



# Volatility in the IT Landscape



**Bert Hooyman**  
Vice President & Chief Architect

# Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Some Real-life Scenarios</b>	<b>3</b>
<b>3. Analysis</b>	<b>3</b>
Information versus Processes	4
Volatility: Differentiated Speed of Change	4
Exceptions to the Rule	5
<b>4. Divide &amp; Conquer</b>	<b>5</b>
Processing in the Stable Tier	6
Data in the Stable Tier	6
Processes in the Fluid Tier	6
Data in the Fluid Tier	7
The Volatile Tier	7
<b>5. Key Stakeholders in the Tiered Volatility Model</b>	<b>8</b>
Stakeholders in the Volatile Tier	8
Stakeholders in the Fluid Tier	8
Stakeholders in the Stable Tier	8
<b>6. Food for Thought</b>	<b>9</b>
Amazon ( <i>amazon.com</i> ) – Very Fluid and Utterly Volatile	9
Transactions Without Side-effects	9
Requirements Engineering Across Volatility Tiers	9
Cross-Cutting Concerns in Application Design	10
From Request/Reply to Publish/Subscribe	10
<b>7. Conclusions</b>	<b>10</b>
<b>8. Acknowledgments</b>	<b>10</b>

# 1. Introduction

Across all industries, in companies large and small, IT is faced today with a need for cost reduction, combined with ever-rising expectations from the 'business' in terms of time to market, flexibility and innovation. In many cases, there is an additional push for technology refresh, for example to cover for the retirement of the first generation IT staff.

This is not an easy problem to solve. The goals are conflicting. The result is that the IT organization is perceived as incapable, too static, and self-centered. This paper introduces the notion of *volatility*, or the susceptibility and willingness to change, as a means to characterize the IT landscape and adapt the IT strategy accordingly.

## 2. Some Real-life Scenarios

A large semi-government organization, initially set up in the seventies, has a paper-based process flow with many business units that are not well integrated. The IT landscape is predominantly based on mainframe COBOL with CICS transactions. The organization suffers from poor customer service, an ageing workforce and a significant lead time for the implementation of any (legal) process change. The IT organization aims to refresh the technology, become customer-centric in its predominant mode of operation and wishes to introduce agility (i.e. reducing the time to market for any new or changed offering).

A top-tier bank fails to distinguish between the strict accounting principles that drive the core back-office functions and the more loosely defined principles of the business development organization. As a result, IT development practices and principles all align with the most stringent of restrictions. Business Development does not understand why it would take the bank so long to introduce a simple new mortgage product. Software standardization precludes the use of modern (model-driven) tools for the business functions that are so urgently needed.

A mid-size insurance company in the Personal and Commercial space, working exclusively through third-party intermediaries, suffers from having to run its insurance systems by the same rules as the "big guys", draining their IT resources. This leaves the business end of the landscape, where the insurer deals with its agents, exposed. The company seeks to re-architect the IT landscape. Flexibility is the main focus point; service orientation the predominant solution direction.

A large government department is charged with the introduction of a completely new way of realizing social security for the citizens. Given the political nature of the change, there is a push for a very rapid development cycle. Agile methodologies are called for. Over thirty different data sources are impacted by this change program, and at least eleven distinct corporate systems.

## 3. Analysis

In all of the examples given above, one of the main problems that each organization is facing is the sheer size of the plan – it goes across all tiers of the IT landscape and touches many of the organizational units (business units). This puts tremendous pressure on the IT organization.

The main theme to apply here is known as 'divide and conquer'. Split the problem into a number of smaller problems, and deal with those in isolation. The question is how can we best divide the problem, given the complexity and inter-relationships of all the components and subsystems?

From past experience, two main strategies seem to arise, both capable of enforcing a proper divide and hence supporting the much-needed conquest. When both strategies are combined, a blueprint for a change strategy is relatively easy to formulate. From there, all enterprise architectures will be different based on the relative priorities, timelines, and budgets of the enterprise.

