

Strategies for payment systems planning

Gary S. D. Farrow

Received: 18th July, 2012

Triari Consulting, 30 Clothorn Road, Didsbury, Manchester, M20 6BP.
Tel: +44 (0)161 445 5958; e-mail: gary.farrow@triari.co.uk



Gary Farrow

Gary Farrow is Director of Triari Consulting, provider of systems integration and IT architecture services for the financial sector. A lead architect on many successful, high-value projects, he has undertaken senior architect roles at major banks and financial services firms in the UK. Gary has broad expertise in large-scale systems integration, enterprise service bus architectures and Service Oriented Architecture (SOA), and deep domain specialism in payments systems. He has written and presented many articles on SOA and payments processing, discussing wide-ranging systems architecture topics. Gary is a member of the IT Architecture Certification Board for the Open Group. His professional qualifications include Fellow IET, Chartered Engineer and Open Group Master Certified Architect. He holds BSc and PhD degrees from Manchester University and is an alumnus of Warwick Business School, UK.

ABSTRACT

The need for a structured approach to payment systems planning is driven by (i) the complexity of the Payments Integration Space, which defines the scale of the payment systems planning problem, and (ii) the Payments Integration Paradox, this being the idea that, while the objectives of the planning activity is simplicity in the IT estate, the phases in achieving the simplification are themselves highly complex. In this context, this paper introduces strategies for payment systems planning. Three macro strategies are described, which highlight long-term approaches to achieving a chosen IT target architecture.

Complementary micro strategies are also introduced, which highlight valuable localised approaches within a specific phase of the overall macro strategy execution. Several target architectures that achieve a simplified payment systems landscape have been identified previously. This paper explores the extent to which these architectures are fulfilled in the execution of the strategies, illustrated using a comprehensive functional model of payments processing. Given the dynamic nature of the payments business environment, any systems improvement initiative must be able to accommodate impacts due to 'shocks' within the environment. An overview of the variety of business events that can occur is provided, and a qualitative assessment of the robustness to each event of the candidate target architectures and associated strategies is made. The paper concludes by presenting case studies highlighting practical experiences in executing the different strategies within two major payment systems modernisation initiatives.

Keywords: payments strategy, payment systems planning, payments hub, service bus, simplification, systems integration

INTRODUCTION

Background

The payments business environment is such that payments initiatives within a bank are many and continuous:

- Competitive drivers demand that new initiatives are brought to market quickly to achieve ‘first mover’ advantage.
- Mandatory regulation and banking industry initiatives often set challenging timescales within which new and often complex systems processing must be implemented.

The supporting systems landscape for payments processing within a bank is typically highly complex. In the context of payment systems improvements, such complexity is a barrier for a bank in meeting the demands of the business environment. A complex payment systems landscape means:

- the time to market of changes is adversely affected; and
- costs to implement changes are very high.

A number of target architecture end states have previously been identified¹ that result in a vastly simplified IT estate. By achieving one of these target states, a bank is then much better positioned to respond to events in the business environment.

A problem arises in how best to undertake the migration of the IT estate from the current state to the chosen target end state via a ‘roadmap’.

This paper explores alternative approaches to payment systems planning. The proposed strategies highlight an incremental planning approach for advancing the payment systems landscape from the current ‘As Is’ architecture to the chosen target ‘To Be’ architecture.

The starting point for the assessment of the strategies is assumed to be:

- a complex payment systems landscape; and
- a relevant target architecture has been identified.

The payments systems planning problem

Figure 1 shows the Integration Space² for payments processing defined in terms of the set of high-level architectural components and their inter-connections. These coarse-grained solution components are termed Architectural Building Blocks (ABBs),³ and an overview of each is provided below.

Payment Gateway

This represents the component that interfaces to a payments scheme. All communication to and from a scheme is typically done via a dedicated Gateway. It also represents components that interact with business partners, for example for the purposes of processing payments on their behalf. Gateways support business to business (B2B) interactions.

Customer Channel

This component represents a channel system through which a customer interacts with a bank. It may also equate to a front/back office system used by internal bank staff to perform payments on behalf of a customer or for the bank itself. Channel systems typically initiate a payment instruction, but they are not recipients of them. Channel systems support bank to customer interactions.

