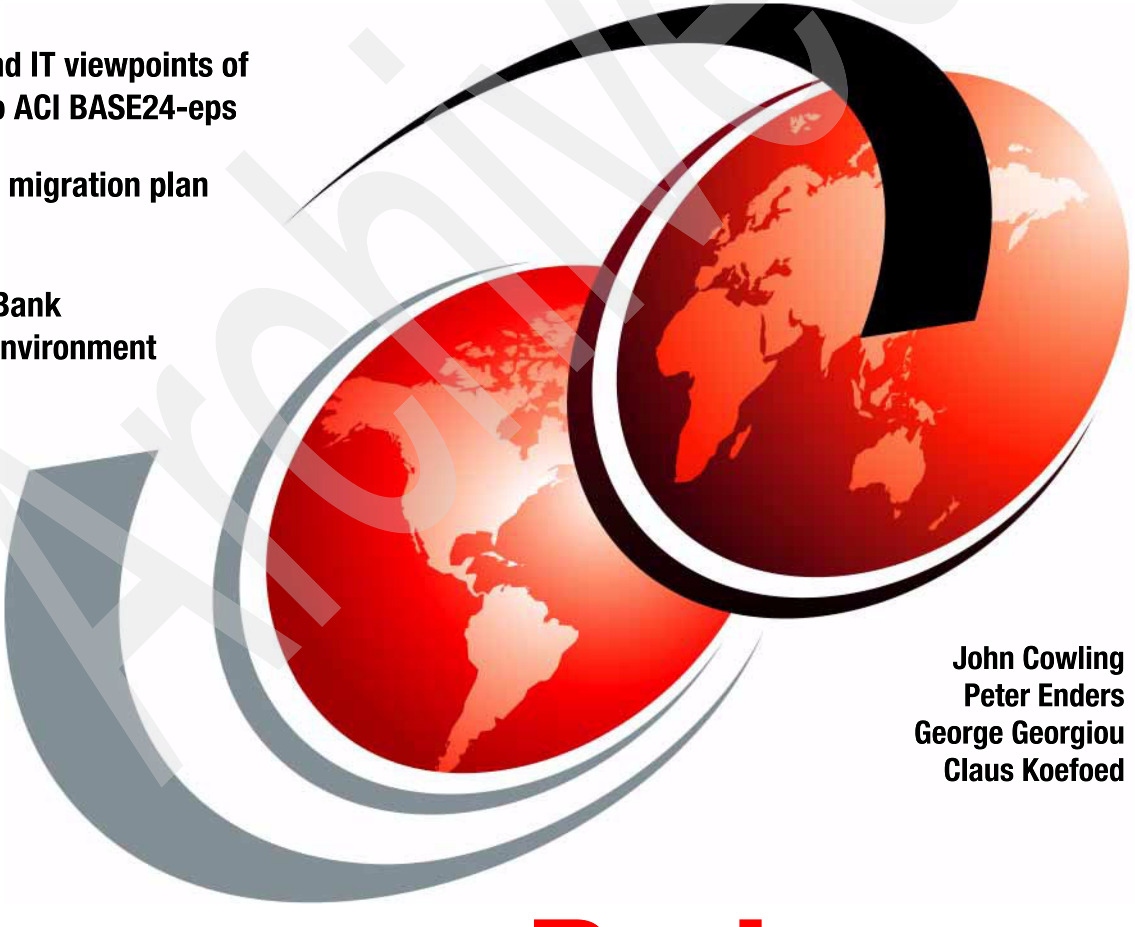
Migration of Payments System to ACI Worldwide BASE24-eps

Business and IT viewpoints of migration to ACI BASE24-eps

Designing a migration plan

IBM Smart Bank showcase environment



John Cowling
Peter Enders
George Georgiou
Claus Koefoed

Redpaper

ibm.com/redbooks

International Technical Support Organization

Migration of Payments System to ACI Worldwide BASE24-eps

August 2007

REDP-4337-00

Note: Before using this information and the product it supports, read the information in "Notices" on page vii.

First Edition (August 2007)

This edition applies to ACI Worldwide's BASE24-eps product.

Contents

Notices . . . vii
Trademarks . . . viii

Preface . . . ix
The team that wrote this paper . . . ix
Become a published author . . . x
Comments welcome . . . xi

Chapter 1. BASE24-eps migration overview . . . 1
1.1 Why migrate at all? . . . 2
1.1.1 What is a migration? . . . 2
1.1.2 Some business aspects of migrations . . . 3
1.1.3 The migration process - an iterative approach . . . 6
1.1.4 Efforts and costs - some orders of magnitude . . . 10
1.2 BASE24-eps application areas or system types . . . 15

Chapter 2. Introducing BASE24-eps . . . 17
2.1 BASE24-eps . . . 18
2.1.1 Consumer transaction support . . . 18
2.1.2 Transaction switching and routing . . . 18
2.1.3 Flexible authorization . . . 19
2.1.4 Integrated consumer data . . . 20
2.1.5 Traditional and emerging delivery channels . . . 20
2.1.6 Reliable security infrastructure . . . 20
2.1.7 Intuitive graphical user interface . . . 21
2.1.8 Scriptable extracts and reports . . . 21
2.1.9 National switch and international card scheme interfaces . . . 21

2.2 Architecture 22
2.3 Operability 23
2.4 Scripting component 23
2.5 The user interface 24
2.6 Rich functionality 25
2.7 Platform independence 25
2.8 Flexible architecture 25
2.9 The ACI Worldwide Payments Framework 26

Chapter 3. Defining a migration plan 27
3.1 Business motivation for change 28
3.2 Current environment - pre-assessment 29
3.3 Determining standard and custom functionality 30
3.4 Determining functionality for BASE24-eps scripting 30
3.5 How complex is a migration? 31
3.6 Summary 31

Chapter 4. Designing a migration plan 33
4.1 Migration principles 34
4.1.1 Maintain the attributes of the system during migration 34
4.1.2 Transparent to cardholders 35
4.1.3 Transparent to merchants 35
4.1.4 Transparent to Card Associations 35
4.1.5 Transparent to Financial Institutions 35
4.1.6 Transparent to branch staff 35
4.1.7 Minimize disruption to system users 36
4.1.8 Realize new business benefits 36
4.1.9 Minimize (dis)investment in temporary transition software 36
4.1.10 Migration phases 36
4.2 The migration landscape 36
4.2.1 Issuing functionality 37
4.2.2 Acquiring functionality 38
4.2.3 Extracts 38
4.2.4 Communications 39
4.2.5 User interface migration 39
4.2.6 Data(base) migration 39
4.2.7 Addressing data conversion risk 40
4.2.8 Archival 41
4.3 The reference strategy for migration 41
4.3.1 The conceptual model 43
4.3.2 Step 0: Mainframe authorization 45
4.3.3 Step 1: Implementation of the new switch 46
4.3.4 Step 2: Issuing functionality 48

4.3.5 Step 3: Acquiring functionality 49
4.3.6 Step 4: Completion 50
4.3.7 Alternatives for Steps 2 and 3 51
4.4 Summary 52

Chapter 5. The IBM Smart Bank showcase environment 55
5.1 BASE24-eps IBM Smart Bank environment 56
5.2 The Smart Bank 56
5.3 BASE24-eps Smart Bank capabilities 56
5.3.1 Smart Bank architecture 57

Chapter 6. Sample of a migration to ACI Worldwide BASE24-eps based on the IBM Smart Bank environment 61
6.1 Migration plan 62
6.1.1 Phase I 62
6.1.2 Phase II 63
6.1.3 Phase III 63

Chapter 7. Closing perspectives 65

Related publications 69
IBM Redbooks 69
Online resources 69
How to get Redbooks 69
Help from IBM 69

Index 71

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

z/OS®
IBM®
Rational®
Redbooks®
Redbooks (logo) ®
System p™
System z™
Tivoli Enterprise™
Tivoli®

The following terms are trademarks of other companies:

ACI, ACI Worldwide, BASE24-es and BASE24-eps are trademarks or registered trademarks of ACI Worldwide Inc. or its affiliates in the United States, other countries, or both.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Java, JavaScript, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Paragon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM Redpaper discusses the migration of a customer's existing payment environment to a new solution, based on ACI Worldwide's BASE24-eps product (formerly known as BASE24-es), running on an IBM® System z™ infrastructure.

This document covers the following topics:

- General considerations on migrations, their necessity, and financial implications
- An introduction to the ACI Worldwide BASE24-eps product
- Items to be migrated and scope of the work
- Designing a migration plan
- IBM Smart Bank showcase environment running in Montpellier, France
- An example of a migration to ACI Worldwide BASE24-eps based on the IBM Smart Bank environment

The choice of an infrastructure based on the IBM System z platform in this paper is primarily for reasons of document scope, and because this platform is frequently found as an integration hub in customer installations. It will be equally possible to perform the portrayed migrations to other infrastructures, based on (but not limited to) platforms, such as UltraSPARC from SUN, System p™ from IBM, Integrity from HP, and Integrity NonStop, also from HP. The reasoning, procedures, and techniques used in conjunction with these platforms will be very similar, although there might be some more significant differences in those areas which are more closely linked to the implemented technology, such as database migrations.

The team that wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

John Cowling is an IBM senior IT Specialist with 25 years of experience in software development, often in real-time, multivendor environments. Over the last 15 years John has been engaged in a variety of projects with strict requirements for high availability, such as debit and credit card online payment processing, contact centers and telephone banking. John is based in Amsterdam, The Netherlands.

Peter Enders is an IBM Certified Sales Specialist and is working as a Consulting Analyst in the field of large systems competitive sales support. He has been active for over 30 years with divers mainframe and second tier systems, especially IBM System z and Sun, FSC BS2000 and HP NonStop. He has held many technical and management positions, such as Second Level Technical Support, Country Marketing Manager, European Sales Manager, and he is based in Frankfurt, Germany.

George Georgiou is an ACI Worldwide Solution Architect with 25 years of experience in software development, marketing and sales, primarily in the wholesale and retail banking environments. He has been at ACI Worldwide for 15 years and has worked on BASE24 and BASE24-es projects. These projects have been throughout the EMEA territory. George has a master's degree in Computer Science from the University of London.

Claus Koefoed is a Program Manager for System z from the CompeteCenter in Copenhagen, IBM Denmark. He has 30 years of experience in the IT industry with IBM. His area of professional focus is IBM mainframe competition, EFT solutions. Claus is editor of a newsletter, Sinet, about mainframe subjects for the IBM community. He holds a master's degree of Economics from the University of Copenhagen.

Thanks to the following people for their contributions to this project:

Alex Louwe-Kooijmans
International Technical Support Organization, Poughkeepsie Center

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

BASE24-eps migration overview

This chapter looks, in a general way, at the whole theme of migrations, their implications, and some common misconceptions associated with them. As it is difficult to give a definitive and generic statement on migrations and their implications, due to the individuality of each situation, it might be useful to consider the theme from the point of view of an industry consultant. This point of view discusses business and IT aspects and comes to some generalized conclusions about effort and associated cost.

1.1 Why migrate at all?

This is normally the first question that is posed to explain the environment of a migration, its benefits and its risks. Migrations are often, if not normally, regarded as something with a slightly negative connotation, as much from the financial as from the risk viewpoint. The normal idiom is the well-known phrase “never touch a running system”. While migrations may seem unwieldy and costly, they are vital and necessary. In the end it is a question of operations more than of technology; and operations, or better operational qualities, are closely associated with business value and business risk. Let us look at this more closely.

1.1.1 What is a migration?

This question, seemingly unsurpassed in its triviality, is to my mind one of the most important questions to answer. And also one of the most difficult to answer exhaustively. Anything else than a complete status quo - a frozen state if you like - has aspects of a migration, and hence should be considered under this rubric. Upgrading a release of an operating system or a middleware system may be just as difficult as migrating a complete application system, and may have just as many repercussions. Also, the risks involved may be similar if not greater, as changes to such system components can affect the operation of the complete software stack running on the processor or cluster of processors. Migrations then are changes to the components in the software stack which make the software more current, more productive, or which reduce the business risks inherent in its operation.

The cadence of migrations can often be defined by the cadence of manufacturer’s changes in the software, which usually go hand in hand with improvements in functionality and, after some time, with the withdrawal of maintenance and other services for the old releases. Another current possibility for cadence is a change in the infrastructure topology, which will usually suggest a change in the balance of decentralized systems to centralized systems for a given production process. Although this rebalancing tends to come in cycles, today we seem to be in a prolonged period of server consolidation (“scon”) or IT infrastructure simplification (“iis”) if you like.

The most elusive cadence will be the one applying to the applications themselves, where a life cycle management (or PLM, product life cycle management) suggests that after a certain period an application, although still fully functional and in full production, might no longer be the optimal solution to a specific production problem. The reasoning for this is normally that the functional richness, integration, and qualities of service available at the time when the application was conceived stand in no relation to what is now possible, so that a continued use of the old application will deprive the business of advantages otherwise inherent in the newer software, and thus limit the business value of the IT support for this application as a whole.

1.1.2 Some business aspects of migrations

Having spent some words on migrations in general and why to do them, it will be necessary to spend some time considering the implications for the business. In general, it is nice to divide such business considerations into two sections - the reduction of business risk, and the understanding of the rudiments of migration processes so as to be able to appreciate the steps necessary to come to a first financially sound decision. Both themes could fill a book in themselves - therefore we will only touch on each insofar as it may be necessary for a comprehension of the rest of this paper as a whole.

We want to portray the question of business risk and its reduction in IT terms by means of the following graphic.

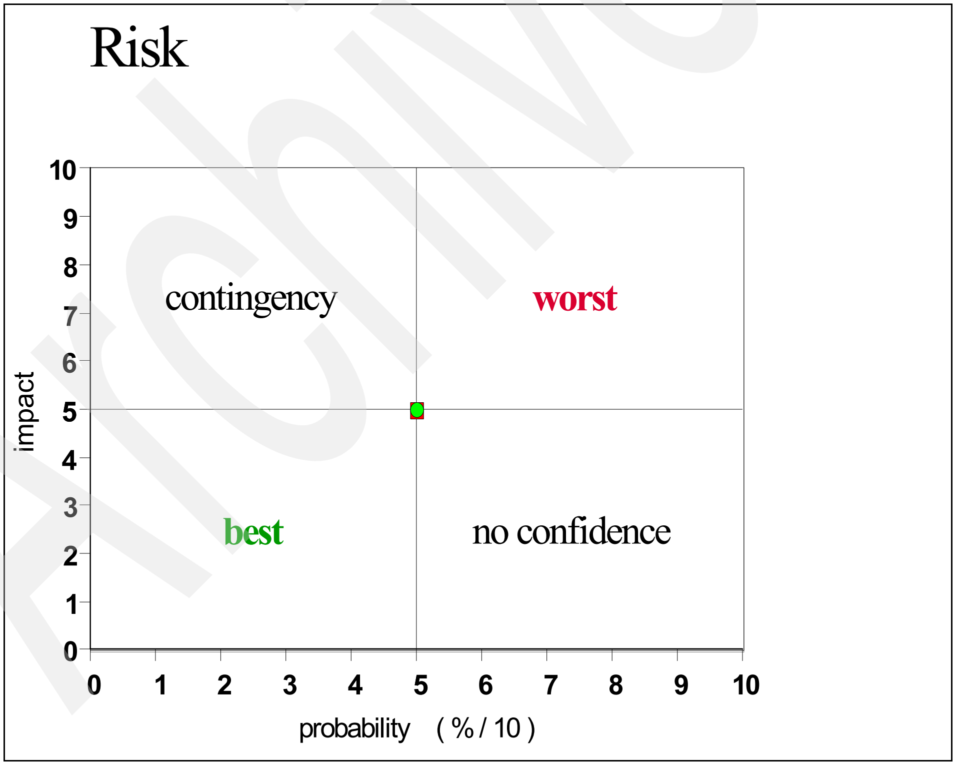


Figure 1-1 Risk

This says in essence that the business risk situation incurred in an IT environment is composed of the probability of an incident occurring and the damage caused by the incident when it does occur. The statistical damage caused in a time period is then the product of these two components in the given time period. The components, *impact* and *probability*, in the graphic are a function of the IT environment for the business solution, meaning the platform and software stack, the application itself, and how the whole is configured, operated, and maintained.