## Information versus Processes

The first strategy is based on the notion that Algorithms + Data Structures = Programs<sup>1</sup>. Put another way, software systems are comprised of data sets and process implementations. Of these, data sets have a tendency to change much less drastically when compared with process change. Data sets, and the accompanying data models or domain models form an extremely solid basis for an IT landscape, even when all else changes. Processes are more volatile than the data they operate on.

It is important to distinguish information from processes because enterprise application integration is, to a large extent, an exercise in data manipulation, not so much process integration. So with a well-defined and understood information strategy, EAI will become much more straightforward (think of the financial benefits of reducing the cost of EAI).

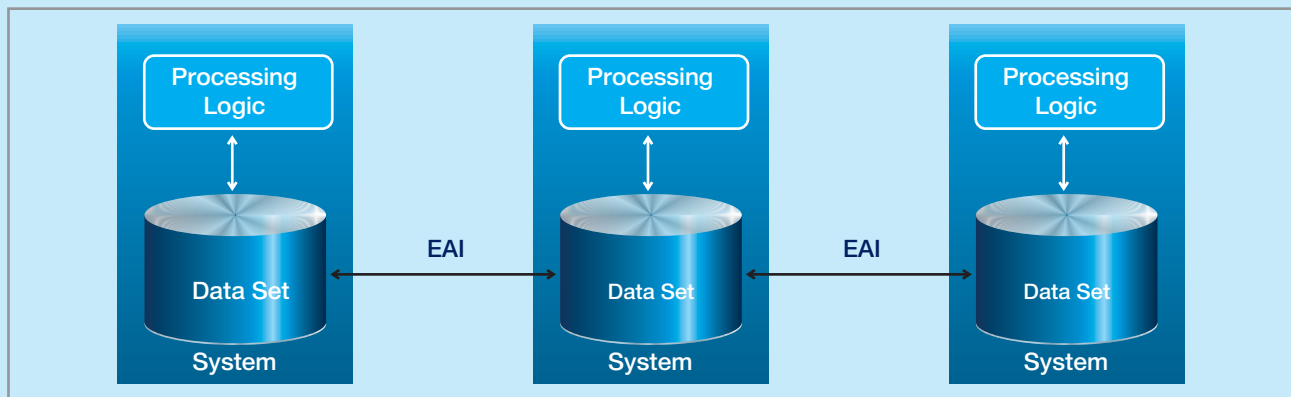


Figure 1: Considering Information Separate from Processing

## Volatility: Differentiated Speed of Change

The second strategy is based on the notion that change occurs at different rates throughout the IT landscape. There is typically a tier with very little change at the 'core' of the organization. This is the 'stable' tier. In this tier will be the systems that support the key business transactions of the organization. Examples of these are insurance claims handling systems, payment systems in banks, commercial lending administrations, manufacturing resource planning, operations support systems in telecommunications, etc. In this tier, the core data sets of the organization are kept.