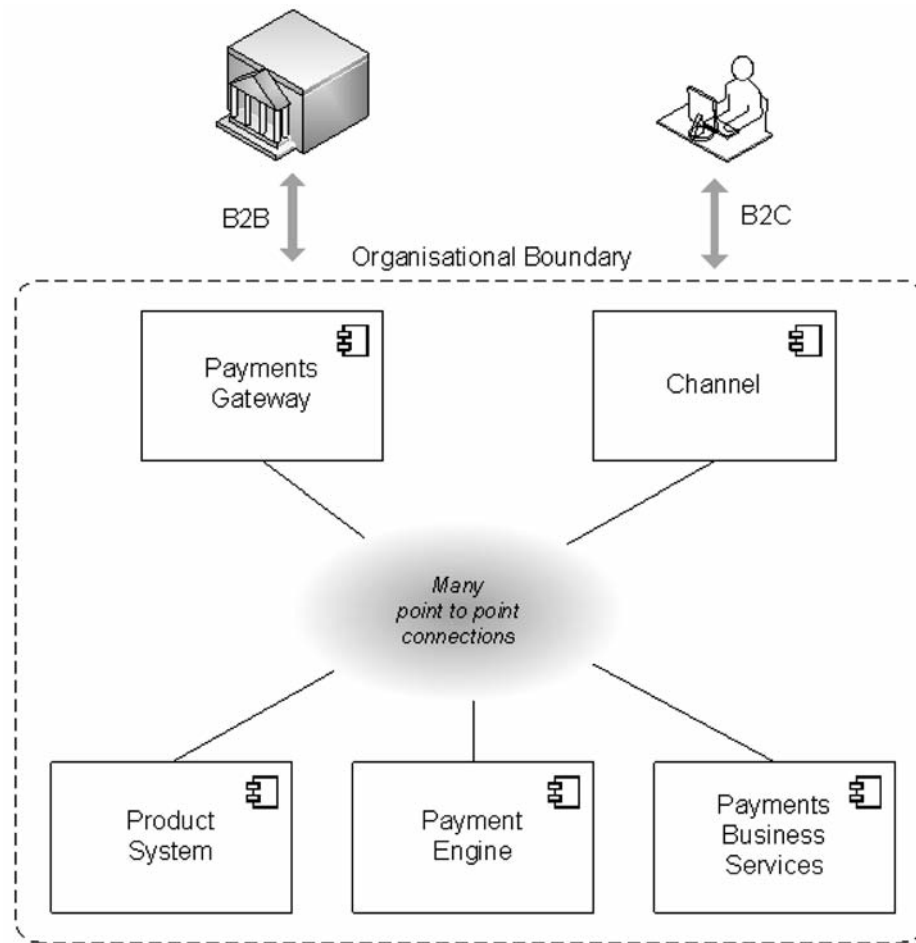
Product System

This component represents a core banking system, that domiciles the customer accounts that are the source and beneficiary of payments. Product systems subsume a variety of payments-processing services. In a simplified IT architecture, an objective is to replace those embedded services with discrete, shared, business services, described below.

Payment Engine

This component performs the activities

Figure 1
Payments
Integration Space



required to process a payment, this being the set discrete processing steps in its value chain⁴ within the organisation. The Payment Engine is invoked:

- upon receiving an inbound payment instruction from the scheme via a payments gateway; and
- upon initiation of an outbound payment by a channel or product system.

Payment Business Services

Payment Business Services represent IT components that provide discrete functional services specific to payments processing. A number of such services are

required and a capability model for payments processing has been defined previously,⁵ reproduced here for convenience.

The set of Payment Business Services is illustrated in Figure 2, and an overview of each provided in Table 1.

The service descriptions include an assessment of the relevance of a service to 'front-end' and 'back-end' processing. Gateways and Customer Channels are considered front-end components as they interact directly with customers and business partners, whereas Payment Engines and Product Systems are considered back-end components.

The classification accounts for whether:

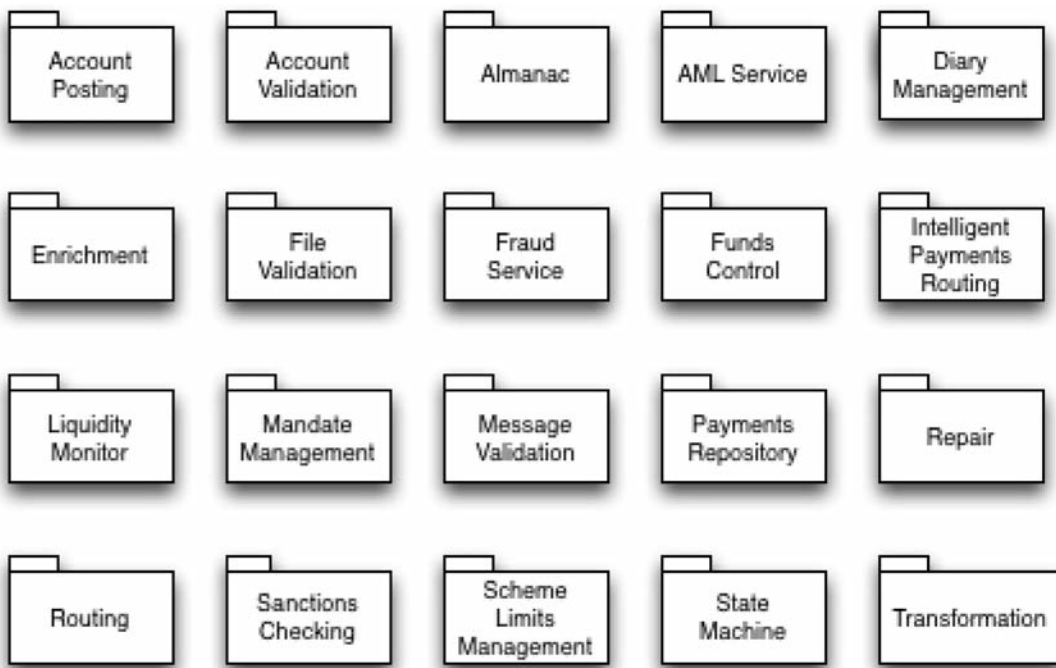


Figure 2 Payment Business Services

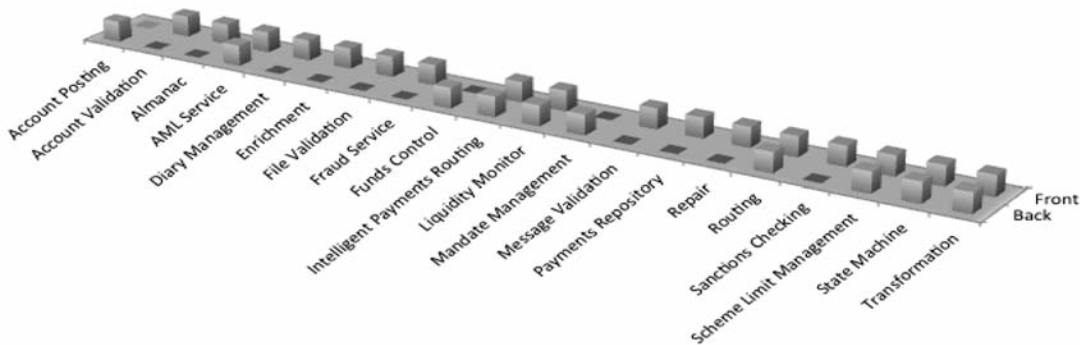


Figure 3 Summary of service classification

- the service provider/consumer is a front-/back-end system;
- the service is triggered by events in the front-end or back-end components; and
- the service is more relevant for front-end or back-end processing.

This classification is relevant for the subsequent discussion of the proposed strategies and is summarised in Figure 3.

Objectives of payment systems planning

Payments systems planning aims to achieve the following IT architecture improvements:

- reduced integration space complexity;
- shared functional business services;
- shared integration services; and
- consolidation, where possible, of the

Table 1: Service capabilities overview

<i>Service</i>	<i>Overview</i>	<i>Classification</i>
Account Posting	Account Posting provides a common service for account updates relating to payment debits and credits. Its purpose is to abstract complexity across multiple product and accounting systems. In the ideal target architecture, the Payment Engine should invoke this service. In some cases, existing legacy channel systems may be configured to invoke account posting directly on payment initiation. In general, however, it is not a service that should be called from the channel systems directly. In a systems improvement initiative, Payment Engines should be re-engineered to call the Account Posting service. Account Posting is a service provided by the Product Systems and is therefore classified as a back-end service.	Back
Account Validation	Account Validation refers to the validation of the beneficiary accounts specified in a payments instruction. This service is required upon creation of outbound payment instructions by customers within the channel systems. In these circumstances, the service is used to validate beneficiary accounts. It also may be invoked to validate beneficiary accounts defined within inbound payments instructions prior to performing account posting. This validation can take place when the payment is received from a Gateway. In a systems improvement initiative, channel systems should be re-engineered to call the Account Validation service. This service is relevant to front-end processing and is therefore classified as a front-end service.	Front
Almanac	The Almanac represents reference data relating to scheme operating times and dates. When used in conjunction with the Diary Management service, it provides alerts relating to scheduled bulk payments. This service is relevant to the scheme Gateways as it is the point at which an inbound bulk file will be received or at which an outbound bulk file will be marshalled for transfer to the scheme. It is therefore classified as a front-end service.	Front
Anti-Money Laundering	The Anti-Money Laundering (AML) service is also a complex service, fulfilling regulatory requirements relating to anti-money laundering. The AML service is typically required to monitor inbound and outbound payments. In this respect, it can be considered a front-end service. In circumstances where the remitter and beneficiary accounts are domiciled in the same bank ('on us' payments), these also require monitoring for AML purposes. The AML service must also be invoked in this context and therefore is also relevant in a back-end processing context. This service is classified as a shared service.	Shared
Diary Management	The Diary Management service is closely coupled to the Almanac Service. Given the metadata defined in the Almanac, the Diary Management capability uses these data to determine the specific diarised events on a given day. It is therefore considered a front-end service as per the Almanac.	Front
Enrichment	Enrichment relates to the enhancement of information in the payment instruction. Enrichment may be required for both inbound and outbound payments. For outbound payments the enrichment typically takes place when bank or scheme specific data must be appended to the payment (eg correspondent bank details). It may also take place for inbound payments, required to be appended with additional information in order to be processed within the bank eg collection accounts mapping to a real account. Since the service relates to payments received at the Gateways or initiated by the Channels, it is classified as a front-end service.	Front

Table 1: Service capabilities overview (continued)