Looking at this in slightly more detail, the top right-hand quadrant depicts those environments which are forever breaking down, and which cause the maximum of damage when they do so. Clearly this is the worst situation imaginable. In the lower right-hand quadrant, the frequency of the outages is still very high, but the damage caused per incident is relatively low. In this situation the confidence of the users in the IT environment will usually be lacking. In the top left-hand quadrant, the environment is such that outages rarely occur, but when they do their impact is extremely high. In this case it will be necessary to have effective contingency measures in place, which will usually be a combination of take-over and disaster recovery scenarios. Finally, the lower left-hand quadrant depicts those environments which have already been substantially optimized, and which have reduced the business risk for their organizations.

Clearly it should be the objective to position oneself in the lower left-hand quadrant. This can be achieved by two complementary means - looking at the IT infrastructure and looking at the IT integrity. And, of course, implementing the indicated improvements in each area.

The following graphic may help to display this.

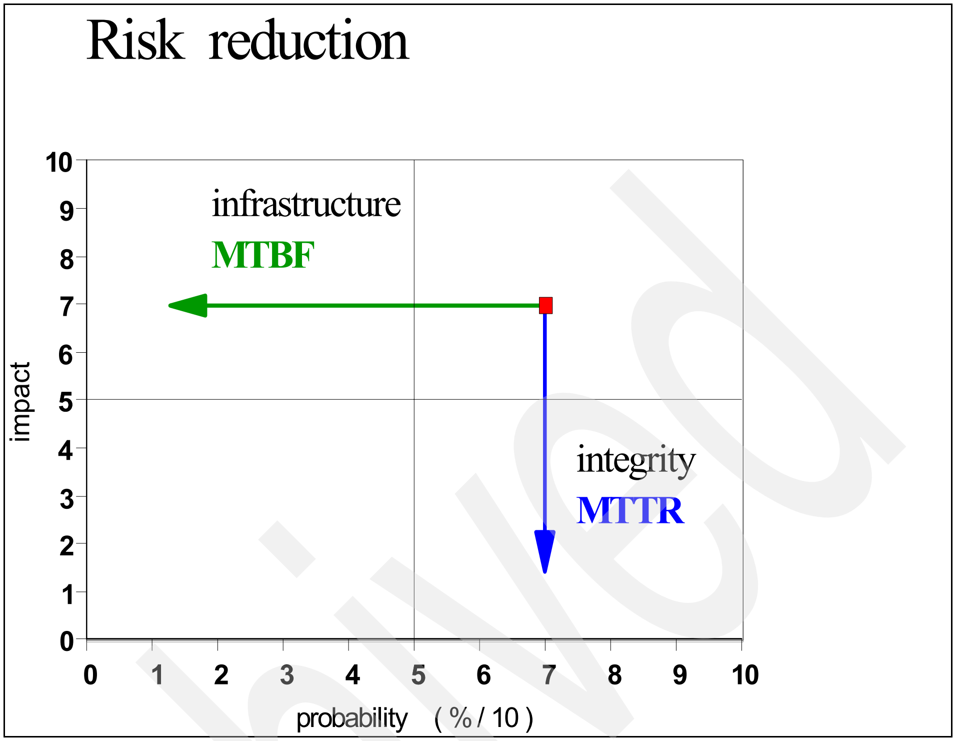


Figure 1-2 Risk reduction

Improving the infrastructure, by which it means here mainly the platform and its associated proximate software, will lower the failure incidence, or probability of failure, and hence increase the time between failures. IT specialists like to talk of this as the MTBF - the mean time between failures. So platform choice and software updating will improve this aspect of the equation. The work to achieve this is, in effect, that of a migration effort. Improving the integrity of the IT environment, for example things like cluster configurations, failover, recovery, automation, backup, and the like, will help to reduce the impact of a possible incident if and when it does occur. Again, the IT specialists like to talk of this as the MTTR - the mean time to repair. Put simply, in the rare cases where it does go wrong, if indeed these do still exist, the downtime will be so low as to be tolerable, even if not completely acceptable. And again, the work to achieve this will be part of a migration effort.

So improving the infrastructure and integrity aspects of the IT environment will lead to a reduction of business risk, which in turn may have positive effects on other business criteria such as rating, and the like.

1.1.3 The migration process - an iterative approach

Having stated that the business aspects of improving IT tend to suggest that migrating to such improved environments makes sound business sense, it is then necessary to look at the migration process, or the flow of work, to discover if we can discern characteristics which in broad terms can describe the procedure leading to a financially sound and profitable project. Essentially, as a very basic concept, migrations are based on knowledge of the IT environment, and partially of the business processes surrounding it. The effect of the business processes will mostly be seen in the operations of the systems rather than in the technology implementations themselves. Here we will concentrate only on the IT environment.

Gleaning knowledge about an IT environment is a recursive process - there will be several iterations where aspects are investigated in more and more detail until a level of knowledge is reached which is deemed sufficient to execute the task in hand, which here will normally be a full proposal for the migration. Each iteration may be accompanied by different IT specialists, from the customer as well as from the supplier side, to look at distinct aspects which may cause additional overhead or even impediments in the migration itself.

Normally we would structure this process into a five-step scheme. However, for the purposes of this short summary, the process can be simplified and depicted generically in four steps, or phases, which we will call *discover*, *analyses*, *design*, and *build*.

The “discover” phase is essentially a data gathering - from the IT departments as well as from the application designers and end-users.

The “analyses” phase tries to assign a meaning to the discovered facts as they pertain to a future solution concept.

The “design” phase then fixes the system architecture and design, incorporating new and current infrastructures and components.

Finally, the “build” phase is the consolidation of all of this work into a priced proposal which can serve as a basis for a migration work contract.

These phases and their recursion can be shown graphically as follows:

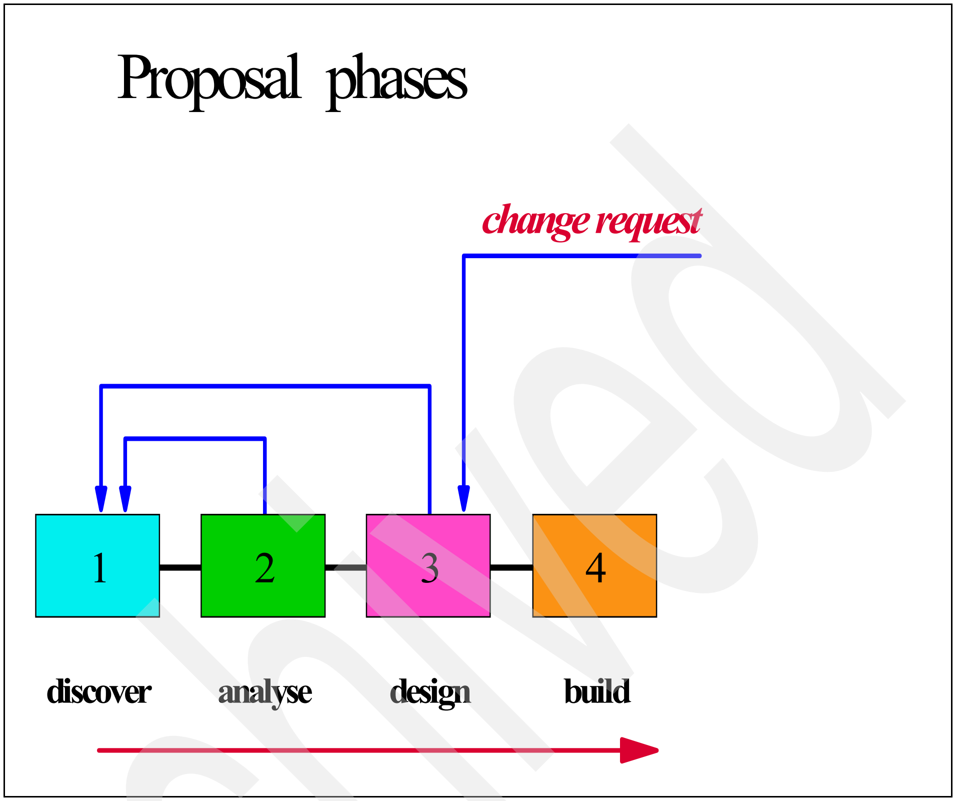


Figure 1-3 Proposal phase

Clearly, this process of coming to a priced proposal is a great deal of work, for the IT supplier as well as for the customer, as the analysis phase cannot be completed without intensive participation of customer personnel. Also, this phase has to take place during ongoing operations, partly referencing the people responsible for this delicate task. In the end, there is always a distinct possibility that the migration solution as envisaged will be suboptimal, and that a concrete realization will only be possible by changing technology and processes in a more far reaching way in the company as a whole. In order to avoid this risk, we tend to propose dividing this proposal process into several subordinate parts or cycles, each of which is self contained, has its own characteristics and produces its individual result.

A diagrammatic representation follows.

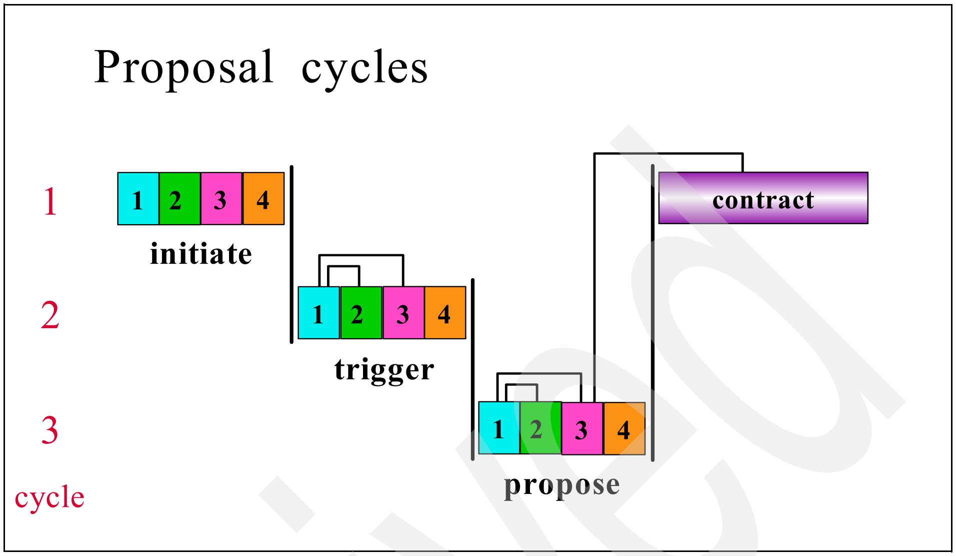


Figure 1-4 Proposal cycles

In this process model, the *initiate* or first cycle is the one in which a decision of principle is reached to embark on the investigation of the feasibility of a migration project. If this sounds somewhat hazy, it should be remembered that even relatively cursory investigations require financing, so that this initiate phase is, in effect, a decision for expenditure, which in turn needs some sort of plausible statement on the expected return on investment (ROI), if indeed this is possible. Often at this stage it is really no more than a statement of intent. The basic idea of this first cycle is to keep it as short and therefore as inexpensive as possible. The level of knowledge of the infrastructure, the applications and the operations necessary for this cycle is one of summary knowledge only.

The next cycle, termed *trigger*, is the one in which the principle directions are defined which a possible future migration will take. This cycle is in effect the first major iteration in the process leading to a supplier proposal. The purpose of this cycle is to reach a point where the decision can be taken to issue a full request for proposal, with the corresponding efforts involved for the customer IT department and the suppliers. By dividing the proposal phase into two cycles, of which this is the first part, we are, in effect, reducing the business risk of committed financial and personnel resources, and increasing the probability of a successful outcome. The level of knowledge required here will be noticeably superior to that of the “initiate” cycle, but the effort required to attain this will be quite tolerable in comparison with the analyses required for the full migration proposal.

This second cycle can also be termed a pre-assessment stage, as its outcome will be the technical and financial pre-assessment required to decide upon the complete migration. The rest of this paper treats the issues and themes associated with such a pre-assessment. The final cycle is then logically the “propose” cycle, in which the full proposal is constructed, usually by the principal supplier. This cycle builds on the output and decisions of the “trigger” or pre-assessment cycle, and carries them forward with finer grain and in greater depth. This last cycle clearly requires the maximum of knowledge and detail about the complete IT environment, and it will be the stage at which the maximum effort and hence pre-construction cost will be incurred.

It should be noted that none of these project cycles are inevitable, in the sense that they have to be executed irrespective of the outcome. A project flow such as this three-cycle scheme will have multiple breakpoints, or exit points, along the way. This means that at several stages in the process towards contracting a migration, and especially at the term of the cycles mentioned here, it may be opportune to admit that the best solution will be not to proceed with the migration as envisaged. The existence of such exit points is in effect an assurance for the customer, as it formalizes the possibility of ending a project before its term and thus optimizing the expense to results ratio. Even if such a migration project is exited at some stage before its term, this does not automatically mean that the investments made up to that point are wasted. It should be noted that the IT and process information gleaned in such exercises are of immense value, as they often do not exist (up to that point) in such a formalized way. This information can often be used very profitably for infrastructure optimization and the like.

In graphic form the possibility of exiting the project could be presented as follows:

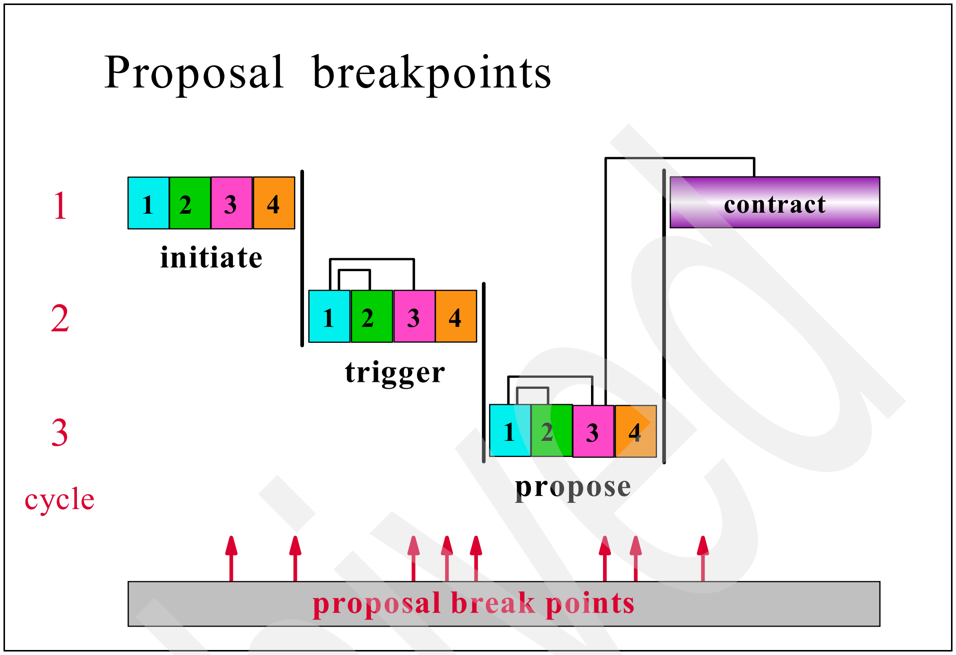


Figure 1-5 Proposal breakpoints

As a point of interest, there is shown here a feedback link to the “design” phase of the “propose” cycle, which is that of the change management (also shown in the “proposal phases” graphic above). In essence this says that at several points in the last cycle a change request may be generated which will noticeably affect the effort involved in this cycle. Also, the number of change requests will increase exponentially as the elapse time of the project progresses, which is another reason for the split proposal cycles. However, this complex is really a subject to be considered in a more detailed project management view, and need not be treated here.