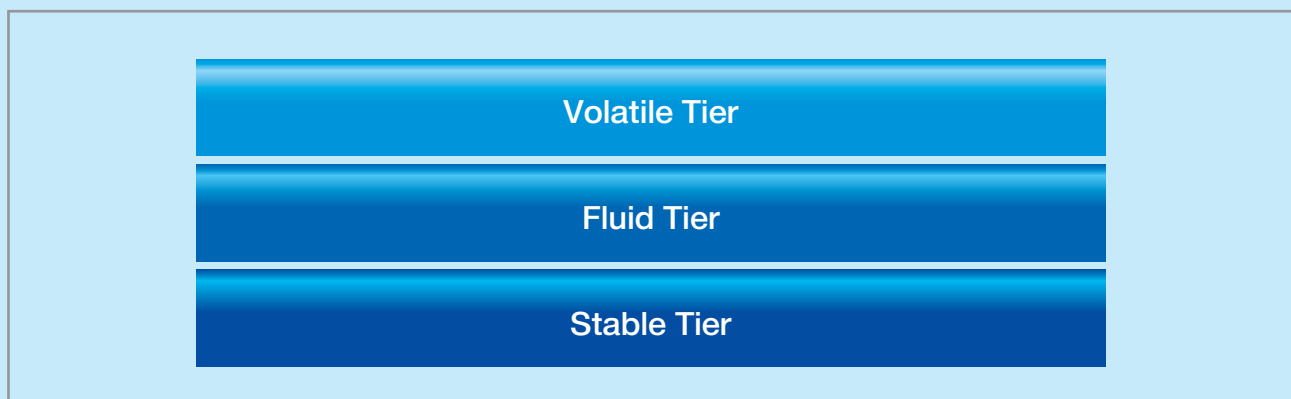


Figure 2: A Three-tier Volatility Model

It is often useful to consider additional, more supportive, processes that may take place in the stable tier. Examples of these are planning, print, billing, staffing, and finance & accounting. These processes are supportive in that they are necessary for the enterprise yet do not play a distinguishing role in the competitive landscape. For this reason, they are best placed in the stable tier.

There is a second tier, the 'fluid' tier, with more volatility, where business-oriented processes are realized. Commercial quotations, advise, new product introductions, and orchestrated business processes across multiple lines of business are all examples of this. In this tier, there is most definitely an element of data management. The core transactions, however, are delegated to the stable tier.

In the third volatility tier, no relevant data is being held. Here, the enterprise exposes itself to the outside world. This is where branding and positioning happen. This is where customer intimacy and partner focus find their home. This tier, known as the 'volatile' tier, realizes only a minimal amount of processing logic, barely sufficient to sustain the user experience. All other processing is delegated to the fluid tier.

To further elaborate the responsibilities of the fluid and the volatile tier, it is perhaps easiest to think of business activities in terms of “what” and “how”. *What* is offered to customers and partners of the enterprise, in terms of products, services and associated processes, is managed in the fluid tier. *How* those services are rendered is determined in the volatile tier. The “how” may still require some flyweight form of processing on the side of the enterprise, but the extent of such processing (and associated state information) is limited to a single interaction session. Any more long-term interaction state is managed in the fluid tier as it pertains to the “what” of the service offering.

## Exceptions to the Rule

It is important to understand that the suggested layering based on volatility assumes that the enterprise is involved in some transaction-oriented business model. Banks, insurance companies, logistics firms, pharmaceutical companies, retailers and manufacturers of consumer goods all fall under this broad category. But there are corner cases. Information vendors, for example, have only the thinnest possible fluid tier as they thrive on selling raw data without much ado. Search engine providers also fall in this category. On the other hand, there are companies that have hardly any stable tier at all. These are typically companies that operate as 'middle man' between a buy side and a sell side.

# 4. Divide & Conquer

We now have a basis for a strategy of divide and conquer; the diagram below illustrates how the IT landscape is organized in six distinct focus areas, each with its own specific focus and responsibility.

	Processing	Data
Volatile	None	Minimal
Fluid	Business Processes	Supportive
Stable	Transactional and supportive	Transactional

Figure 3: Divide & Conquer Framework

We can now classify these six focus areas, and define sensible IT strategies against each.

## Processing in the Stable Tier

For the core transaction processing, the most significant concern that the IT organization may have is the need for modernization of legacy applications. Legacy replacement may be driven by cost (“we’re running a mainframe system for this and it’s very expensive”), by quality (“over time we have added pieces of functionality left and right; today nobody knows how the system works. It’s become very brittle”), or by human factors (“we’re finding it difficult to find qualified REXX developers with sufficient domain knowledge even to keep the lights on, let alone implement system changes”).

Other issues arise in the context of mergers & acquisitions, when supplementary product administrations are introduced.