<i>Service</i>	<i>Overview</i>	<i>Classification</i>
File Validation	File Validation provides the capability to perform validation of bulk payment files. Files are validated in terms of structural integrity and business content. Bulk files are received from the payments schemes via the Customer Gateways. Outbound validation of outbound bulk files created by the banks' systems is not usually necessary, since the systems should be designed and tested to produce correctly formatted files. In this respect, this service is classified as front-end.	Front
Fraud Service	As per the AML service, the Fraud Service is generally complex, fulfilling regulatory requirements relating to fraud detection. The Fraud Service must be invoked upon receiving inbound payments from an external remitter and so is relevant for front-end processing. Similarly, for 'on us' payments, the Fraud service must also be invoked and is therefore relevant as a back-end service also.	Shared
Funds Control	The Funds Control service is used to check if there is sufficient credit in a remitter's account to fund a payment. This service is a function of the customers' account balance, domiciled on a Product System, and is therefore classified as a back-end service.	Back
Intelligent Payments Routing	Intelligent Payments Routing (or Method of Payment [MoP]) is the capability to select the most suitable payments scheme for affecting a payment, given a broad set of characteristics provided by a customer. This is relevant in the context of: • <i>ad hoc</i> payments initiated from the Customer Channels; and • a value dated payment being initiated from the Payments Repository upon reaching their value date. These are considered front-end components, and this service therefore has front-end processing relevance. Periodic outbound payments being initiated by a back-end Mandate Management service may also require a decision as to which scheme to use. Thus, this service is relevant in both front-end and back-end contexts and is thus classified as shared.	Shared
Liquidity Monitor	The Liquidity Monitor service captures information relating to the scheme settlement position. To do this, the service requires visibility of all inbound and outbound payments. Introducing this service between a Hub and the Gateways is guaranteed to achieve visibility of all payments and so a front-end integration point is optimal for the service. The service is therefore classified as a front-end service.	Shared
Mandate Management	Mandate Management refers to the creation and management of payment mandates, these being periodic payments instructions. Data for this service are: • received at the Gateways from the scheme for automated debit mandates; and • channel systems that create and manage standing orders and direct debits. The mandate service is a source of payment instructions created on a daily basis according to a defined schedule. The mandate capability is often provided in the existing legacy Product Systems. The service is therefore modelled here as a back-end service, as the impact of providing a discrete service is typically to the back-end Product Systems.	Back
Message Validation	Message Validation refers to the checking of the structural integrity of a payment instruction and its business content. This service is relevant in the context of Customer Channels that initiate <i>ad hoc</i> outbound payments and for Gateways that receive inbound payment instructions. This is classified as a front-end service.	Front
Payments Repository	The Payments Repository service refers to the storage of <i>ad hoc</i> future dated payments initiated from the channels. When their value date is reached, a payment instruction is then created and payment initiation completed.	Front

Table 1: Service capabilities overview (continued)

<i>Service</i>	<i>Overview</i>	<i>Classification</i>
Repair	This service relates to deferred payments from the channels and is therefore classified as a front-end service. The Repair service is invoked for inbound payments. It may also be used to repair outbound payments initiated by customers. It is therefore classified as a front-end service.	Front
Routing	Routing refers to the selection and transmission of a payment to a specific system destination. This service is required to: • route inbound payments to the appropriate Payment Engine; • route payments to the appropriate Product System; and • route outbound payment initiated by the Customer Channels. The above circumstances highlight the need for both front-end and back-end routing and so this service is classified as a shared service.	Shared
Sanctions Checking	Sanctions Checking relates to a specific form of financial crime service to block payments to or from certain individuals or organisations named on watch lists. Sanctions Checking is performed as payments are received or leave the bank. This service is classified as a front-end service.	Shared
Scheme Limit Management	Scheme Limit Management refers to managing the bank's risk exposure to the payments scheme. Limits can apply to individual payments and to the total net value. Above a scheme limit, payments submitted to the scheme may be rejected. The management service provides alerting and withholding capabilities such that the bank's exposure to the scheme can be proactively managed. A service validating payments against the scheme limits is useful in the context of a customer initiating an outbound payment. In this respect it has front-end relevance. The service also has relevance in the case of mandate processing, typically implemented in back-end systems. Since the back-end system is effectively initiating the payment, scheme limits may need to be applied at source. In this context, this service is then relevant to back-end processing. It is therefore classified as a shared service.	Shared
State Machine	The State Machine is a technical capability that enables the status of a payment to be stored as it progresses through a succession of processing activities. The state machine service is required when 'straight through' processing of payment is interrupted. This can happen for several reasons: • the payment is withheld for further investigation by AML checking; • the payment instruction fails validation and requires manual intervention to repair it; and • the payment fails a fraud check, for example due to it being from a stolen 'hot' debit or credit card. Therefore, the introduction of 'real-time' financial crime services (specifically AML or fraud) and payment validation and repair services require the introduction of the State Machine service as these services may pause systems processing. Since the aforementioned services are classified as either front-end or shared, the State Machine service is also classified as a shared service.	Shared
Transformation	Transformation refers to the capability to transform the payments instruction from an industry standard format to a bank internal format (or vice versa). It is required upon receiving an inbound payment instruction from a Gateway. It may also be needed to transform internally formatted outbound payments: • <i>ad hoc</i> payments created by customers in the Channel components; and • bulk file payments created by the Product Systems. Similarly, transformation is required for transforming outbound payments originating from a legacy system and for inbound payments destined for a legacy Product System. It is evident that this service is used in both front and back-end contexts and is therefore classified as shared.	Shared