1.1.4 Efforts and costs - some orders of magnitude

It is extremely difficult to specify the effort, and hence the cost, of a migration in such generic terms that they can find application to most concrete cases. However, by restricting the scope to the three solution proposal cycles as outlined above, it will be possible to define effort and hence cost ranges of the pre-contract work to a measure that will at least allow a rough sizing of the expenses which will be incurred. It should be noted that these figures can only be average values, and that it is a characteristic of any average that it has a deviation or bandwidth. Where exactly a migration project lies in a possible bandwidth of effort will always be an object of conjecture and possibly also of dispute.

Generically, it can be said that the greatest effort in a migration proposal lies in the “discover” phase (Figure 1-3 on page 7). It is not uncommon to see this phase representing approximately 60% of the complete effort, or maybe slightly higher. This then shows the importance of a very close interlock between the customer and the supplier, so as to keep this phase highly efficient and the costs within bounds. This analysis is presuming an optimal project governance occasioned by high management attention.

The “analyses” and “design” phases will vary with the complexity of the environment, but their variation will be commensurate with that of the “discover” phase, so that their percentage of the total effort can be seen to be relatively constant.

The effort for the “build” phase will be slightly higher as a percentage for the smaller and simpler projects, as there are always a certain number of formalities which have to be gone through irrespective of the size of the project. This then would give a slight bias towards larger projects from an efficiency point of view.

Put graphically, the full effort for the complete proposal will typically divide as follows:

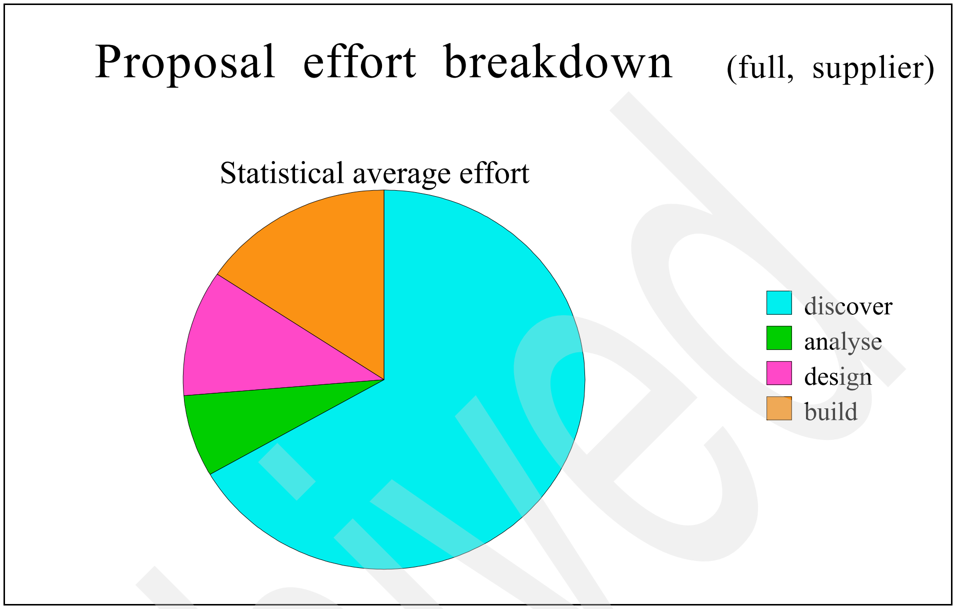


Figure 1-6 Proposal effort breakdown

As mentioned, it is not conceivable for a supplier to establish any kind of a migration proposal without the close cooperation of the customer. This then means that the customer personnel will have to expend effort (mainly invested time) to aid the supplier in the common endeavour. In general, and using all the caveats of averages as above, we find that the customer effort is around 75% of the supplier effort for producing the proposal. This will vary depending on the availability of prepared and succinct infrastructure information, but it is one of these generalized averages. Clearly, the breakdown of this effort will be different to that of the supplier - almost all of the customer effort will be in the “discover” phase. There will be some small involvement in the “analyses” phase, and maybe a little more (coordination effort) in the “design” phase, but the “build” phase will be the sole responsibility of the supplier.

Again, represented as a graphic, this looks somewhat as follows:

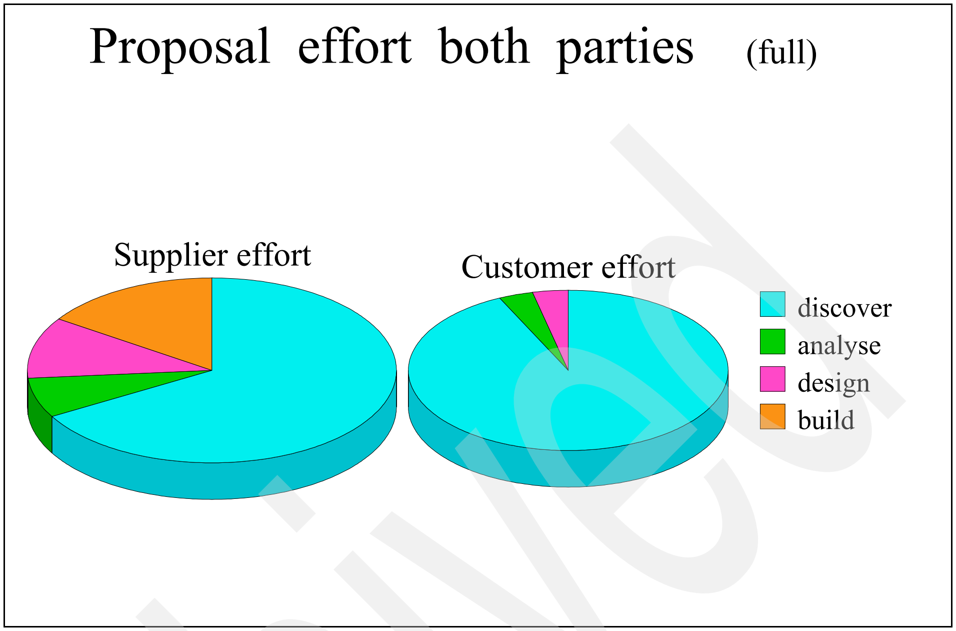


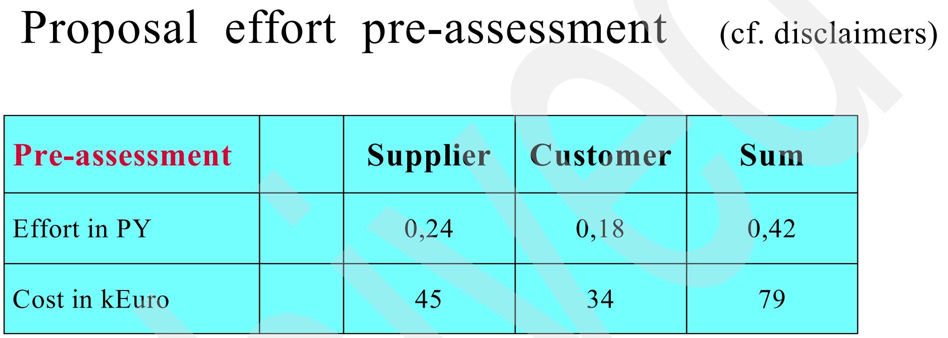
Figure 1-7 Proposal effort both parties

Given all of this, we then come to the primary (and sometimes only) question which interests customers (and often supplier) management - “what does it all cost?”.

As mentioned, we have up to now looked at migrations from a generic point of view. And, being generic, and with an inherently very large bandwidth, it would be quite presumptuous to try to allocate any kind of more specific effort or cost numbers to the migration proposal. So now we have to leave our generalities and define what we are doing, and with what. The following chapters of this paper treat the case of a specific financial payments solution, and its implementation on a specific platform. This narrows the effort and hence cost bandwidth considerably, as we are now talking about a packaged solution rather than a RYO (roll your own), defined and often standardized interfaces to other products and solution areas, and defined implementation criteria.

We can then further narrow down the bandwidth by applying the three-cycle proposal model as outlined above. For the purposes of the rest of this paper we are primarily interested in the “trigger” or pre-assessment cycle, as this will lead us to a more precise view of whether a migration to the new payments solution environment appears to make sound financial and business sense. The outcome of the pre-assessment, as mentioned, will be the recommendation to proceed (or not) to a full proposal, as well as some basic indices for the future solution which will already give a high confidence level about how the full final proposal will look. However, it should be reiterated that only a full proposal can give the concrete effort and cost parameters which, together with the ROI considerations, will confirm the business value of the new solution.

Considering all of this, and considering the myriads of conditions, caveats and disclaimers mentioned in this point of view, we would venture to suggest that the following table can be a summary of the cost and effort necessary for the "trigger" or pre-assessment phase.



Proposal effort pre-assessment (cf. disclaimers)

Pre-assessment		Supplier	Customer	Sum
Effort in PY		0,24	0,18	0,42
Cost in kEuro		45	34	79

Figure 1-8 Proposal effort pre-assessment

This table then gives the effort necessary for a pre-assessment of the new payments environment given the boundary conditions as explained. The effort is given in person-years (PY) and will normally be composed of several people with divers skills over shorter periods of time. The costs are given in thousands of euros (keuro), and are based on rates typically found in central Europe at the time of writing. The costs on the customer side are simply the incurred effort converted at the same rates as for the supplier. This might not be strictly true, and some might argue that such costs do not really occur at all (or at least not in this magnitude), but we have chosen to leave them in this representation to recall that for a pre-assessment an effort and hence costs are required on both sides, for the supplier and for the customer.

Considering these numbers, it should be evident that customers looking at the business impact and modernization of payment systems should request a pre-assessment of their solutions and IT infrastructure from their supplier to evaluate the need for a full proposal. The solution discussed in the rest of this paper is a means to achieve risk reduction and cost efficiency in this specific environment, and to my mind warrants a closer consideration. This short digression or point of view might have been of some use in the optimization of the incumbent payments IT infrastructure.

1.2 BASE24-eps application areas or system types

In the rest of this document, reference will usually be made to the ACI Worldwide product using the generic term “BASE24-eps”. However, it should be noted that the BASE24-eps product has a considerable scope of functionality, and can cover several distinct application areas.

These application areas, which can also be termed system types, can be enumerated as follows:

- Acquiring Systems - This is where BASE24-eps will manage devices such as POS and ATM Devices. Typical POS Acquiring Systems can range from thousands of devices to hundreds of thousands of devices. Typical ATM Acquiring Systems can range from tens of ATMs to tens of thousands of ATMs;
- Issuing Systems - This is where BASE24-eps will manage Cards and Positive Balance Files. Typical Issuing Systems can range from half a million cards to many millions of cards;
- Switching Systems - This is where BASE24-eps switches transactions between member banks. Typical Switching Systems can range from domestic in country switches to international card scheme switches;
- Acquiring/Issuing Systems - This is where BASE24-eps manages both Acquiring and Issuing. This is the most common configuration for BASE24-eps.

This rest of this document will provide guidelines about how to migrate all the above application areas to a BASE24-eps system.

2

Introducing BASE24-eps

At the heart of any payments network is a payment engine. The payment engine must acquire, switch, route and authorize transactions from a variety of sources. ACI Worldwide is a leader in mission-critical e-payment software, offering solutions to manage transactions from the point of access to settlement. BASE24-eps is part of the ACI Payments Framework, a broad suite of products developed by ACI Worldwide.

2.1 BASE24-eps

The ACI Worldwide BASE24-eps product represents the latest in payment engine technology, offering robust features and functions to the financial payments industry.

2.1.1 Consumer transaction support

BASE24-eps provides comprehensive support for consumer e-payment transactions initiated by a variety of transaction instruments that include credit, debit and chip cards.

As the payment industry evolves and new instruments emerge (for example, customer ID, mobile telephone numbers), the flexible nature of its architecture enables BASE24-eps to easily adapt to provide continued value.

Today, BASE24-eps supports a comprehensive cardholder and administrative transaction set that can be accessed from any appropriate delivery channel. Administrative transactions are also supported for settlement and reconciliation purposes.

BASE24-eps can be configured to maintain card and associated account information. Authorization logic can be scripted to perform a variety of tasks including check current status of the card, compare cardholder use against limit profiles, determine whether the transaction is allowed based on a number of configurable options, and more. In addition, BASE24-eps can provide alternate routing or stand-in authorization if a configured primary external authorizer becomes unavailable.

2.1.2 Transaction switching and routing

BASE24-eps provides a highly flexible routing structure for transactions. This flexibility not only routes transactions to the appropriate network, card association, processor or internal system for authorization, but it also helps users gain lower interchange charges by factoring in the total path when determining the authorization destination.

Transactions are routed based on a combination of the following:

- Source Profile
- Destination Profile
- Transaction Type
- Account Type 1 (FROM account type)
- Account Type 2 (TO account type)

- Method of consumer authentication (PIN present, magnetic stripe, and so on)

This flexibility in transaction routing accommodates different account types that may reside on different systems, as well as different platforms. Users can customize their transaction processing at various stages in the transaction life cycle, including:

- Pre-screening before transactions are sent to an external authorizer
- Defining the processing steps for real time internal authorization
- Defining the processing steps for stand-in authorization
- Specifying the destination of advice messages following authorization
- Specifying how the database should be impacted during the post-authorization process

2.1.3 Flexible authorization

BASE24-eps supports consumer authentication and authorization processing via a powerful scripting engine. The scripting engine enables organizations to control and define application logic without having to modify product source code.

Using an interpreted scripting language with JavaScript-like syntax, BASE24-eps allows users to create a variety of authorization scripts to tailor the authorization logic to meet specific business requirements or service agreements. Organizations can decide what data is used in the authorization process and when the data is used, regardless of whether the data is part of the transaction or from an alternative source.

ACI Worldwide provides a set of sample scripts that cover the basic positive, negative or usage-based authorization processes. This flexibility shortens the time needed to develop new products and services or accommodate changes requested by the business department of an organization. Separating the business logic (in scripts) from the product source code also facilitates script compatibility with future releases.

Users can choose what level of authorization should be performed on BASE24-eps. This can range from full authorization using a card, limit, account and balance information, to handling pre-screening before using a host for authorization, to just providing a stand-in capability using negative card data. All limits are user-defined to provide full flexibility in controlling use of the cards and accounts.

2.1.4 Integrated consumer data

The component architecture of BASE24-eps is designed for flexibility to leverage consumer data resident in external systems and databases. Users can develop components that expose consumer data to the scripting engine to allow more intuitive authorization functionality. This consumer data can be held in customer relationship management (CRM) systems, fraud management systems, customer information files, core banking systems and other applications.

By exposing more consumer information to the authorization process, organizations can improve consumer relationships by approving transactions based on more comprehensive consumer information. They can also intelligently manage risk by denying transactions based on certain risk factors.

2.1.5 Traditional and emerging delivery channels

In addition to support for traditional delivery channels, including ATM and point-of-sale (POS), BASE24-eps can process payments initiated through Internet shopping networks, personal digital assistants (PDAs), mobile telephones, Web ATMs, and home banking and branch systems. BASE24-eps offers a powerful, flexible foundation for delivering common services across multiple consumer access channels, computer systems and databases.

Through the use of XML- and ISO-based interface standards and other industry-specific formats, transaction services can be exposed to any channel. Thus, BASE24-eps can provide a single point of access across an enterprise for the service of consumer payments, eliminating the costs of maintaining multiple service points.

2.1.6 Reliable security infrastructure

Organizations' transaction security requirements can vary greatly depending on the environment. An organization typically requires an integrated system of software, industry-standard hardware, and procedures to properly implement financial transaction security.