A good strategy in this domain is to isolate the actual systems by wrapping them in service wrappers. This is the well-known façade pattern; once a façade is in place the IT organization is free to modernize the legacy infrastructure without disruption to the rest of the organization. Legacy modernization is not without its complexities, though. Often, legacy systems are fully self-contained solutions that combine elements of the stable, the fluid, and even the volatile tier. This phenomenon is known as “transactions with side effects”. Yes, the system does capture a business transaction in a consistent and correct way, but as a side effect it also triggers some business workflows, it generates some reports or it creates some notifications to end users. These are not the responsibilities of the stable tier, ideally. So, legacy transformation involves de-activating those side effects as well as service-enabling the core transactions. In preparing for a legacy transformation project, understanding the strengths of the various tiers and separating system functions accordingly is an indispensable first step. More thoughts on legacy transformation can be found in “Transactions Without Side-effects” on page 9 of this paper.

There is a second class of processes in the stable tier; that are the supportive processes (sometimes identified as ‘shared services’ across multiple lines of business). In many cases, these services should be cost-optimized since their value contribution to the enterprise is low by definition. Depending on the appetite for cost reduction, a simple approach could be to move towards a shared service model if not already in place. This would remove redundant systems from the landscape. A more aggressive next step would be to outsource the operation of the supportive processes, ultimately leading to a pay-per-use model, possibly realized in the cloud.

## Data in the Stable Tier

The information that is managed in the stable tier should be considered as the most fundamental component of the IT landscape. Here, correctness, completeness, and consistency reign. Agility, and flexibility in general, are less important provided that the domain model is generic enough to support current as well as future transactional models. Investments in this tier should focus on cross-enterprise consistency and unification of definitions. Once established, this domain should not be exposed to ongoing changes.

## Processes in the Fluid Tier

Processes in this tier are predominantly business processes, i.e. they realize the user stories (or use cases) of real-life business scenarios. To support the desired levels of agility and flexibility that business stakeholders expect from IT, process definitions in the fluid tier are not hard-coded but instead driven by (externalized) process descriptions. In many cases, Business Process Management (BPM) suites are used to create the desired levels of flexibility.

A typical example of an effective use of a fluid tier is when an international enterprise wishes to consolidate all of its transactional processing in a single set of back-office systems. Inevitably, there will be regional differences in how the business operates, and there will be local regulatory compliance requirements that force the enterprise to differentiate its process implementations across borders. Here, a BPM solution in the fluid tier will be used to realize the regional process variants, calling upon the transactional capabilities of the shared services in the stable tier.

A second class of processing in the fluid tier is everything related to Customer Relationship Management (CRM). CRM processes do not result in business transactions in the classical sense, i.e. the information produced by a CRM system is not the type of data one would consider for the stable tier. Instead, the CRM data is directly actionable in the fluid tier itself, to drive customer interactions in the volatile tier.

Related to CRM is the domain of Business Intelligence. Though typically considered as an element of the stable tier, business intelligence really belongs to the fluid tier as it is not part of the core transactional basis of the enterprise. The output of business intelligence drives business development. And modern BI is based not only on transactional data in the stable tier but also on customer interaction data (originating in the volatile tier and captured in the fluid tier). All of this is especially true in the context of real-time customer analytics.

To satisfy the needs of internal reporting (Management Information or MI), we believe the same levels of flexibility are needed as those that Business Development wants for the main business features and functions. Hence, MI belongs to the fluid tier. In fact all reporting should be organized in the fluid tier, to the extent that the underlying data is readily available.

## Data in the Fluid Tier

There will be a significant amount of information in the fluid tier, necessary to support dynamic (BPM) processes. Most of this information never reaches the stable tier as it has a more temporary nature and does not reflect the transactional business that is processed in the stable tier. As an example, a mortgage administration system is only interested in mortgage contracts, not in commercial propositions and alternative payment schedules that are discussed between a bank and its customers. Hence, these proposals and the financial data on which they are based on reside in the fluid tier only.

The fluid tier is also a good place to capture customer interaction data such as contact history, website visits and clickstreams, use of teller machines, kiosks and other services. This data is the basis for customer profiling and drives marketing campaigns. When a CRM solution is in use, the data it uses as well as the information it produces belong to the fluid tier.