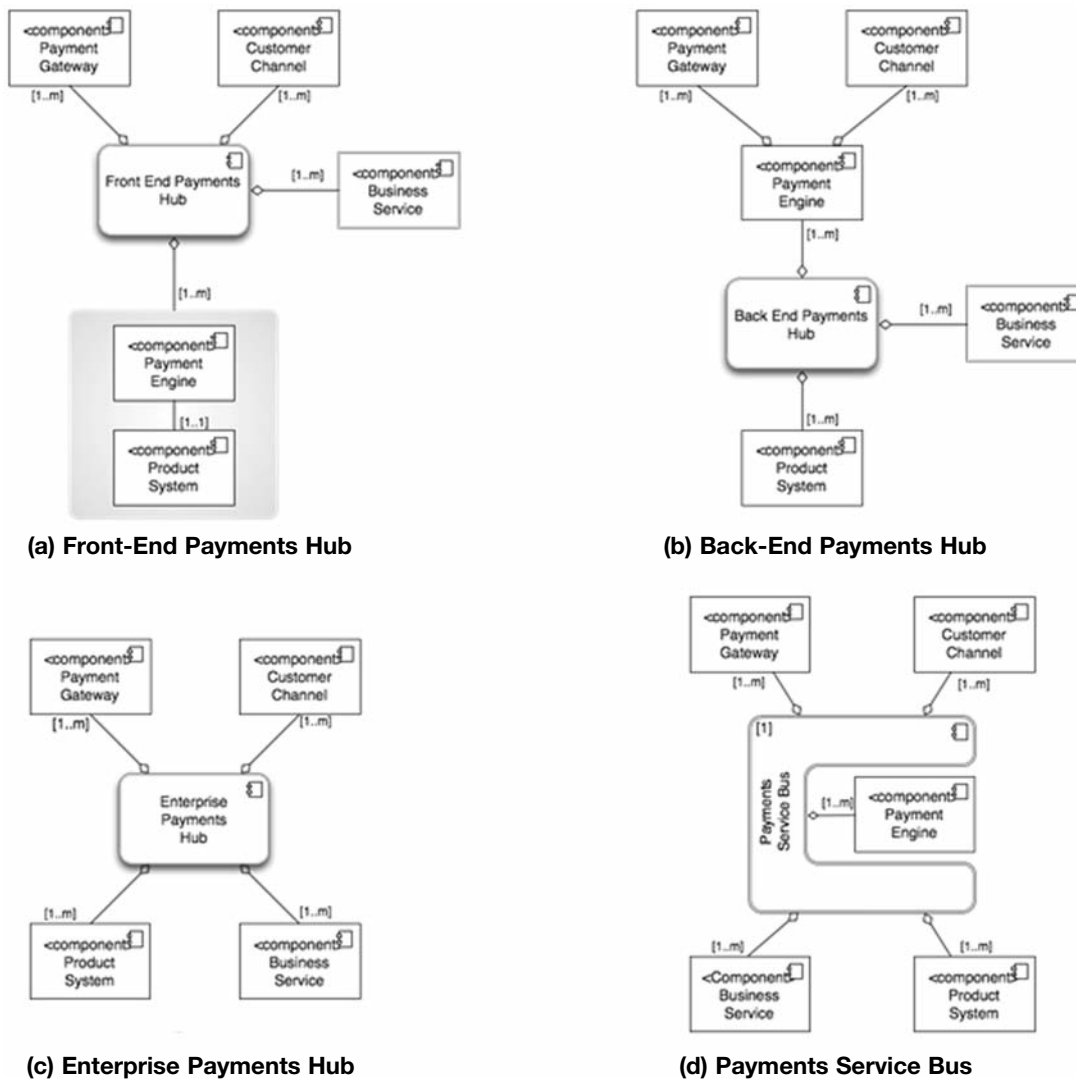


Figure 4 Target architecture patterns

technologies used to implement the key payments ABBs.

The possible target end states are summarised below.

Target end states

Figure 4 illustrates the different target architecture end states that have been suggested, defined in terms of the high-level system components identified.

An overview of each architecture is provided below.

Front-End Payments Hub

This architecture addresses the ‘front-end’ integration problem. It introduces a new integration construct — a Front-End Hub — that connects the channel components to the existing Payment Engines. The back-end integration problem is not addressed with this pattern. Business serv-

ices are provided by the Hub or components external to the Hub.

Back-End Payments Hub

This architecture addresses the ‘back-end’ integration problem. It introduces a new integration construct — a Back-End Hub — that connects the Payment Engine to the Product Systems. The front-end integration problem is not addressed with this pattern. Business services are provided by the Hub itself or components external to the Hub.

