects in the financial services industry (FSI) who are interested in applying the BIAN Industry Standard in their organization. The authors of the book expect the readers to have an in-depth knowledge of IT architectural principles and methodologies.

For those architects and organizations already familiar with the TOGAF framework, we have added Appendix 2 which describes how one can apply the BIAN standard with the TOGAF framework.

1.2 HOW TO USE THIS BOOK

This book will provide you with in-depth knowledge to help you understand the full construct of BIAN artifacts, how to apply them and how you can contribute to help the BIAN standard fulfill your (organization’s) needs. We will start with a short introduction to the BIAN organization, its goals, the deliverables and the future state.

Due to the constant development and evaluation of the BIAN models, additions to this publication will be publicly available at the BIAN homepage (www.bian.org).

This initial chapter gives you a high-level overview of all the topics that we will discuss in more detail in the designated chapters that follow:

- Chapter 2: BIAN's primary purpose and approach;
- Chapter 3: Understanding the theory;
- Chapter 4: The BIAN Service Landscape;
- Chapter 5: How to apply the BIAN standard;
- Chapter 6: Assembling a representative enterprise blueprint;
- Chapter 7: An enterprise blueprint is a framework for analysis.
1.3 BIAN, THE BANKING INDUSTRY ARCHITECTURE NETWORK

The Banking Industry Architecture Network (BIAN) is a global, not-for-profit association of banks, solution providers, consultancy companies, integrators and academic partners with the shared aim of defining a semantic standard for the banking industry covering almost all the well-known architectural layers.

The BIAN was formed in 2008 by a group of banks and solution providers with the shared aim of defining a semantic Service Operation standard for the financial services industry. At a later stage, other standards bodies, like ISO and IFX, joined along with some academic partners.

BIAN’s expectation is that a standard definition of business functions and service interactions that describe the general construct of any bank will be of significant benefit to the industry. When compared to an increasing number of proprietary designs, a dedicated industry standard, like BIAN, provides the following main benefits:

- It enables the more efficient and effective development and integration of software solutions for and between banks;
- It significantly lowers the overall integration costs;
- It improves the operational efficiency within and between banks and provides the opportunity for greater solution and capability re-use within and among banks;
- It supports the current need for more industry integration and collaboration through the usage of (open) APIs;
- It supports the adoption of more flexible business service sourcing models and enhances the evolution and adoption of shared third-party business services;
- It supports FinTechs and RegTechs to gain an easy insight in the complex financial services industry structure.

BIAN refers to the collection of designs that makes up its industry standard known as the BIAN Service Landscape. The BIAN Service Landscape’s development is iterative, relying on the active contribution of industry participants to build consensus and encourage adoption. The BIAN Association coordinates the evolution of the BIAN Service Landscape on behalf of its members with regular new version releases and seeks feedback to help continually expand and refine its content.

It is helpful to understand that BIAN Working Groups govern Service Domains. Each Service Definition Working Group covers an associated area of business expertise. The scope covered by individual Working Groups is defined in their charter so that, collectively, Working Groups cover the whole landscape with no overlaps between them.

---

1 This book refers to banking, but all examples and models are applicable for other sectors in the Financial Services Industry.
The governance of Service Domains within a business area is assigned to a Working Group. The Working Group is then responsible for the initial specification and any subsequent updates to its assigned collection of Service Domains. This implies the content creation is driven by the BIAN members using their experts’ knowledge and experience.

### 1.4 THE BIAN SERVICE LANDSCAPE, AN OVERVIEW

The BIAN Architecture is a layered/componentized one. These layers and components are identified in figure 1.

![Components of the BIAN Service Landscape](image)

This set of architectural artifacts is defined as the BIAN Service Landscape, it includes:
- The BIAN Meta Model, based on the ISO 20022 Meta Model;
- The BIAN Business Vocabulary;
- The high-level BIAN reference map: the BIAN Service Landscape;
- The BIAN Business Architecture;
- The BIAN Business Capability Model;
- The BIAN Service Domain Definitions;
- The BIAN Service Operations Definitions;
- The BIAN Business Scenario Definitions;
- The BIAN Application Architecture;
- The BIAN Application Capabilities (also called: Vendor Agnostic IT Model);
- The BIAN API/Message Definitions;
- The BIAN Information Architecture;
- The BIAN Business Object Model, fully aligned with ISO 20022;
- The BIAN API Classification Guideline.
The BIAN standard is published in a UML repository, as well as an HTML read-only version which is freely available on the BIAN website (https://www.bian.org/). In addition, a collection of supporting documents is maintained and released with each revised release of the BIAN standard.