There is a third class of data sets that, in many cases, is good to have in the fluid tier. This data set is best understood as 'a replica of all transactional data in the stable tier that contributes to the 360° view on the customer'. The driver for this is that in the volatile tier, where customer interaction happens, there is an ongoing need for up-to-date status information from the stable tier (e.g. current account balance, fifteen most recent transactions). And realistically, the systems in the stable tier are not well equipped to provide such data in a timely manner. From the perspective of the transactional systems, these data queries come as *random data requests*, which is what can bring many good legacy systems to their knees. From our experience, we have also observed a need to have the transactional data available irrespective of whether the back-office systems are down because of scheduled maintenance. Internet business runs 24x7 and does not typically observe batch window outages. Hence, having a replica of customer-relevant transactional data available in the fluid tier brings great benefits to the customer-facing side of the enterprise. As a side effect, processes in the fluid tier that need data from the stable tier can make use of this replicated data, too.

With a relevant set of transactional information present in the fluid tier, it makes sense to consider to what extent the fluid tier can service all BI, MI, and operational reporting needs of the enterprise.

## The Volatile Tier

As is obvious from figure 3, the volatile tier is essentially empty. Yes, there will be some logic coding in this tier, and yes, there will be some data captured and held in this tier, but none of it is relevant to the company<sup>2</sup>. The purpose of logic and data in the volatile tier is to provide customer intimacy and loyalty, to enhance the user experience and to adapt the presence of the company to the channel and the location of interaction.

<sup>2</sup>The data and the logic may be irrelevant, but the presence of the volatile tier is essential.

In short, the volatile tier is icing on the cake. For this reason, it is often dismissed by core Information Technology management. We believe this is a mistake, because this icing is the most visible piece of Information Technology in the whole company. Thus, the volatile tier is an opportunity for IT to shine. Having said so, it is important to realize that all successes in the volatile tier are based on a capable fluid tier, which in turn is dependent upon a solid transactional basis in the stable tier.

## 5. Key Stakeholders in the Tiered Volatility Model

We started this paper by looking at the problems of the IT organization as it is faced with shrinking budgets, aging legacy, and increasing expectations from the business. With a Divide & Conquer framework in place, it is now time to identify key stakeholders in every area of the framework.

For this, we revert to the simple tiered model of figure 2. The stakeholders are concerned about data and processes combined; hence there is no reason to consider data and processes in isolation.

### Stakeholders in the Volatile Tier

As indicated before, the volatile tier is the icing on the cake and it is the most visible part of the IT landscape. The key stakeholders for the volatile tier are within the marketing organization. This includes product marketing, channel management, corporate branding and any regional marketing capacity. As a general rule, it's fair to state that if an organization does not have anybody working in any of these responsibilities, there will not be any success in the volatile tier. Strategy and vision in the volatile tier is a must-have.

The central themes in the volatile tier today<sup>3</sup> evolve around mobile tools, social media, customer intimacy, loyalty, and satisfaction. The focus is on “how” to attract new business from new customers, “how” to leverage cross-sell and up-sell opportunities, etc. Marketing (and hence the volatile tier) is not about “what” is being sold. That is for business development to define.

### Stakeholders in the Fluid Tier

The fluid tier is where the real innovation takes place. Significant IT investments may be needed to realize the desired levels of flexibility and adaptability. Key themes in this tier include customer relationship management, new product introduction, customer analytics, orchestrated business processes, regional process differentiation, as well as management information. Key stakeholders not only come from the business development organization, product management and line of business strategy, but also from business analysis and reporting.

This tier defines “what” is being offered by the enterprise. IT decisions in this tier define what is possible in terms of time to market, product configurations/compositions, and white labeling.

### Stakeholders in the Stable Tier

The stable tier is home to the IT staff itself and this is where the daily operations of the core back-end systems are run. Depending on the organizational model of the enterprise, there will be line-of-business silos or domains, each with their own experts around the core systems. Main themes in this tier include availability & disaster recovery, auditing, security, and correct implementation of accounting principles. It may come as a surprise that most of these concerns are non-functional.