BASE24-eps is designed to be flexible in its transaction security support and to provide a range of hardware options. The application addresses the diverse needs of large-scale transaction processing systems where the originator of a transaction may operate under an entirely different transaction security scheme than the authorizer. Regardless of the origin or destination of payments, BASE24-eps meets the current industry requirements for security, including Triple DES and EMV support.

BASE24-eps operates in a network environment where sensitive data, such as the personal identification number (PIN), is secured via encryption, and the system provides cryptographic functions such as PIN encryption, PIN verification, message authentication, chip authentication, and card verification using interfaces to external hardware security modules (HSMs).

2.1.7 Intuitive graphical user interface

The ACI user interface presents a task-oriented view of the application for multiple users ranging from business to technical to administrative, and it incorporates graphical elements such as hyperlinks, buttons, and pull-down menus. Users can also choose to display text labels in their local language to accommodate adaptation into their environment. Integrated help at the window level and the field level minimizes the need for extensive user training, while streamlining business processes and providing greater flexibility.

Written in Java™ and C++ and using XML message formats, the ACI user interface provides a flexible operating environment that is used by multiple ACI applications. A security administrator configures access permissions through the ACI user interface where a user security and user audit environment are shared by all ACI applications.

2.1.8 Scriptable extracts and reports

The BASE24-eps scripting engine gives users added flexibility and control over the reporting and extracting of financial transaction data. In an area where customization is virtually required to meet integration needs, the scripting feature allows real time definition of essential and ad hoc reports, as well as user-defined file layouts to serve as input to existing batch processes or reporting tools.

2.1.9 National switch and international card scheme interfaces

BASE24-eps incorporates off-the-shelf support for a range of international card scheme interfaces, including Visa, MasterCard, and American Express, as well as many national switch interfaces. These interfaces are built using a framework methodology covering ISO 8583 (1987), ISO 8583 (1993), and XML standards. This methodology makes use of *inheritance* to facilitate reusability of components and allows new interfaces to be built quickly by either ACI or the customer organization.

2.2 Architecture

ACI uses an object-oriented design and development style to implement its Enterprise Services architecture of BASE24-eps. This architecture helps reduce impacts associated with extending the core product code. The use of object-oriented programming languages, such as C++ and Java, enhance the extensibility of BASE24-eps solutions and minimize time-to-market for new products and services. By extending integration flexibility, BASE24-eps allows access to more customer information.

BASE24-eps software components use this architecture to create flexible business services that allow users to quickly develop and deploy new products and deliver enhanced customer service. The components are organized according to the function they perform to support processing for the required business services. Each business component performs a specific type of processing (that is, authorization or routing) or controls a specific part of the file system (for example, account or customer).

The architecture of BASE24-eps is designed for multiple platform configurations. The platform is defined as the hardware, operating system, middleware, and file system. However, platform-specific processing is isolated into specific components to allow the rest of the application to be common across all platforms.

BASE24-eps software components are organized according to the function they perform, as described here:

- Adaptors manage the information exchanged between the user and the business components. Adaptors can be designed to communicate with any acquiring or issuing entity, including Internet portals, hardware devices, service providers or interchanges (Visa, MasterCard, and so on), and in-house systems.
- Business components perform the processing required for the business services offered. Each business component performs a specific type of processing (that is, authorization, routing) or controls a specific part of the file system (for example, prefix, perusal).
- Foundation components provide information and processing required by business components regardless of the software module. These are the basic building blocks for any application. For example, the time component obtains the current system time in the format needed by the application. When any business component needs the system time, it obtains it from this foundation component.

- Platform-specific components insulate the application from changes in the operating system, middleware, and data structure. For example, processing performed by the time component differs across platforms.

2.3 Operability

BASE24-eps supports high volume, high availability, 24/7 operability through a scalable, high available software architecture that runs on a variety of platforms, as explained here:

- Flexible journal configuration and settlement cutover

  This allows for 24/7 cutover processing and uninterrupted processing across separate time frames.
- Implement new business logic without downtime

  Code and file system changes that affect configuration do not require a system restart.
- Hardware resilience

  The system and data access layers take advantage of each platform’s failover processing capabilities, all with the same set of application code.
- Consistent processing cost

  The asynchronous messaging model of BASE24-eps provides a consistent per transaction processing cost regardless of the transaction volume, which allows the application to grow as needed with a predictable hardware requirement.

2.4 Scripting component

The BASE24-eps scripting facility gives organizations a powerful JavaScript-like syntax to allow modification of application logic without having to modify the source code. The application uses these scripts to create journal perusal queries, define journal extract and reporting requirements, and as part of the authorization process.

Scripts are maintained and compiled through the user interface. Users can display a script repository that shows all scripts available for use by BASE24-eps. A script editor allows the user to add, edit, delete and compile scripts through the user interface. During the compile process, scripts are checked for syntax errors and saved on the BASE24-eps system. Rather than compiling into machine code, these scripts are ordered into a list of serial instructions that the script engine may interpret in real time during transaction processing.

Compiled scripts are loaded into memory where they can be retrieved for execution during transaction processing. If a change must be made to script logic, then the script can be updated, recompiled and placed back into use without ever taking the affected programs out of service.

Scripts have access to data from multiple sources; the primary source is from transaction data elements. Application files for authorization are also available to the script facility, and these include card file, limits file, usage accumulation file, account with balance file and pre-authorization file. For greater decision-making flexibility, access to additional customer information can be obtained through custom written components that “expose” the proprietary structure of a customer’s file to the script. These custom files may be core banking system files, ACI or other third-party card management systems, or a variety of other sources.

A single script can contain all of the tasks that BASE24-eps must perform to authorize a transaction. The tasks can also be split into multiple scripts that are organized in a hierarchical structure.

Because of the component-based design of the application and the scripting language, scripts implemented by the user may not need to be modified when a new release of the application becomes available, allowing users to easily upgrade to new releases of the product. If a user plans to take advantage of new functionality from within the script, then changes will be required.

2.5 The user interface

The BASE24-eps user interface employs Java, C++ and XML technologies, providing the user interfaces needed to manage all components of the application.

With the system’s built-in user security feature, users are assigned roles that grant them permission to specific functions and tasks associated with various windows. Users are authenticated during the logon process, thereby minimizing the risk associated with unauthorized users gaining access to functions they are not permitted to perform.

The user audit function is responsible for maintaining a secure audit database where all file maintenance transactions and modifications to the user security database are recorded. Before and after images of the affected record will be logged wherever appropriate.

The user interface design incorporates the flexibility for users to alter the layout and wording on the desktop to meet individual organization needs. All text and positional information is maintained in configuration files, so adapting the user interface without altering the product code is particularly easy. This structure also incorporates multi-language capability.

2.6 Rich functionality

BASE24-eps provides full functionality to support payment transactions across multiple channels. The software is parameter-driven, allowing users to configure a system that will meet their unique business requirements. ACI Worldwide’s product investment strategy accommodates periodic new releases of software providing support for both regulatory changes as well as new trends in electronic delivery.

2.7 Platform independence

BASE24-eps supports a broad range of computing environments, allowing customers to operate ACI Worldwide’s best-of-breed software on their choice of industry-standard platforms. BASE24-eps operates on a variety of HP, IBM and Sun servers. On each platform, the ACI software takes advantage of the best in systems software for reliability, availability and high-performance throughput.

2.8 Flexible architecture

Since the design of BASE24-eps includes support for scripting, there is little need for customer technical staff to have knowledge of the core languages of the application. A working knowledge of JavaScript™ programming methods will prepare an experienced programmer for maintenance and creation of the business logic necessary to meet the institution’s authorization processing needs.

Because the BASE24-eps application is component-based, ACI Worldwide customers have the freedom to develop components in-house, which extends product functionality. To accomplish this task, some basic skills concepts are required as outlined below:

- UML (Unified Modeling Language)
- C++
- Java

2.9 The ACI Worldwide Payments Framework

ACI Worldwide offers mission-critical e-payment software solutions to manage transactions from the point of access to settlement. BASE24-eps is part of the ACI Worldwide Payments Framework, a broad suite of products developed by ACI. The ACI Worldwide Payments Framework includes software to enable transaction processing through evolving Internet and wireless channels as well as traditional ATM and POS channels. ACI Worldwide solutions process transactions in real time and automate the back-office functions associated with settlement, dispute processing, fraud detection and account service.

3

Defining a migration plan

This chapter provides guidance about *what* is to be migrated. From a very high level this can appear to be quite simple, because a current environment exists and there is a desire to migrate to BASE24-eps.

How difficult can migration of a Payment Engine be? Many factors confound this question, as an organization’s systems are technically complex, mission critical, highly available, customer facing, and heavily embedded into a financial institution's information technology infrastructure. Therefore with a clear plan, the need for a pre-assessment phase during which the migration issues are quantified, the available approaches are scrutinized, and a migration scope and approach are agreed and understood by all involved parties.

Important factors in determining what to migrate are:

- Understanding the organization’s business motivations
- Understanding the organization’s current environment
- Determining the functionality to migrate

3.1 Business motivation for change

A truism regarding migrations is that all migrations are different. Reasons for this include:

- Every existing system is different - the combination of hardware platform, software environment, functionality and connectivity is likely to be unique to that solution;
- Every organization has different motivations for the migration - reasons could include obsolescence of hardware platform, obsolescence of software solution, consolidation following mergers and acquisitions, or support for new initiatives such as SEPA or EMV that the current solution cannot easily handle.

Therefore it is good practice to conduct a migration workshop where the business motivations for migration can be explored and example scenarios white boarded to reduce the number of possible migration options to a manageable number, which can later be analyzed at more depth. This would usually take the form of a 2-3 day business workshop that addresses the following:

1. The organizational objectives for the BASE24-eps migration.
2. The key players in the project and their roles and responsibilities. As well as the customer's stakeholders, this may identify, or even involve, third parties such as ATM/POS manufacturers, Card Producers, HSM vendors, System Integrators, and so forth.
3. A business overview of an outline solution, which in this particular case would be ACI Worldwide's BASE24-eps but could include other products in the ACI Worldwide Payments Framework.
4. A presentation of a BASE24-eps configuration that satisfies the customer's requirements.
5. Implementation Considerations such as 'compelling events' which the planning should take into consideration - for example, compliance deadlines, or the withdrawal of hardware or software support for the current environment on a particular date.
6. A presentation of the proposed migration approach: An outline strategy, a program/project structure, and project methodology.

In order that the Business Workshop is as beneficial as possible, the supplier team must be well prepared with a good understanding of the technical scope of the intended migration. Therefore, it will be necessary for Consultants to first assess the functionality which is to be replaced during the migration to BASE24-eps.

3.2 Current environment - pre-assessment

Designing a successful migration strategy requires that the customer's existing operational environment is thoroughly understood. Normally a number of interviews or workshops with customer subject matter experts would be held to determine the details of the incumbent IT solution.

The following list shows the major factors which have to be determined. In each case, it is also important to ascertain if the functionality should migrate “as is” or if change is to be introduced as part of the migration:

1. Volumes - This information is required for sizing the BASE24-eps solution on the appropriate hardware platform.
2. Communications - What communication protocols are currently in use, for example, HTTP, TCP/IP, X.25, SNA?
3. Transaction Flows - How do POS/ATM transactions flow through the system?
4. ATM Acquiring - Which types of ATM are in use, which protocols, which functionality?
5. POS Acquiring - Which types of POS terminal are in use, which protocols, which functionality?
6. Routing - How does the current environment route transactions, internally and externally? What are the routing rules or criteria for transactions presented to the system for authorization, and for *not-on-us* (as issuer) transactions that the system has acquired?
7. Interfaces - Which interfaces to the International Card Schemes, Domestic Card Schemes, and the Back End Host must be supported? Which protocols are used?
8. Authentication - How is PIN and Card Verification Authentication managed?
9. EMV - Does the system support Chip Cards?
10. Authorization - What are the business rules for the authorization of transactions? Are there Stand-In arrangements, and how are the stand-in processing communicated?
11. End of Period Processing - What is the financial end of day?
12. Operations - How is the system interfaced to an Enterprise Management System?
13. Has the Hardware environment for the new solution been determined, such as which operating system? Is there experience within the organization of running such configurations? Will retraining of the operational support organization be required? Are there standards for operational management with which the new solution should comply?

14. Contingency - How should the new solution handle Disaster Recovery?
15. Online Extracts - To which systems should the new solution interface? Are the interfaces well understood and accurately documented? Should information be presented in real time to a data warehouse?
16. Which departments will be using the ACI Desktop - the BASE24-eps User Interface? Will retraining be required? Will infrastructure upgrades be required at the desktop?

3.3 Determining standard and custom functionality

For the majority of BASE24-eps implementations there will be some functionality that can be replaced by standard (or slightly modified) BASE24-eps components, examples are:

1. External Interfaces - These modules are highly standardized as they must comply with mandates from the International Card Schemes, for example, VISA, MasterCard, American Express
2. Device Handlers - These can be POS Device Handlers such as SPDH, APACS, Hypercom, etc., or ATM Device Handlers such as NDC+, IFX, etc.

Any functionality that is not supported by modules of the off-the-shelf BASE24-eps package solution will have to be fully identified and defined at this stage. This can range from a small modification to an existing BASE24-eps component to a new subsystem.

3.4 Determining functionality for BASE24-eps scripting

BASE24-eps supports scripting in the areas of authorization, extract, reporting and journal perusal (as described further in 2.4, “Scripting component” on page 23).

Functionality such as the business rules for authorization will therefore be replaced by BASE24-eps scripting. BASE24-eps provides a library of Authorization Scripts which implements functionality equivalent to BASE24 Authorization. Consequently, the migration effort required in this area of a BASE24-eps migration is usually significantly less than if the scripts had to be designed, built, and tested from scratch.

3.5 How complex is a migration?

This is a difficult question which does not have a generic answer. Factors that contribute to a complex migration include:

- The business supports both Issuing and Acquiring Functionality.
- The target system needs to support multiple countries.
- The target system needs to support many regional/domestic interfaces.
- The existing system is heavily customized.
- The existing system is poorly documented.
- The migration is required to replace the system in a like for like manner. This can sometimes restrict the flexibility of the migration approach.

3.6 Summary

In this chapter we have explored the issues which have to be considered when determining the scope of a migration. Typically workshops, interviews and checklists are the tools used by the consultants and solution architects engaged in assessing the size and complexity of the migration.

Once the scope is properly understood and agreed, the migration approach and planning can be determined; this is discussed in the following chapter.

4

Designing a migration plan

Having defined the *what* for the migration to BASE24-eps by determining the migration scope, this chapter will discuss the *how* of the migration.

An important decision to make is whether the transition from the current Payment Engine to BASE24-eps should be an instant switch-over, that is, all at once, or a phased approach. As a Payment Engine is considered mission-critical by most customers, a phased migration approach is usually chosen to reduce migration complexity to a series of smaller, more manageable steps, thereby reducing the risks involved in the migration. A further significant advantage of a phased approach is that as each business component completes its migration, it can be moved into maintenance and further developed to meet changing business and compliance needs. In today's competitive debit and credit card business, it is most unlikely that the functionality of the whole system can be frozen for the duration of a complete migration. Splitting the migration into smaller migration phases means that the business only has to freeze the part of the system currently in the migration process.

During a phased transition, the details of which steps to undertake and in which order need careful consideration and will be influenced by factors like the flexibility of the parts of the organization impacted, the transaction volumes, etc. A phased transition also dictates the need to run parts of the old and new environments together so that they function as a 'hybrid whole'. This impacts both IT and business operations.

This paper recommends that a migration of any mission-critical installation be managed as a program of projects, preceded by an assessment and planning phase which defines the overall migration strategy, in terms of major milestone steps, each of which can be managed as an independent project within the overall program. Program management will co-ordinate and oversee these multiple related projects to implement the chosen migration strategy.