The following options are in place to collect and process your feedback:

- BIAN members are encouraged to provide feedback by using the BIAN Wiki, to the Architectural Committee, Architecture Framework & Foundation Working Group or via their representatives.
- Non-members are invited to post their suggestions by using the BIAN website www.bian.org.
- Feedback can also be posted to how-to.guide@bian.org.
BIAN’s primary purpose and approach

2.1 INTRODUCTION

Since 2008 the financial services industry has faced a series of challenges in respect to their business models, customer relations and information technology. The desired business changes in banks are often slowed down by an inflexible and complex systems landscape. The primary reason for the difficult transformation and modernization of that landscape is the fact that the components are tightly coupled.

The BIAN Association strives to enhance the flexibility and agility of financial services systems by improving the integration with an architecture that is based on services. Those financial services-specific semantic services are the cornerstone upon which to achieve this flexibility. The value of BIAN is the standardization of those functional services based on a well drafted architecture framework with elements carefully chosen from industry best practices. It is the ambition of the BIAN Association to achieve a consensus on the service definition among leading banks and providers in the financial services industry, which in due time should lead to standardized services.

The goal of the BIAN Association is to develop the most important content, concepts and methods in interoperability, supporting the aim of lower integration costs in the financial services industry and to facilitate business innovation and agility by:

■ Providing an architecture framework with all of the necessary elements, tools and methodologies for a sustainable operational model through the adoption of and alignment to available market standards.
■ Focusing on the definition of semantic services and/or API-definitions to improve the semantic integration of the financial services landscapes.
■ Enabling the financial services industry to develop and run successfully a loosely coupled environment.
■ Acceptance by the members of the BIAN Association and the industry of the way that the requirements will be implemented by both financial institutions and solution suppliers, resulting in the defined services becoming the de-facto-standard in the financial services industry.
2.2 A DIFFERENT APPROACH TO A WELL-ESTABLISHED PROBLEM

Many financial services industry participants, including the founding members of the BIAN Association, have frequently observed a common and enduring problem: excessive complexity in most application portfolios. This complexity results in inflexible/unresponsive systems, inflated enhancement, increasing maintenance and operational costs, and an inability to leverage rapidly evolving advanced solutions, technologies, approaches and business models.

The BIAN Association was set up to address this issue by developing a common industry standard to define functional partitions and Service Operations that could be used inside any financial organization resulting in the anticipated benefits already noted. However, the objective of the BIAN Association raises a key question: “Why should the BIAN model and approach be successful in addressing application portfolio and interoperability complexity?”.

2.2.1 BIAN’s capability view versus a traditional process view

At the core of the proposition of the BIAN Association is the adoption of a capability-oriented approach to architecting the systems that support the financial organization. This approach is fundamentally different from the prevailing ‘process-centric’ designs. To highlight this critical difference, a comparison can be made with architectural disciplines when applied to the highly tangible problem of designing the layout of a city as opposed to the much less tangible design of a commercial enterprise such as a financial institution, see figure 2.

Figure 2: Comparing enterprise and city planning

Copyright protected. Use is for Single Users only via a VHP Approved License.
For information and printed versions please see www.vanharen.net
Any design is a combination of the ingredients that are used and the behaviors that the design is intended to support. The ingredients relate to static or persistent things that are ‘deployed’ and the behaviors refer to more dynamic patterns of desired responses to anticipated events or triggers. An architect develops an overall design based on an understanding as to how the ingredients need to be configured to support the intended behaviors. In the case of the town planner this is a town plan. The ingredients seen in the town plan are the buildings, parks and communications infrastructure that need to be in place to support the anticipated behaviors of the town’s inhabitants. These behaviors could be traced as journeys or ‘days in a life’ on the town plan.

Comparing building architecture as practiced by the town planner and enterprise architecture that might eventually be used to design the applications for a bank reveals an important shortfall in the arsenal of tools for business architects.

The ingredients that make up the bank are not tangible things like buildings and roads, they are the far less tangible business capabilities that a bank must establish in order to execute business. The behaviors that are modeled as journeys through the town are the business processes that the bank supports. Enterprise/business architects have extensive experience in modeling processes. The key issue for the business architect is defining the generic capability building blocks that they should select and configure to create the equivalent of the town plan for the bank. These capabilities can, in different combinations and sequences, then support those more familiar processes.

The result of building without a governing town plan is a shanty-town – buildings and roads are put up as and when they are needed and, over time, chaos is inevitable. Without a town plan for the business, systems built to meet the immediate needs of the processes as they are today will eventually lead to the same inevitable chaos in terms of overlapping and redundant applications, as shown in figure 3.

![A city where new construction is not coordinated with a town plan](image1.png)

![An enterprise where application development is not coordinated with an enterprise plan](image2.png)

Figure 3: Building without a plan – shanty town and application portfolio

The problem of application complexity goes much further than the obvious problem of redundancy in the overlapping applications. It is greatly exacerbated when the