In the situation of an aging IT staff, that staff itself is also a stakeholder in this tier as most of these older employees will have jobs around the core transaction-processing systems of the organization.

## 6. Food for Thought

With the volatile tiers model in place, it is always interesting to consider some extreme cases and interpret those in the context of volatile tiers.

### Amazon (*amazon.com*) – Very Fluid and Utterly Volatile

Amazon (*amazon.com*) is an interesting enterprise for many reasons. A lot of case studies, reports, and books have been devoted to the subject already. When looking at Amazon from a volatility point of view, a few things quickly become apparent. First, their stable tier is a very straightforward transactional retail model. It is a reliable backbone for doing business on the Internet. In the fluid tier, Amazon is able to innovate. New product introductions are “business as usual”, sales models change and adapt at a high rate. The concept of being a market-maker for second-hand books, I am sure, is fully realized using 'fluid' solutions.<sup>4</sup>

In the volatile tier, Amazon has always been a leader. They thrive on customer intimacy and loyalty. Interestingly, Amazon also learned that you can run an eCommerce operation *without any volatile tier at all*. Through their service API's and toolkits, Amazon allows third parties to build custom user interfaces, effectively reducing the cost of the volatility tier and extending their reach at the same time. This concept, more formally known as 'white labeling' is an interesting corner case of the volatility model.

### Transactions without Side-Effects

We indicated earlier that the responsibility of the stable tier is to execute the key business transactions of the organization. Many legacy systems combine this core transaction processing with additional system responsibilities. When those systems were designed, this was seen as a benefit. “Integrated solutions” was a positive qualifier. Today that may no longer be the case. “Integrated” suggests monolithic. It implies a system that spans multiple volatility tiers. “Integrated” is now a negative qualifier.

How would a pure transaction-oriented core business administration system look? In some way, it is little more than a glorified database system, focusing on transactional integrity (ACID and all that), availability & disaster recovery, auditing, security, and correct implementation of accounting principles. There will always be some level of processing around the core data sets, for example daily interest calculations in a banking system, or accruals in general ledgers. These are examples of processes that are part of the stable tier. But the primary responsibility of the core business systems is transaction processing – without side effects.

Some of the newer commercial-off-the-shelf offerings for core banking and insurance indicate that this is indeed a trend: these products offer a transactional service API which is truly free of side effects. So where “integrated” was once the positive qualifier to look for, “service-oriented” or “componentized” is what organizations should look for today. Note that “service-oriented” is not the same as “service-enabled”. Service enabling means that a legacy system is not changed, it is only wrapped in a service tier. The side effects may still exist, inside the wrapper. Service-oriented means (or suggests) that the system is decomposed, that is capabilities are accessible directly, without side effects.

### Requirements Engineering Across Volatility Tiers

When an IT landscape is organized along volatility tiers, the requirements specification team must recognize and respect the boundaries of the tiers. Specifically, in the fluid tier, business process requirements must aim to base their requirements on existing (transactional) services, and especially they must align with enterprise data models established in the stable tier. Similarly, a user experience in the volatile tier must be designed with the available business service catalog in mind – it is fine to identify new requirements that are not met by existing business services, but it is essential that the requirements engineering team is aware of what is present in the fluid tier before starting the requirements analysis.

<sup>4</sup> Note: I'm guessing here, I'm not aware of how Amazon is actually doing these things.

Another key concern is that the requirements engineering team should refrain from including features and functions that are not part of the tier they are engineering for. In other words, when a stable feature is missing, it is fine to flag the feature as missing from the stable tier, but it would be an error to include that feature in the fluid tier just because it is not part of the stable tier yet.

## Cross-Cutting Concerns in Application Design

In the software engineering domain, aspect-oriented programming aims to address cross-cutting concerns of software applications.<sup>5</sup> A similar notion exists in the larger space of application portfolios, or, in our context, in the fluid tier of our volatility model. Across all of the business processes realized in the fluid tier, there is an extensive range of features and functions that cut right across all those processes. Typical examples include data validation, persistence, globalization which includes localization, information security, caching, logging, monitoring, and business rules. Many if not all of these aspects can be realized as *shared services in the fluid tier*. This leads to a further decomposition of business processes. As a matter of fact, some of these aspects such as business rules are typically isolated already in modern BPM packages.