This chapter sets out a set of generic principles to which all but the most trivial of migrations should aspire to adhere. It then presents a set of system factors which will influence the detailed migration planning. Finally, it defines a *reference strategy* for a typical Migration Program, which may be used as a blueprint against which the phasing for any specific Migration Program may be assessed.

4.1 Migration principles

Any Migration Program must be designed to satisfy the wants and requirements of the organization which operates the Payment Engine, and may be constrained by commercial, logistic and technical limitations. In this respect, no two migrations will be the same, but it is nevertheless illustrative to consider the following criteria. Any migration should be designed to follow as many of these guiding principles as possible (in as far as they are applicable to that particular migration). However, it should be realized that it is highly unlikely that in any real-world program can be satisfy all of these guidelines:

- Maintain the attributes of the system during migration
- Make the migration as transparent as possible to:
  - Cardholders
  - Merchants
  - Card Associations
  - Financial Institutions
  - Branch staff within the organization
  - System users within the organization
- Realize new business benefits as soon as possible
- Minimize (dis)investment in temporary Transition Software
- Make migration phases as small as is sensible from a cost, risk and control point of view while minimizing the number of phases

4.1.1 Maintain the attributes of the system during migration

This principle is *to keep the system working*. The aim here is to have minimal downtime in the migration from the current environment to BASE24-eps.

Furthermore, even if the migration strategy dictates that there will be periods when old and new components will be running together as part of a 'hybrid whole', it must be possible to demonstrate that the total solution remains in (financial) balance.

4.1.2 Transparent to cardholders

This principle is to ensure that no unnecessary change is made to the cardholder, for example, re-issue all cards. Sometimes this is unavoidable, for example, a migration to EMV Chip and PIN cards.

4.1.3 Transparent to merchants

This principle is to ensure that no unnecessary change is made to the merchant base, for example, install new POS devices. Sometimes this is unavoidable, for example, merchant may need to perform an end of day to reconcile the POS device.

4.1.4 Transparent to Card Associations

This principle is to ensure that no unnecessary change is made to the interface to the Card Association, for example, revert to a previous mandate. Sometimes this is unavoidable, for example, there maybe a need to recertify the interface.

4.1.5 Transparent to Financial Institutions

This principle is to ensure that no unnecessary change is made to the interface(s) to the Financial Institution(s), for example, migrate from ISO 8583 1987 to ISO 8583 1993.

This could make commercial sense if the interface to the Financial Institution is customized and BASE24-eps can replace this interface with a standard product interface that is maintained, enhanced and supported by ACI Worldwide.

4.1.6 Transparent to branch staff

This principle is to ensure that no unnecessary change is made to the systems within the branches, for example, replacing the ATMs. Sometimes this is unavoidable, for example, branch staff may need to manually reconcile the ATM.

4.1.7 Minimize disruption to system users

This principle is to ensure that business processes impacted by the move to new technology have a 'clean' transition. Business processes (such as fraud analysis or chargeback processing) would generally rely on data produced from a 'back office' environment. However, it may be that there are groups of users that work directly in the Payment Engine environment. These users should ideally be faced with only one transition from old to new, migration steps should avoid the need for the users to work on both old and new at the same time.

4.1.8 Realize new business benefits

This principle accepts that a critical success factor for a migration is to realize a business benefit that the current environment cannot do. This will encourage the 'business' to provide the budget for the migration where technical or infrastructure reasons may not. For example, EMV, SEPA, Stand-in, etc. The earlier these benefits can be realized in the Migration Program the better.

4.1.9 Minimize (dis)investment in temporary transition software

This principle is to minimize the need for temporary interfaces between the old and new systems. Such 'throwaway' software costs time and money to build and test, and puts extra strain on operational management. However, it has to be accepted that all migrations will involve some functionality that is specifically required for the migration and serves no useful purpose later. A good example is conversion programs.

4.1.10 Migration phases

This principle reconciles the number of migrations against the size of the migration. Every migration requires a lot of effort, go-live support, unsociable hours, etc. So it is recommended to migrate in as small a number of phases as possible. But this is balanced against the risk in having too few phases, for example the 'big-bang' approach.

4.2 The migration landscape

This section briefly discusses the most significant system and environmental factors that will need to be taken into consideration when designing the Migration Strategy.

4.2.1 Issuing functionality

This is the part of the Payment Engine that authorizes transactions based on at least a Card File. Account information such as Balances can be stored either in BASE24-eps or in a Core Banking System.

There are two major Issuing transactional components to a system such as BASE24-eps, these are:

1. Interfaces - for example, VISA and AIB interface (Application Interface Block - a type of host interface) in the Smart Bank Environment
2. Authorization

Interfaces

In an Issuing environment interfaces receive transactions (and file updates) from the International and Domestic Card Schemes.

Considerations for interfaces are:

1. Interfaces are well defined.
2. Interfaces are few, for example, in the Smart Bank system this is the VISA Interface and the AIB Interface to the Fidelity Core Banking System.
3. Interfaces for the International Card Schemes require certification. Certification slots need to be booked in advance with the International Card Schemes.

Authorization

Considerations for authorization are:

1. Flexibility of BASE24-eps Authorization can be introduced to the customer's Payment Engine to introduce new business functions, for example, EMV, Stand-in processing, referral management, etc.
2. The impact can be restricted to segments of the customer's card portfolio, for example, debit cards.
3. Customer's authorization processing can be extremely customized.
4. Customer may not fully understand the authorization logic of the existing system to be able to convert this into BASE24-eps Authorization Scripts.
5. Usually requires a Card File (and possibly a balance file) to be migrated from existing Payment Engine to BASE24-eps.

4.2.2 Acquiring functionality

This is the part of the Payment Engine that acquires transactions from devices or interfaces. An Acquiring System usually has a terminal database that has information about all terminals connected to it.

There are two major Acquiring transactional components to a system such as BASE24-eps, these are:

1. Device Management - for example, IFX ATM Interface in the Smart Bank Environment
2. Interfaces - for example, VISA and AIB Interface in the Smart Bank Environment

Device Management

Considerations for migrating devices are:

1. There are many devices. Some Payments Engines manage hundreds of thousands of POS devices and tens of thousands of ATM devices.
2. Devices involve merchants for POS and branches for ATM. Introducing change for these environments requires very careful planning, especially with the merchants and the branch staff.
3. ATMs and POS Devices are customer facing so reputation risk can be high in an unsuccessful migration.

Interfaces

In an Acquiring environment BASE24-eps sends transactions (and file updates) to the International and Domestic Card Schemes. Considerations for interfaces are:

1. Interfaces are well defined.
2. Interfaces are few, for example, in the Smart Bank system this is the VISA Interface and the AIB Interface to the Fidelity Core Banking System.
3. Interfaces for the International Card Schemes require certification. Certification slots need to be booked in advance with the International Card Schemes.

4.2.3 Extracts

In BASE24-eps this is the mechanism of extracting transactions that are logged in the BASE24-eps Journal into a 'flat file' that can be imported into a Back Office.

In BASE24-eps there is no fixed extract format. The format is defined via BASE24-eps Scripts. This means that there is a large amount of flexibility in how to provide the extract file to a Back Office.

4.2.4 Communications

The data communications 'landscape' within which a BASE24-eps solution is to be implemented would normally be dictated by the customer's existing communications environment. Requirements relating to firewalls, routers, IP sniffers, and so on, have to be quantified and it must be determined how they can be accommodated. Sometimes a migration can be a catalyst for change between the current environment and the BASE24-eps solution, for example, ATMs can be SNA or X.25 connected in the current environment whereas for the BASE24-eps solution they may be TCP/IP connected.

4.2.5 User interface migration

This is usually a re-training exercise, that is, the customer is familiar with their “green screen” interface and now will use the ACI Worldwide Desktop. This discussion can broaden into a browser or Web Service debate, especially if there is a need to integrate BASE24-eps into a Branch or a Call Center environment.

4.2.6 Data(base) migration

The analysis required for mapping the database of one Payment Engine to BASE24-eps is a significant task, the success of which will largely dictate the success of the migration program as a whole.

The best way to migrate data sets is not to convert them at all. This strategy is usually appropriate for the data sets required for the Issuing side of the business; typically these data sets are maintained as 'shadow' copies of Back Office data sets. As update and refresh interfaces probably have to be developed anyway for ongoing maintenance of this data in the new environment, these new interfaces should be used for initial population of the data set in that new environment. For the BASE24-eps target environment this strategy would usually be applied to the:

- Cardholder File - This is refreshed from a Back Office, for example, ACI Worldwide Payments Manager or ACI Worldwide Card Management System.
- Positive Balance File - This is refreshed from a Core Banking solution, for example, in the Smart Bank solution this would be Fidelity Core Banking.

The data sets used by an Acquiring system usually present more significant challenges. The BASE24-eps target environment requires:

- Merchant Files
- Terminal Files
- Security Files

Typically this data is 'owned' by the existing Payments Engine, and it therefore has to be migrated from the existing environment.

For the migration of relatively 'static' data there are two main options, which are:

- Designing and programming Custom Conversion programs that understand both the source of the data and the destination of the data and transform the required data entities into a format appropriate for the BASE24-eps target environment.
- Use of standard tooling which can be parameterized to map data from one format to another and from one platform to another.

As a general rule, any migration strategy should avoid situations where data, and especially data of a dynamic nature, has to be kept synchronized across both the old and new environments. However, this cannot always be avoided, and is best solved using replication tools that can detect if changes have been made and re-apply those to BASE24-eps.

4.2.7 Addressing data conversion risk

Conversion of a major financial data set involves significant costs and business risks, which are difficult to quantify. Data has often become 'polluted' over the years, often as a result of maintenance via data entry applications which do not enforce integrity and consistency conventions. These uncertainties can be addressed by engaging data conversion specialists, at, for instance, IBM Center of Competency for Data Migration Solutions. This organization specializes in analyzing, cleansing, transforming, and migrating mission critical financial systems that require exacting attention to detail. Staffed by a dedicated team of specialists who apply a repeatable methodology, developed and enhanced during upwards of 2000 successful conversions, this organization would usually commit to a fixed-price participation in a migration program following an initial analysis study.

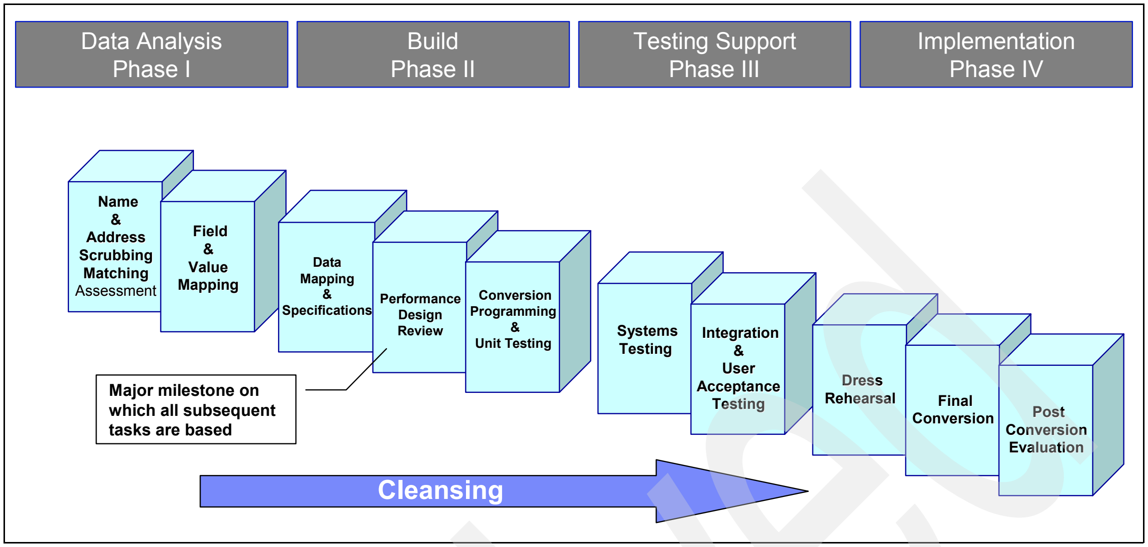

Data Analysis
Phase I
Build
Phase II
Testing Support
Phase III
Implementation
Phase IV
Name & Address Scrubbing Matching Assessment
Field & Value Mapping
Data Mapping & Specifications
Performance Design Review
Conversion Programming & Unit Testing
Systems Testing
Integration & User Acceptance Testing
Dress Rehearsal
Final Conversion
Post Conversion Evaluation
Major milestone on which all subsequent tasks are based
Cleansing

Figure 4-1 Data migration phases

4.2.8 Archival

There will usually be legal, business, and compliance requirements for archival of transaction records which have to be taken into account. A requirement to have records available for 360 days is not unusual. If there are log files or other data that are subject to these requirements, and these are in a format or on a media that can only be read on the old infrastructure, then it has to be decided if (part of) the old infrastructure be left in place until the data retention period has elapsed, or alternatively to convert the archive records for use by the new solution.

4.3 The reference strategy for migration

The following sections show a high-level strategy for the migration of an online transaction processing environment.

At a fundamental level it has to be decided which part of the business processing to migrate first. This choice can be quite simple if the system is only an Acquiring System or Issuing System but not both. If the System supports Issuing and Acquiring then the migration will be more complex and more migration choices will exist. The environment modeled here is designed to be representative of a typical combined Issuing and Acquiring system, but can be readily adapted to suit other 'as-is' situations.

The strategy shown is described at a level where each step would typically be performed as a sub-project within the context of an overall Migration Program. The strategy is described in terms of the online transaction processing, and is intended to function as a frame of reference, based on which an actual strategy, tailored to the needs and priorities of a real Migration Program, can be planned.

Any detailed migration program planning will obviously have to take clearing, settlement, data maintenance, reporting, audit, customer service, IT Operations and other processes not shown here into consideration. For example, the Card Processing infrastructure within the Financial Institution can be huge determining factor in resolving the sequence of a BASE24-eps migration. This can include:

- The Core Banking System for Debit transactions
- The Card Management System for Credit transactions
- The Fraud Management System

This migration strategy is intended as a reference model, not a magic solution to suit all circumstances. It has however been designed to be a good match against the guiding principles for migration (see 4.1, “Migration principles” on page 34). In particular, transactions are either carried by the old or the new environment, but never routed through both. This means that each system can be independently shown to be consistent.

4.3.1 The conceptual model

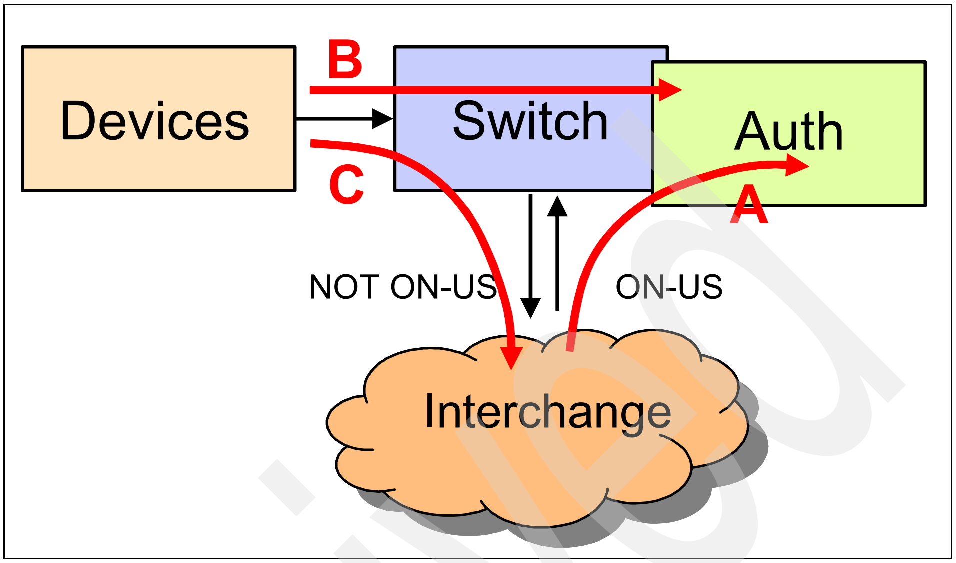


Figure 4-2 The conceptual model

The strategy will be described in terms of the simple environment model shown here. Note that the terms ON-US and NOT ON-US are used as seen from the Issuer standpoint, that is, an ON-US transaction is one for which the migration environment has authorization responsibility, a NOT ON-US transaction is one for which request messages have to be routed elsewhere.