Payments Service Bus

In this architecture, a new integration construct is introduced that integrates both front-end and back-end components. The concept of discrete Payment Engines is retained. In this architecture, the Payment Engine provides the payments process orchestration capability and the Service Bus provides the integration capability around the Engine. The approach allows for leveraging the bank’s existing Payment Engines or for introducing replacement Engines if required.

Enterprise Payments Hub

In this architecture, as per the Payments Service Bus pattern, a new integration construct is introduced that integrates both front-end and back-end components. Unlike the Bus pattern, however, the Payment Engine capability is subsumed into the Hub. The Enterprise Payments Hub thus provides both the payments process capability and the integration capability around the Engine. The pattern therefore prescribes the replacement of the existing legacy engines with a consolidated capability provided by the new Hub.

Interim Transition States

The Front and Back-End Hub patterns may be considered as end-state solutions or as interim states to achieving one of

either the Enterprise Payments Hub or Payments Service Bus end-state solutions. The set of transition states is shown in Figure 5.

It is seen that the role of a migration strategy is to accomplish transition from the start state to one of the chosen end states. This is the essence of the payments planning problem.

The Payments System Planning Paradox

Figure 5 highlights the possible target architectures with associated transition states. The integration space is significantly simplified once the migration to the chosen end state is completed. In achieving the end state, the payments IT landscape is migrated from a position of high integration complexity to that of a low one. To achieve such a reduction in complexity, however, involves:

- the introduction of a new integration construct;
- the disconnection of established interfaces between the identified ABBs;
- reconnecting the ABBs via the new Bus or Hub component; and
- potentially replacing selected or all of the Payment Engines.

These are considered highly complex systems integration activities. This complexity is compounded by the fact that many of the interfaces are often likely to be undocumented. The migration activity is therefore a significant undertaking. Therein lies the paradox — to achieve simplification, one has to undertake a programme of huge complexity, fundamentally re-engineering the bank’s payments-processing systems.

Given the nature of the criticality of payment systems processing, undertaking such a migration to a new architecture must be achieved in a phased manner, de-

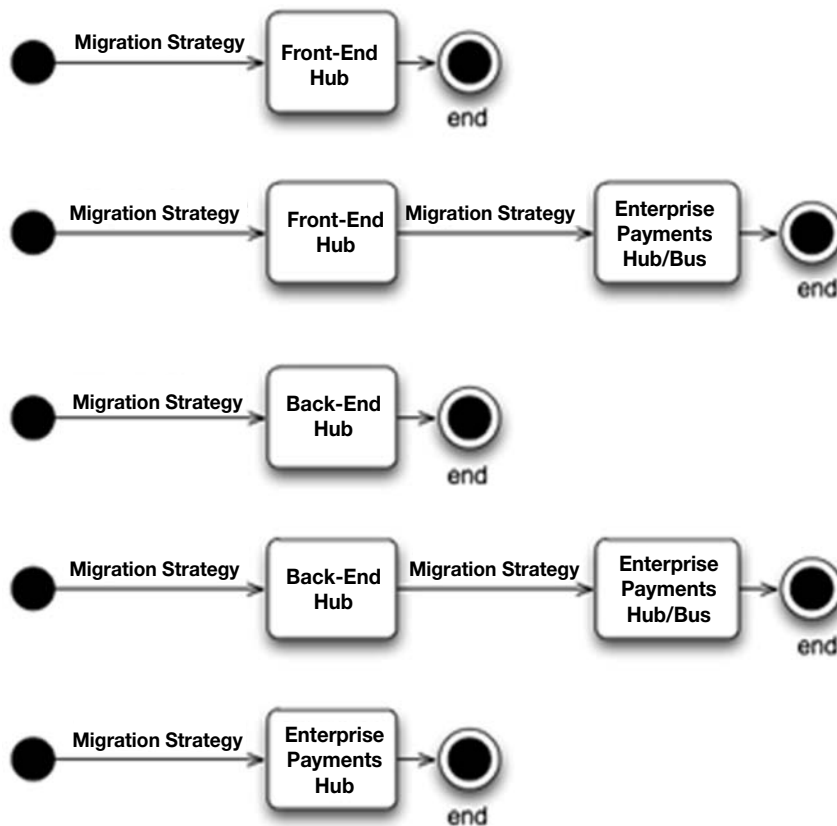


Figure 5
Candidate target
end states

risking delivery and ensuring the bank's payments-processing capability is not compromised. A problem arises in determining the most suitable strategy for migrating to a chosen target architecture.

A structured approach to the planning problem is considered to be essential, ensuring that systems improvements are affected in a way that:

- de-risks the IT deliveries associated with each stage of the roadmap;
- maximises the chance of success of the payment systems improvements initiatives;
- minimises complexity associated with each stage of the roadmap;
- minimises activity required to achieve the target; and

- ultimately ensures that the business benefits of modernised payment systems are realised.

Several payment systems planning strategies are now identified and assessed.