## From Request / Reply to Publish / Subscribe

Beyond componentized transactional systems, a current trend in the IT industry is to consider event-driven architectures (EDA) for the purpose of decoupling IT systems. Event-driven architectures combine quite nicely with the volatility tiers, as event-driven solutions in the fluid tier can be triggered by the completion of transactions in the stable tier.<sup>6</sup>

Where a traditional service-oriented implementation of a transactional system uses a request/reply model (R/R) for integration, an EDA approach uses publish/subscribe semantics (P/S) to further decouple IT systems. How this works is that transactional systems in the stable tier issue notifications of relevant business events (“a new insurance policy has been established”) without knowing what other systems in the IT landscape are interested in this information. Interested systems, on their part, subscribe to these notifications in order to trigger relevant business processes. Such business process will, more often than not, be realized in the fluid tier.

# 7. Conclusions

A tiered framework based on the speed of change or volatility helps identify key concerns and stakeholders, which are different for every tier. Depending on the pain points being addressed, such as cost optimization, legacy transformation, or customer satisfaction, the relevant tier with its stakeholders can be identified. IT strategies and decisions are made based on the typical characteristics of every tier. This divide & conquer approach helps IT management meet the expectations that their organizations place on them.

# 8. Acknowledgments

The author wishes to thank Suresh Nair, MphasiS Chief Architect for Financial Services, and Frans Galema, MphasiS Associate VP Business Development for the Insurance Market, for their valuable feedback and comments.

<sup>5</sup> See [http://en.wikipedia.org/wiki/Cross-cutting\\_concern](http://en.wikipedia.org/wiki/Cross-cutting_concern) for a gentle introduction on aspect-oriented programming and cross-cutting concerns

<sup>6</sup> For a more in-depth treatment of this subject, see “Event Processing – Designing IT Systems for Agile Companies”, K. Mani Chandy and W. Roy Schulte, McGraw Hill 2010, ISBN 978-0-07-163350-5

# About the Author



**Bert Hooyman**

**Vice President & Chief Architect, Europe Mphasis**

Bert joined Mphasis in 2001. He has worked with numerous clients across Europe. Before joining Mphasis, Bert has worked as a consultant for a software vendor specializing in business process monitoring. Earlier, he worked for a UNIX computer system vendor in an international presales consulting role. Bert is a senior IT architect specializing in business process management, event-driven solutions and data management applications. He is the principal architect for Mphasis in Europe.

# ABOUT MPHASIS.

Mphasis is a \$1 billion global service provider, delivering technology based solutions to clients across the world. With over 41,000 people, Mphasis services clients in Banking and Capital Markets, Insurance, Manufacturing, Communications, Media & Entertainment, Healthcare & Life Sciences, Transportation & Logistics, Retail & Consumer Packaged Goods, Energy & Utilities, and Governments around the world. Our competency lies in our ability to offer integrated service offerings in Applications, Infrastructure Services, and Business Process Outsourcing. To know more about Mphasis, log on to [www.mphasis.com](http://www.mphasis.com)

For more information, contact: [sales@mphasis.com](mailto:sales@mphasis.com).

USA: 460 Park Avenue South, Suite #1101, New York, NY 10016, USA  
Tel.: +1 212 686 6655, Fax: +1 212 686 2422

UK: 88 Wood Street, London EC2V 7RS, UK  
Tel.: +44 20 85281000, Fax: +44 20 85281001

AUSTRALIA: 9 Norberry Terrace, 177-199 Pacific Hwy, North Sydney, 2060, Australia  
Tel.: +61 2 99542222, Fax: +61 2 99558112

INDIA: Bagmane Technology Park, Byrasandra Village, C.V. Raman Nagar, Bangalore 560 093, India  
Tel.: +91 80 4004 0404, Fax: +91 80 4004 9999



1111