The following components of the environment can be distinguished:

- The *Devices block* represents all ATM and PoS devices which fall within the scope of the environment being migrated, but also represents any other channels that source authorization, withdrawal, transfer or deposit requests towards the Payments Engine (for example, contact center agent applications, IVR, referral department, Internet bank, etc.).
- The *Interchange cloud* represents one or more Interchange networks to which the migration environment is connected. In any real-life situation there would often be multiple interchange network connections (for example, American Express, MasterCard, Visa, domestic) to be migrated. Note that the acquiring and issuing connections to the Interchange cloud might well be over the same physical connection.
- The Auth block represents the migration environment's own authorization process(es), and associated data structures such as cardholder balance information. Authorization nodes belonging to other Issuers are considered to

  be in the Interchange cloud, as are delegated stand-in authorizers which may act on behalf of the migration environment; these are both assumed to be unaffected by a migration.

  Three Authorization architectures are commonly found:

  - End-point authorization modules on a remote machine, typically an IBM mainframe application.
  - End-point authorization modules on the same platform as, but independent of, the Switch.
  - *Nodes* on the switch that are configured as “permanent stand-in” authorization modules. Commercial switch products often include functionality to authorize “in the switch” if a remote authorization end-point (such as a mainframe-based authorizer) is not available. If the implementation does not actually have an end-point authorizer then this mechanism may be used permanently to host the authorization functionality.

- The *Switch* component represents the system logic that passes requests and responses to and from devices, interchange networks, and authorizers. As such, the Switch is responsible for routing according to the appropriate business rules, and for conversion of message layouts, transport protocols, and security.

The conceptual model for the “as-is” situation supports the following major transaction flows:

- A: Transactions on cards *belonging to* the migration environment that are acquired elsewhere are routed through the system for authorization.
- B: ON-US transactions from the migration environment's own acquiring network are routed through the system for authorization.
- C: NOT ON-US transactions from the migration environment's own acquiring network are routed out to the interchange network(s) for authorization elsewhere.

4.3.2 Step 0: Mainframe authorization

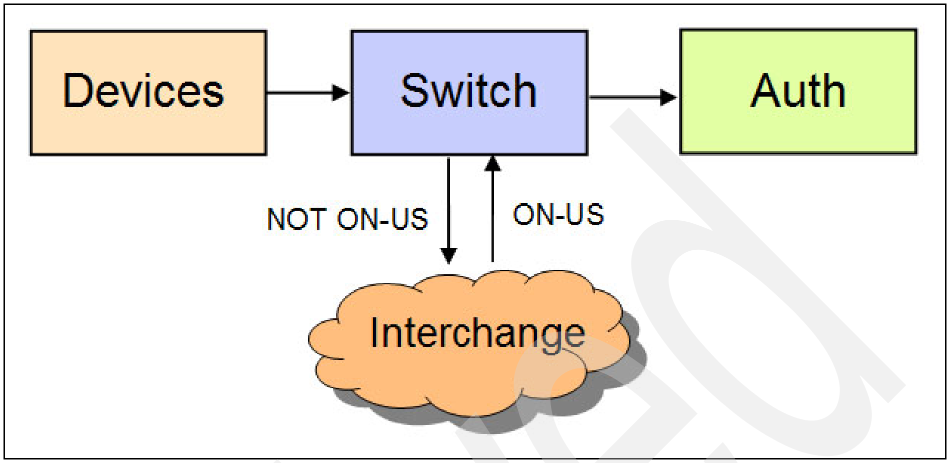


Figure 4-3 Step 0: Mainframe authorization

The purpose of this initial step is to move the Authorization functionality away from the existing Payments Engine and onto the new target platform. If this can be achieved the Migration Program can provide significant early benefits to the business: typically the deployment of BASE24-eps authorization technology gives the business access to a powerful scripting engine which allows the definition of transaction processing rules without having to modify program code, and the potential for integration with other systems (for example, core banking, CRM) for data to be used in the authorization process.

Depending on the particular circumstances of the 'as-is' situation this step may be combined with Step 1 in order to avoid a necessity for changes to the Switch component. For each of the three commonly found authorization architectures the following considerations apply:

- End-point authorization modules on a remote machine: In this case if the authorization is already on the target platform this step could be omitted. However, a decision could be taken to replace the existing Authorization module by BASE24-eps authorization as a discrete, initial step. Also if the authorization is not on the intended target platform, it could be replaced by BASE24-eps authorization. In both cases there is a trade-off to be considered: the benefit of early implementation of the new authorization module might require modifications to the existing switch (to change the way it communicates with the new remote authorization module). It could well be preferable to combine this step with Step 1 in order to avoid the necessity of building and testing such a temporary interface.

- End-point authorization modules on the same platform as, but independent of, the Switch: Implementation of BASE24-eps remote authorization should be considered. Again, any potential benefit should be weighed against the effort required to make the Switch and new remote Authorization modules communicate.
- 'Nodes' on the switch that are configured as 'permanent stand-in' authorization modules: In this case the existing Switch has standard functionality to support remote authorization end-points. Again, the considerations in the previous bullet apply.

4.3.3 Step 1: Implementation of the new switch

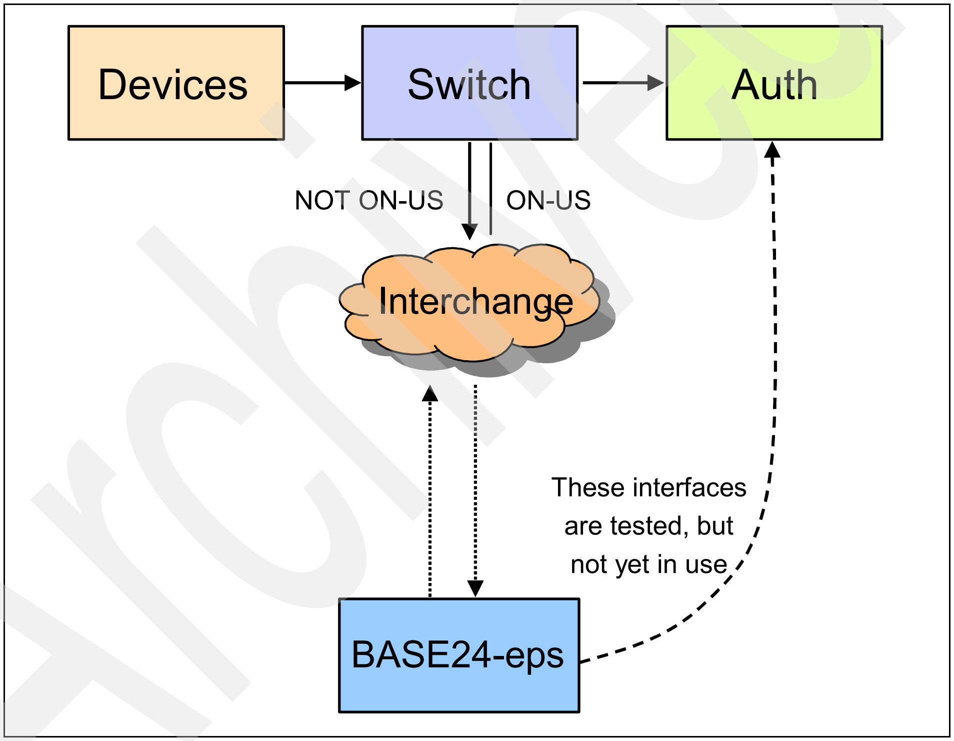


Figure 4-4 Step 1: Implementation of the new switch

In this step the Switch functionality of the new environment is designed, configured, tested and commissioned, and connected to the mainframe authorization modules.

This step has of itself no inherent business value because it carries no production traffic. However, as a risk reduction strategy it is not to be underestimated. Typically, when the introduction of any new software solution brings with it the necessity to introduce unfamiliar infrastructure, there is a steep learning curve for the operational departments charged with managing the required new hardware partitions, applications, networking, security infrastructure, file maintenance and associated operating processes. This translates into business risk. By explicitly separating the technical activation from the activation of business functionality this risk can, to some extent, be mitigated.

As the components being 'technically implemented' are typically subject to stringent performance, scalability and availability requirements, this step requires major requirements analysis, IT development and architecture effort and is a significant milestone with the following benefits:

- The new IT and communications technology has gone live. Responsibility for the new platform and authorization interface has moved from development to IT Operations. IT Operations can gain experience with the organizational processes required to monitor and manage the system within the agreed Service Level Agreements.
- Any batches or other things (for example, file maintenance) needed for running the old and new Payments Engines in parallel during later steps can be brought into production.
- Because there are as yet no financial transactions carried by the new environment there is little or no risk to business continuity.

4.3.4 Step 2: Issuing functionality

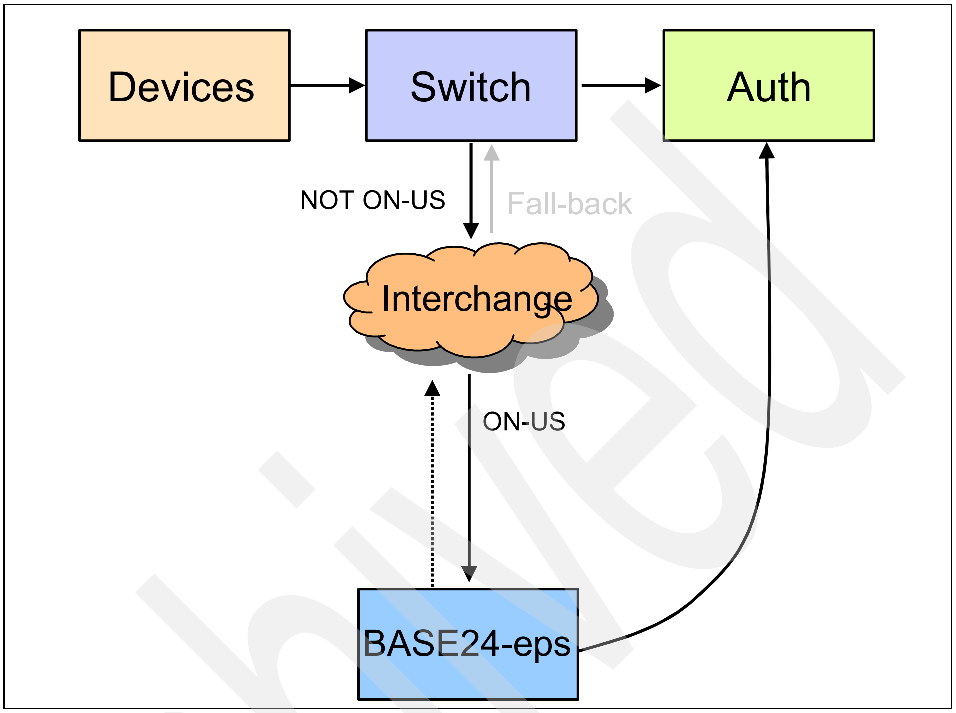


Figure 4-5 Step 2: Issuing functionality

If the interface to the interchange gateway supports multiple physical connections, then there is an opportunity to mitigate risk by taking smaller, business process oriented migration steps, as shown here where the new environment is connected to the interchange network(s) for incoming (ON-US) traffic. This moves issuing traffic (and the associated business processes) onto the new infrastructure, but defers the migration of transaction acquisition.

This step would normally be implemented one network at a time. Each interchange partner could even change the routing by card sub-range, so that volumes were ramped up at a manageable rate. Per interface sub-step the fallback is to have the interchange partner put transactions back on the old route to the old switch component.

On completion of the step all of business flow A (transactions on cards acquired elsewhere are routed through the system for authorization) has migrated. Assuming that throughout this step all authorization processing is centralized on the mainframe, there is always a consistent customer view in one place.

During the transition period there are a number of challenges to be addressed:

- Transaction log files will be spread across the hybrid whole. However, both environments can be balanced independently in order to verify integrity.
- If there is stand-in functionality in the switch (rather than in the cloud) then there could be a necessity to manage and keep in sync two stand-in environments.
- Two sets of security infrastructure have to be managed and kept in sync (or alternatively the old and the new environment must be able to use the same security devices).
- Depending on what was achieved in Step 0, there may not yet be one single point of authorization which feeds details of authorized transactions to the clearing process (for matching against the incoming clearing). The clearing system will have to maintain interfaces to both the new and old authorization modules.

4.3.5 Step 3: Acquiring functionality

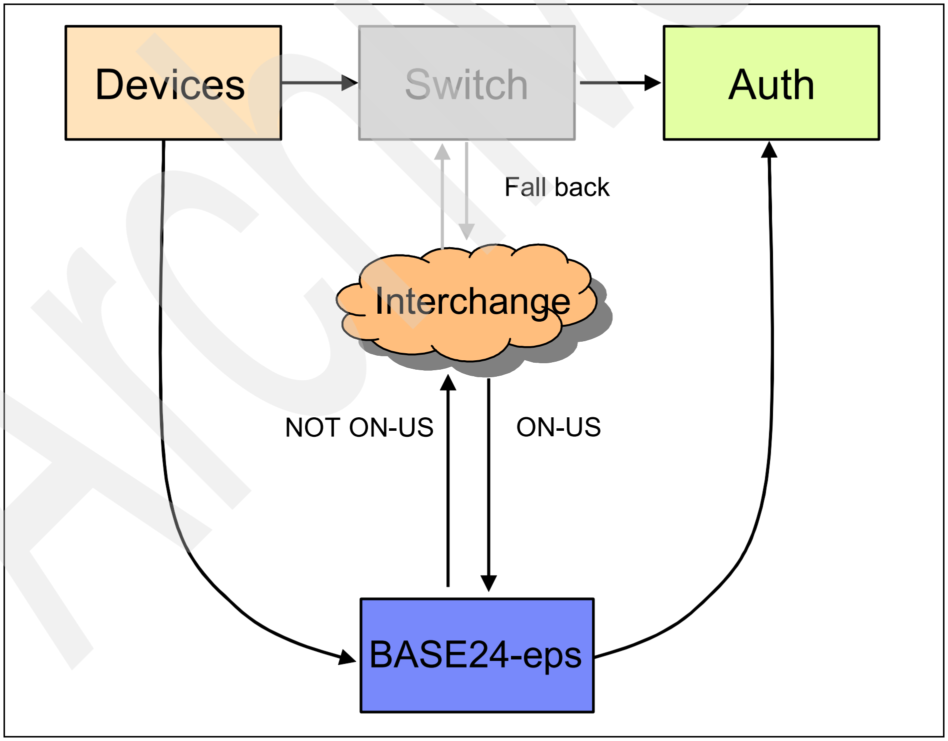


Figure 4-6 Step 3: Acquiring Functionality

This step migrates the devices across to the new environment. This activates the remaining business flows B and C on the new infrastructure.