STRATEGY DESCRIPTIONS

Macro strategies

A macro strategy defines an overall approach to migrating from the current systems landscape to the chosen target architecture. In each of these strategies, it is always assumed that a new payments integration construct is to be introduced, this being one of the forms of Payments Hub. The strategies also identify the potential

for introducing shared components each providing a Business Service.

The strategies are:

- (i) Channel by Channel Migration;
- (ii) Back-End Modernisation; and
- (iii) Scheme-by-Scheme Migration.

A key objective of the payment systems improvement is to introduce the shared business services defined by the capability model. This section explores the extent to which new business services can be introduced during the execution of the chosen strategy. In this respect, the analysis presented builds on the Payments Hub introduction strategies identified by Hayden *et al.*⁶ and provides greater detail in terms of the functional scope that each target architecture and migration strategy can achieve.

The target end-state patterns identified address the front-end integration problem, the back-end integration problem or a combination of both. The classification identified in 'Payment Business Services' is therefore used as a key determinant of the relevance of a service to a particular strategy.

Channel by Channel Migration Strategy

Overview. In this strategy, the chosen dimension of migration planning is the channel systems. In this respect, Customer Channels and Gateways are migrated such that they connect with a new Payments Hub. Point to point connections from Channels/Gateways to Payment Engines are replaced with a common integration capability provided by the Hub. The migration is undertaken on a channel component by component basis. As the channels are connected to the new Hub, this provides the opportunity for some of the shared business services to be introduced, replacing the existing legacy services and reducing duplication of functionality.

If the chosen target architecture for payment systems processing is a Front-End Payments Hub, this strategy clearly directly achieves this particular end-state pattern. If the candidate end state is one of the other potential end states, however, specifically Enterprise Payments Hub or Payments Service Bus, this strategy can also be used as a stage in achieving these desired end states. In these circumstances, once the Front-End Hub is fully deployed, the remaining task is then one of integrating back-end systems and components. For this purpose, either the Back-End Modernisation or Scheme-by-Scheme strategies can be employed.

Scope. The Channel by Channel Migration strategy clearly has the potential to fulfil all the front-end services and the shared services, but does not realise any of the back-end services. The scope of the resulting Front-End Hub on execution of the Channel by Channel Migration strategy is shown in Figure 6.

Back-End Modernisation

Overview. In this strategy, the dimension of migration planning is the back-end business services. In this respect:

- New services are introduced that are relevant to the back-end systems processing.
- Payment Engines and Product Systems are migrated such that they connect via a new payments hub.
- Point-to-point connections between Payment Engines and Product Systems are implemented using a shared integration capability provided by the payments hub.

The strategy is focused on provisioning the back-end services, simplifying the back-end systems integration. The introduction of the Account Posting and Funds Control services are key to this strategy.

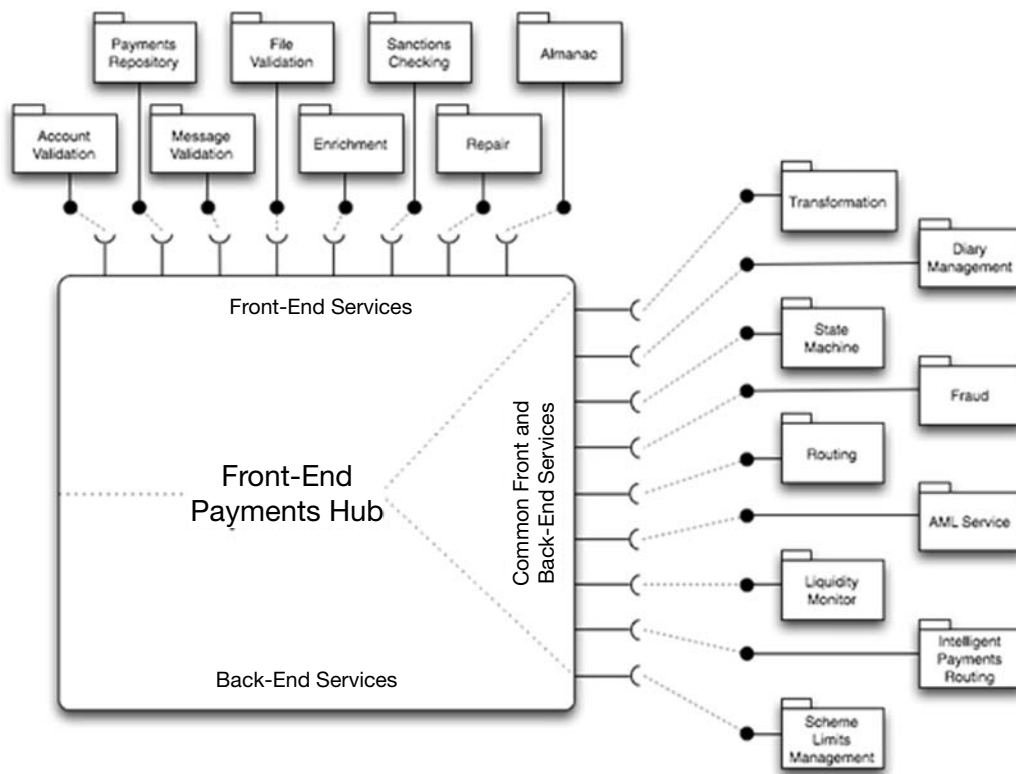


Figure 6
Front-End
Payments Hub
functional scope

These services fulfil the abstraction of the underlying Product Systems from a payments-processing perspective. Implementing these services is considered a major activity, given the underlying complexity of legacy Product Systems.