Migration from old to new can be managed at a speed and complexity as determined to be appropriate by the IT and business organizations, for example:

- Per transaction sort (PoS, ATM, Internet bank, etc.)
- Per interface
- Per device type

4.3.6 Step 4: Completion

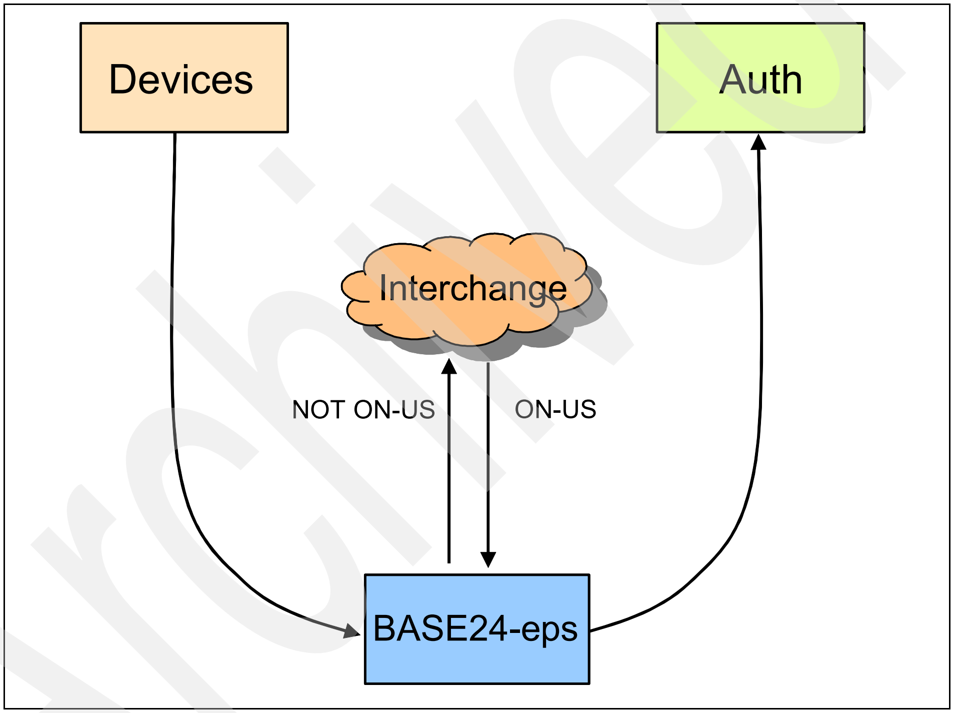


Figure 4-7 Step 4: Completion

The final step is to decommission the old infrastructure.

This sub-project should take transaction archival requirements into account (see 4.1, “Migration principles”) and therefore may involve some data conversion in order to make archives available in the new environment. Consideration should also be given to particular requirements for the secure disposal of encryption equipment.

4.3.7 Alternatives for Steps 2 and 3

As previously noted, the approach described in Steps 2 and 3 is only feasible for interchange connections where is it possible to terminate the ON-US and the NOT ON-US transaction flows independently of each other on the old and new switch machines.

If this is not possible, there are alternative strategies, although these require larger migration steps, or the introduction of temporary interfaces (that is, additional effort and risk).

The first alternative is the combination of steps 2 and 3. The physical interface to the Interchange network moves in one step from the old switch to BASE24-eps, and with it the logical ON-US and NOT ON-US transaction flows. The consequence of this is that the devices also have to migrate to the new environment, otherwise NOT ON-US transactions would have no route through the old switch onto the interchange network. The resulting combined step essentially becomes a 'big-bang'.

The second alternative augments Step 3 by engineering a transaction path from the old switch to the Interchange via a temporary interface to the BASE24-eps switch component, as shown in the diagram below. This defers migration of the devices to a discrete Step 4, but requires development of a 'throw-away' interface in the old and new switches. However, this extra work might be considered preferable to a 'big-bang' scenario.

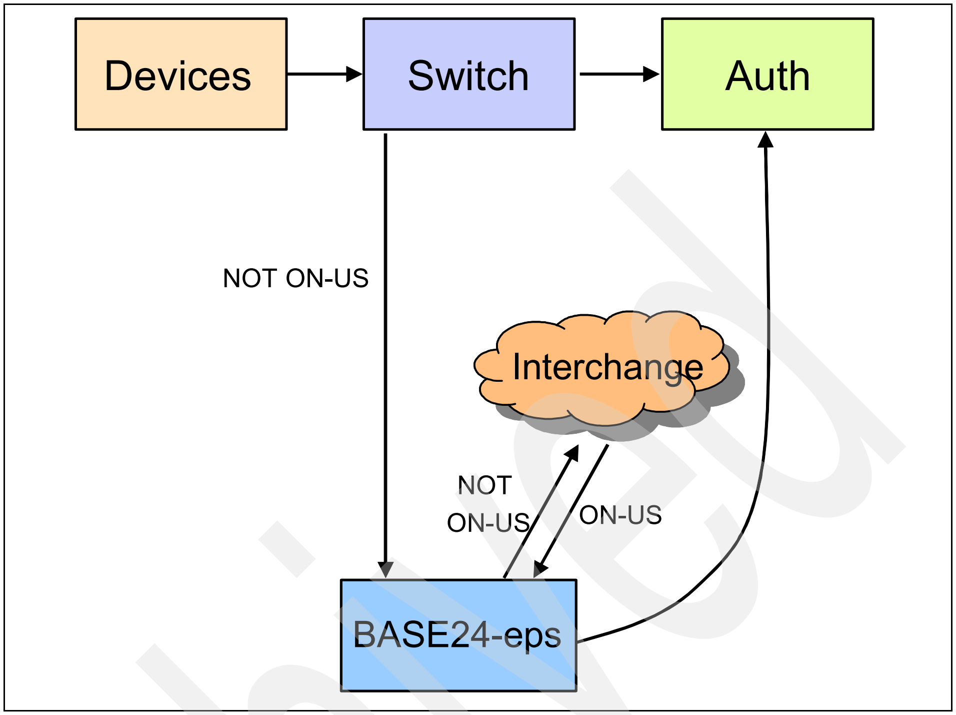


Figure 4-8 Alternatives for Steps 2 and 3

4.4 Summary

A significant output of a pre-assessment study should be the top-level strategy for a feasible migration. Feasibility is not just a technical and architectural issue, but also encompasses time and budget constraints, and should address business uncertainties and realization of potential benefits. Such a strategy should:

- Minimize, control and manage risk to the business
  - Clear planning of each step so that the impact on the organization is understood.
  - Avoid 'big-bang' scenarios: move in manageable steps with clear fall-back strategies.
  - Where possible, avoid end-user departments having to use both systems at once. Each impacted department should ideally only have to undergo one set of retraining and one migration moment.

  - At all stages 'system balancing' should be possible so that it can be shown that there are no missing transactions.

- Minimize 'throw-away' costs
  - The migration strategy should minimize (ideally remove) the need for temporary interfaces between the old and new systems. These cost time and money to build and test, and put extra strain on operational management.
- Minimize the elapsed time
  - … but not at the cost of risk to the business. It is counterproductive for the technical implementation to move fast if the business organization cannot keep up. Conversely, the planning should also take into account the quality and availability of knowledge (documentation, motivated people) of the existing functionality, as this will directly influence the ability of the technicians to keep up with ambitious domestically imposed by business needs.
  - During the migration period:
    - Two systems have to be licensed
    - A migration team has to be maintained
    - There is a potential confusion/disruption in the user organization

The options to consider when designing the Migration Strategy are:

- Big-Bang: Generally perceived as the most risky strategy. Not normally considered for large or complex systems.
- Phased:
  - The details of the migration phasing will obviously depend upon the nature of the business being migrated (Issuing, Acquiring, both). If the System supports both Issuing and Acquiring then it is usually simpler to migrate the Issuing component before the Acquiring component. A reference strategy has been presented as a basis for designing a real-world Migration Program.
  - In the example of migrating to the BASE24-eps Smart Bank showcase solution the following strategy was derived from the reference strategy:
    - Migrate the Debit Issuer functionality as Phase 1.
    - Migrate Acquirer 'not-on-us' transactions as Phase 2.
    - Migrate the Devices (that is, ATMs) as Phase 3.
  - Within each phase business volumes can be ramped-up at manageable rates by successive enabling of infrastructure or business components, such as:
    - Interchange Interfaces

- Card Programs
- Business processes (PoS, ATM, credit, debit, etc.)
- Device type

In this chapter we have illustrated the need to determine an implementation strategy based on reducing risk, showing early business benefit and preserving financial integrity. Furthermore, a high-level reference model is proposed on which a real-life strategy can be based. A Migration Program approach is recommended as the best way to manage the implementation of the chosen strategy.

5

The IBM Smart Bank showcase environment

The Smart Bank showcase, located in IBM Montpellier, France, is an example of a target platform for BASE24-eps migration. The Smart Bank solution is based upon an IBM System z. The showcase is to demonstrate the power of the on demand operating environment, software, and infrastructure. The live operations simulate a medium sized European bank and inject a realistic and representative workload to the operating environment. This activity is used as the back-drop to demonstrate some of the key business proof-points and challenges that are facing the retail banking industry today and in the future. The timeline of a typical banking day is used as the background for the showcase.

5.1 BASE24-eps IBM Smart Bank environment

The BASE24-eps IBM Smart Bank comprises:

- 3,000 IFX ATMs. Most of these will be simulated by the IBM Rational® product, but real Wincor IFX ATMs are also in the solution.
- Managing the ATMs using the Tivoli® Enterprise™ Management product.
- The VISA interchange to Authorize Credit Card transactions for the Smart Bank. This simulates a Financial Institution that has outsourced its credit card portfolio to a third-party processor. The VISA interchange will be simulated by the Paragon® product.
- Acquiring transactions through the VISA interchange.
- Sending *Not-On-Us* transactions to the VISA interchange.

5.2 The Smart Bank

The IBM Smart Bank Showcase has:

- 3,000 ATMs
- 4,805,787 current accounts
- 3,587,178 savings accounts

5.3 BASE24-eps Smart Bank capabilities

The BASE24-eps Smart Bank capabilities are:

- Extend continuous operations to cover the retail payments context.
- Show how BASE24-eps performs and benefits on the underlying z/OS® platform.
- What qualities of service does it inherit?
  - Availability and resiliency
  - Scalability (linear from the benchmark results)
  - Security
  - Workload Management - Qualities of service of the channels - response time
  - Longevity of the platform

- Manage failover and recovery scenarios and inclusion into the existing showcase proof-points.
- Show statistical reports by ATM Device and by BIN (Bank Identification Number) - these are the key customer reports showing routing and ATM Device profitability.
- Integration of BASE24-eps to Retail Banking Core System (Fidelity Corebank). Create the interface to the card issuer - Fidelity Corebank in our case, so that the card issuer can authorize the transactions either from our own ATM network or from the international network acquired by other banks.

5.3.1 Smart Bank architecture

The following diagrams show the architecture of Smart Bank.

This diagram shows the applications that are present in Smart Bank, which include:

- BASE24-eps as the Payment Engine
- Fidelity as the Core Banking application
- Siebel® Data Warehouse

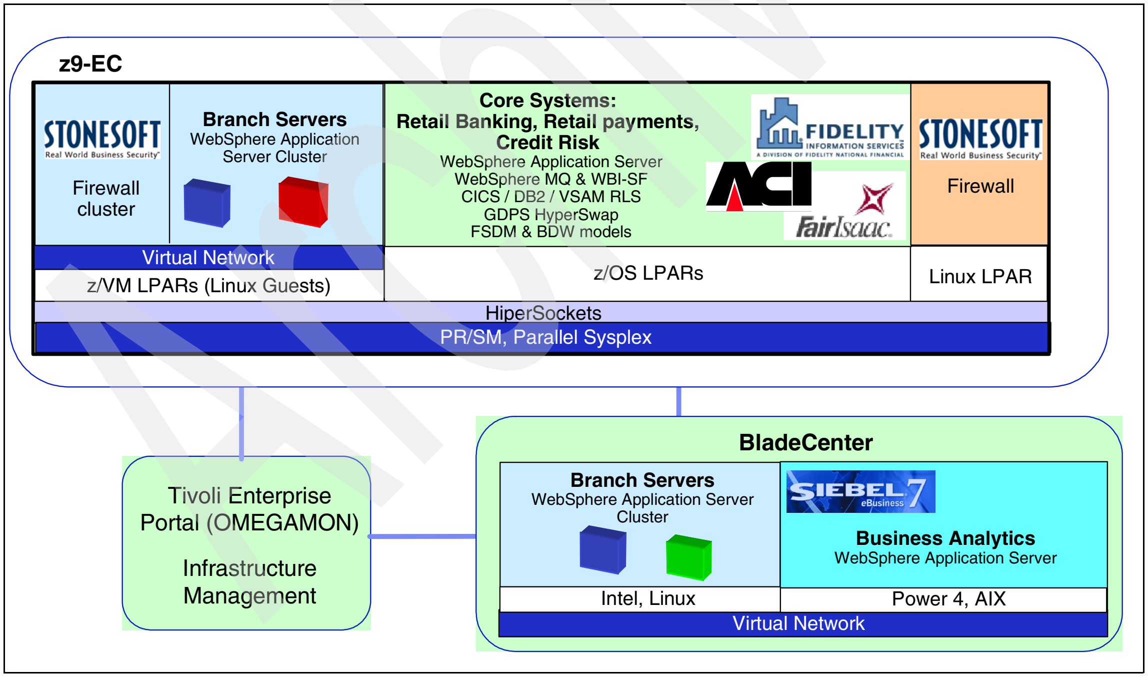

z9-EC
STONESOFT Real World Business Security
Firewall cluster
Branch Servers
WebSphere Application Server Cluster
Core Systems:
Retail Banking, Retail payments, Credit Risk
WebSphere Application Server
WebSphere MQ & WBI-SF
CICS / DB2 / VSAM RLS
GDPS HyperSwap
FSDM & BDW models
FIDELITY INFORMATION SERVICES
ACI
FairIsaac
STONESOFT Real World Business Security
Firewall
Virtual Network
z/VM LPARs (Linux Guests)
z/OS LPARs
Linux LPAR
HiperSockets
PR/SM, Parallel Sysplex
Tivoli Enterprise Portal (OMEGAMON)
Infrastructure Management
BladeCenter
Branch Servers
WebSphere Application Server Cluster
SIEBEL 7 eBusiness
Business Analytics
WebSphere Application Server
Intel, Linux
Power 4, AIX
Virtual Network

Figure 5-1 Smart Bank Architecture

This diagram shows how transactions are injected into Smart Bank, namely:

- Rational as the IFX ATM Network
- Paragon as VISA
- Rational as the Internet Banking channel
- Rational as the Branch channel

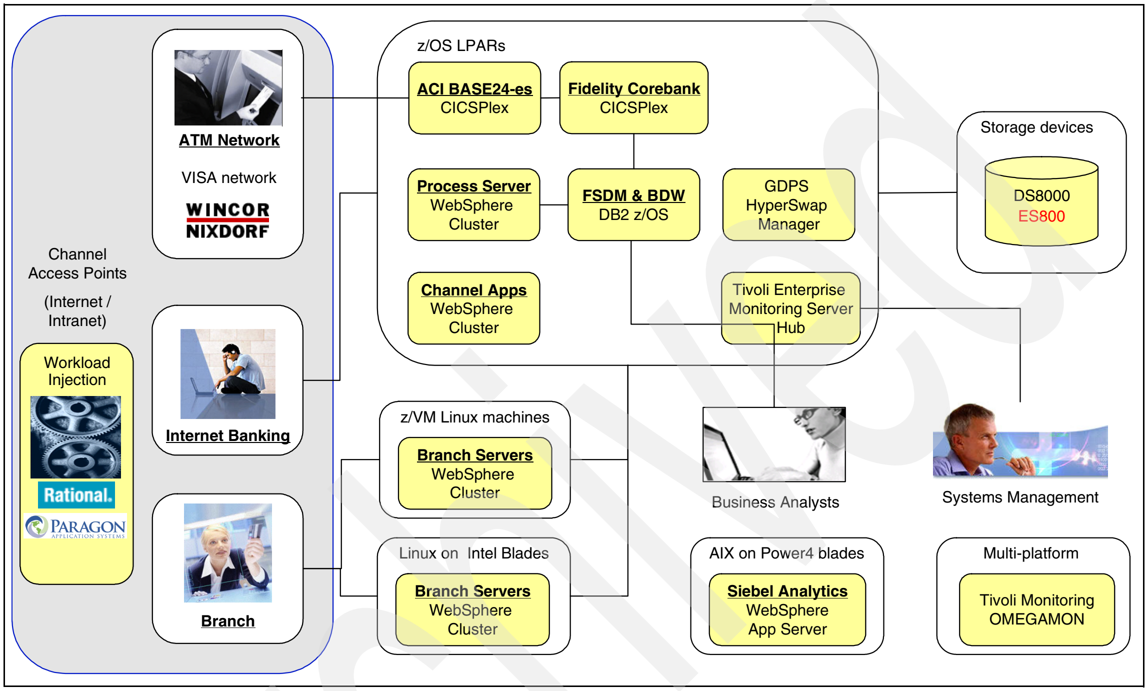


Figure 5-2 Smart Bank injection

This diagram shows the BASE24-eps Smart Bank architecture, namely:

- IFX Device Handler receiving messages via TCP/IP
- VISA Interface receiving messages via TCP/IP
- Fidelity Core Banking receiving messages from BASE24-eps via a SOAP Web Service

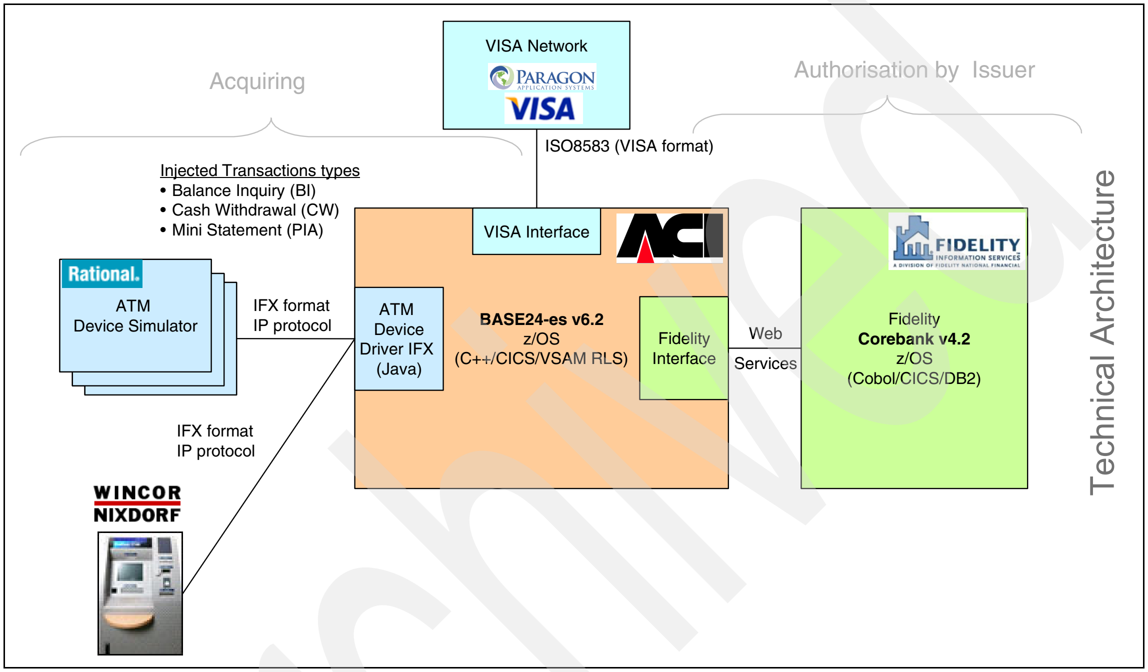


Figure 5-3 Smart Bank components

6

Sample of a migration to ACI Worldwide BASE24-eps based on the IBM Smart Bank environment

A BASE24-eps system is built around four basic components; a foundation component, an online transaction support component, a user interface component, and an operation support component. These components are a prerequisite for every BASE24-eps implementation. The components provide essential functions and support services for BASE24-eps, such as data and memory management, basic transaction processing services, and the user interface.

6.1 Migration plan

The migration plan provides the overall strategy for migrating and implementing the current customer Payment Engine to BASE24-eps. This migration will be in three phases, namely:

1. Migrate the debit issuer functionality as Phase I
2. Migrate acquirer *not-on-us* transactions as Phase II
3. Migrate the devices (that is, ATMs) as Phase III

6.1.1 Phase I

This phase migrates the Issuer Host Interface and authorization from the current environment to BASE24-eps.

This requires:

1. BASE24-eps to support the Fidelity Core Banking Interface (that is, the AIB) so it can communicate with the Smart Bank Core Banking application.
2. BASE24-eps to support a Card File, which will be refreshed by the Fidelity Core Banking solution. In other customer migrations this file could be supplied by ACI Worldwide Payments Manager or the ACI Worldwide Card Management System.
3. BASE24-eps to support a Positive Balance (Pbal) File, which will be refreshed by the Fidelity Core Banking solution. The BASE24-eps Pbal is required for Debit card stand-in processing.
4. BASE24-eps to send advice to the Core Banking solution in case of BASE24-eps 'standing-in' for the Fidelity application.

Important: The only Customer Specific Modifications (CSMs) for the migration to BASE24-eps on the Smart Bank environment was the AIB Interface.

Table 6-1 Phase I - BASE24-eps Products

Product Module
Central Services Package
Transaction Routing and Authorization
Card Authorization
Account Authorization

Transaction Security Services
Thales e-Security Device Control
ACI Worldwide Generic File Update Interface to BASE24-eps
BASE24-eps API VSAM Database
Payments Manager Online Extract Interface
EMV Issuer only Module
EMV Acquirer only Module
Multiple Currency Module

6.1.2 Phase II

This phase migrates the VISA interface. This requires BASE24-eps to support the VISA interface.

Some assumptions for this scenario are:

1. The VISA interface will be certified in a test environment.
2. BASE24-eps will need to provide all Interchange Prefix Routing Files (for example, IPF Refresh for VISA).

Table 6-2 Phase II - BASE24-eps Products

Product Module
VisaNet ISO (BASE I and SMS) Interface
IPF Refresh - VISA/PLUS/Interlink Interface

6.1.3 Phase III

This phase is for migrating all the ATMs from the current environment to BASE24-eps. This requires BASE24-eps to support IFX for the ATMs.

Table 6-3 Phase III - BASE24-eps Products

Product Module
IFX ATM Manager
BASE24-eps Automated Key Distribution System

Chapter 6. Sample of a migration to ACI Worldwide BASE24-eps based on the IBM Smart Bank

7

Closing perspectives

In the course of this paper we have concentrated on these major topics as they pertain to payments systems:

- The business needs that might lead to a migration of the IT solution
- The processes and mechanisms of the migrations themselves, and the associated infrastructure adaptations
- An example of a migration of a payments environment, using the IBM Smart Bank showcase as it is implemented in Montpellier, France, as a basis
- The principles and advantages of carrying out a pre-assessment project

These items are not really final objectives in themselves, but rather they are interrelated steps on the road to providing a payments infrastructure with the qualities of service demanded by the current (and probably future) globalized business ecosystem. The synthesis of the individual points contained in these topics with the solution provided by the ACI Worldwide BASE24-eps product, as described in the different chapters of this paper, can form the hub of a payments solution fulfilling the ever more challenging industry requirements. By further implementing this payment solution on an IBM System z infrastructure, it will, in all probability, be possible to attain levels of service, availability, resiliency and efficiency that is normally not found in the use of former or existing (legacy) or RYO (roll your own, or inhouse) application systems.

Another way of looking at migrations as they pertain to payment systems is to consider a rough breakdown of the worldwide financial services market for this industry segment. This breakdown, as applied to the top 500 banks worldwide, can be represented generically as follows.

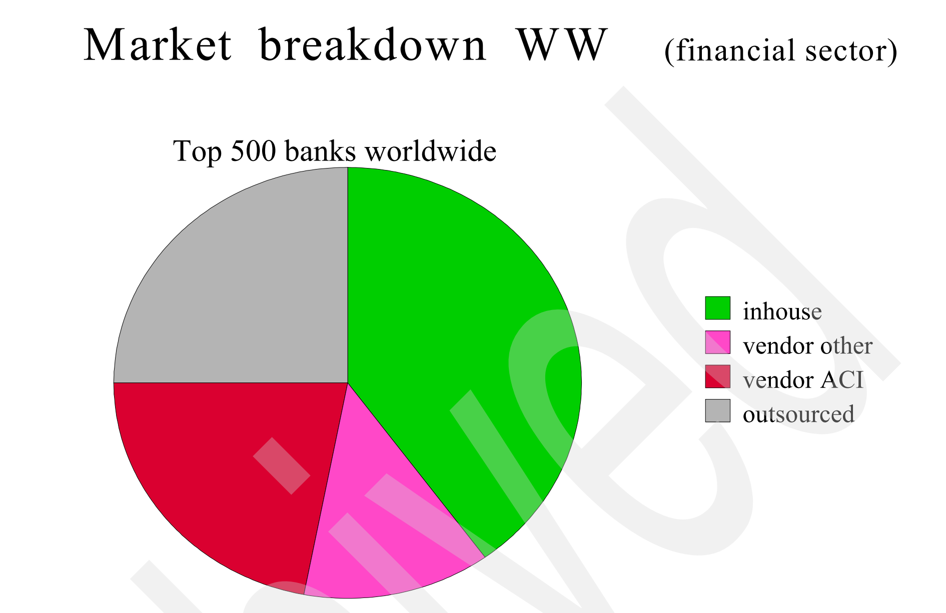


Figure 7-1 Market breakdown WW

This graphic shows that for this financial market segment, only about 35% of the installations are actually covered by vendor products. However, of these 35%, over 60% run on the ACI Worldwide product, which should lead us to the conclusion that the product has its renown in the market. Ignoring the outsourced installations for the moment (at least, for the purposes of this viewpoint), we see that the sector representing the inhouse-built solutions is actually slightly larger (40%) than that of the vendor products. We could then reasonably conclude that this considerable segment of the market could profit even more from the close interlock which exists between the payments solution based on the ACI Worldwide BASE24-eps product and the IBM System z which can serve as an infrastructure. The qualities of service of this platform, namely scalability, on-demand responsiveness, security, integrity, and the like, will allow consolidated operations and end-to-end integration in a measure scarcely possible with former or existing (legacy) implementations.

Furthermore, the cost of maintaining high availability and integrity in the former or existing (legacy) application environment should make the solution based on ACI Worldwide's BASE24-eps running on the IBM System z platform an attractive business proposition.

Whichever way we decide to look at an incumbent payments solution, from the technical or from the business viewpoint, we will probably find that the current environment can profit from the consideration of a migration to a more leading-edge implementation which in turn will ready the whole payments ecosystem to the current and upcoming industry challenges. In order to solidify our findings we have suggested that a pre-assessment study is probably the best path on which to proceed. We hope that this paper has provided the framework, the insights, and the encouragement to look afresh at existing payment solutions and their infrastructure, and to optimize their business value in the light of new product possibilities.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 69. Note that some of the documents referenced here may be available in softcopy only.

- *A Guide to Using ACI Worldwide's BASE24-es on z/OS*, SG24-7268

Online resources

This Web site is also relevant as a further information source:

- ACI Worldwide

  http://www.aciworldwide.com/

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

ACI Desktop 30
ACI Worldwide 17, 28
  BASE24-eps 61
  BASE24-eps product 15, 18, 65
  Card Management System 39, 62
  Desktop 39
  Generic File Update Interface 63
  Payments Framework 26
  Payments Manager 62
  product 66
  solutions process transaction 26
ACI Worldwide Payments Framework 26, 28
Acquiring/Issuing System 15
application area 15
Application Interface Block (AIB) 37, 62
ATM Device
  Handler 30
  profitability 57
Auth block 43

B

BASE24-eps implementation 30, 61
BASE24-eps Journal 38
BASE24-eps Pbal 62
BASE24-eps Script 39
BASE24-eps solution 22, 29, 39
business risk 2–3, 40, 47

C

card file 24, 37, 62
card verification 21
chargeback processing 36
component-based design 24
control point 34
customer relationship management (CRM) 20
cutover processing 23

E

EMV Chip 35
end-to-end integration 66

F

Fidelity Core Banking
  Interface 62
  solution 62
  System 37
Fidelity Corebank 57
Financial Institution 27, 34, 56
  Card Processing infrastructure 42

H

hardware security modules (HSMs) 21
HSM vendor 28
HTTP 29

I

IBM Montpellier 55
IBM System z
  infrastructure 65
  platform 66
IFX ATM
  Manager 63
  Network 58
industry-specific format 20
industry-standard hardware 20
industry-standard platform 25
inhouse-built solution 66
Interchange network 43
  old switch 51
Interchange Prefix Routing Files (IPF) 63
Internet bank 43
Internet portal 22
IP sniffer 39
IPF Refresh 63
ISO 8583 21, 35

L

logical ON-US 51

M

mainframe-based authorizer 44
mean time
  between failures 5

to repair 5
middleware system 2
migration effort 5, 30
migration issue 27
migration plan 27, 33, 62
Migration Program 34
fixed-price participation 40
migration strategy 29, 35–36
mission-critical installation 34
multi-language capability 25

O
object-oriented design 22
on-demand responsiveness 66
ON-US transaction 43
overall program 34
independent project 34

P
payment engine 17, 27, 33, 57
payment solution 67
personal identification number (PIN) 21
Phase II 62
Phase III 62
platform-specific component 23
platform-specific processing 22
point-of-sale (POS) 20
POS device 35
POS terminal 29
post-authorization process 19
potential confusion/disruption 53
pre-assessment cycle 9
pre-assessment phase 14, 27
pre-assessment project 65
pre-assessment stage 9
pre-assessment study 52, 67
significant output 52
pre-authorization file 24
pre-contract work 10
program management 34
project structure 28
pull-down menu 21

R
real time
definition 21
internal authorization 19
rebalancing 2
reconciliation purpose 18
Redbooks Web site 69
Contact us xi
regional/domestic interface 31
remote authorization
end-points 46
module 45–46
re-training exercise 39
return on investment (ROI) 8
robust feature 18
ROI consideration 14

S
sub-project 42

T
task-oriented view 21
terms ON-US 43
three-cycle scheme 9

U
Unified Modeling Language (UML) 25
user interface 21, 23, 61

V
Visa 30, 37
VISA interchange 56
VISA/PLUS/Interlink Interface 63
VisaNet ISO 63

X
XML standard 21
XML technology 24

Migration of Payments System to ACI Worldwide BASE24-eps

Redpaper™

Business and IT viewpoints of migration to ACI BASE24-eps

Designing a migration plan

IBM Smart Bank showcase environment

This IBM Redpaper discusses the migration of a customer's existing payment environment to a new solution, based on ACI Worldwide's BASE24-eps product (formerly known as BASE24-es), running on an IBM System z infrastructure.

This document covers the following topics:

- General considerations on migrations, their necessity, and financial implications
- An introduction to the ACI Worldwide BASE24-eps product
- Items to be migrated and scope of the work
- Designing a migration plan
- IBM Smart Bank showcase environment running in Montpellier, France
- An example of a migration to ACI Worldwide BASE24-eps based on the IBM Smart Bank environment

The choice of an infrastructure based on the IBM System z platform in this paper is primarily for reasons of document scope, and because this platform is frequently found as an integration hub in customer installations. It will be equally possible to perform the portrayed migrations to other infrastructures, based on platforms such as UltraSPARC from SUN, System p from IBM, Integrity from HP, and Integrity NonStop, also from HP.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-4337-00