This strategy provides two sub-options, depending on the chosen end-state architecture:

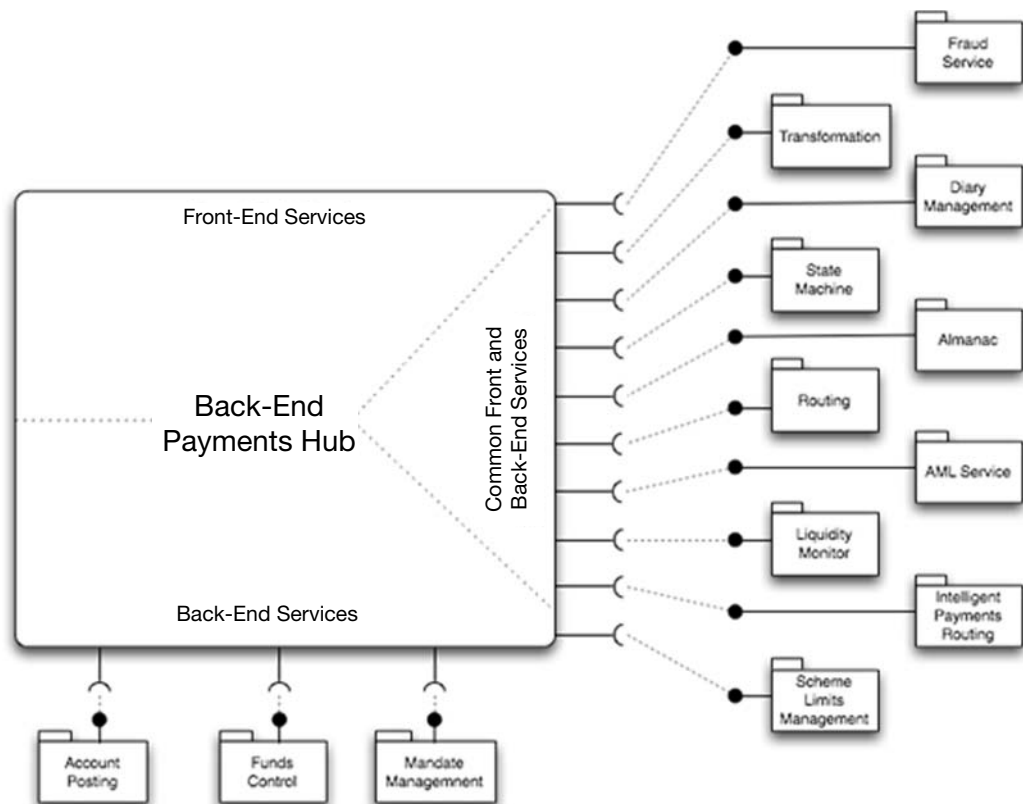
- (i) The existing Payment Engines are retained and integrated with a new Payments Service Bus. This offers the opportunity to leverage existing legacy Payment Engines rather than commit to wholesale replacement of them, and reduces migration activity significantly. This approach allows for the re-engineering or replacement of some of the Payment Engines, decided on a case-by-case basis.
- (ii) The Payment Engines are subsumed

into a Payments Hub, achieving consolidation of all Payment Engines into a single solution.

As the Engines are connected to the new Bus or Hub, this again provides the opportunity for some or all of the identified shared Payment Business Services to be introduced, replacing the existing legacy services and reducing duplication of functionality.

If the chosen target architecture for payment systems processing is a Back-End Payments Hub, this strategy directly achieves this. Again, if the candidate target architecture is one of the other Hub forms, this strategy can also be used as a stage in achieving these desired end states. In these circumstances, once the Back-End Hub has been fully deployed, the remaining task is one of integrating front-end systems and components. For this pur-

Figure 7
Back-End Payments
Hub functional
scope



pose, either the Channel by Channel Migration or Scheme-by-Scheme strategies can be employed.

Scope. The Back-End Modernisation strategy clearly has the potential to implement all the back-end services and the shared services, but not realise any of the front-end services. The scope of services that can be provided by the Back-End Hub on execution of the Back-End Modernisation strategy is illustrated in Figure 7.

Scheme-by-Scheme Migration

Overview. In this strategy, the dimension of migration planning is the payment schemes themselves. Architectural improvements relate to changes to both the front and back-end components that support a specific scheme. These compo-

nents are migrated simultaneously such that they connect with a new Hub or Bus providing the necessary integration capability. In this respect:

- Point to point connections from Payment Engines to Product Systems are provided by the new Hub.
- Point to point connection from Channel and Gateways to the Payment Engines are provided by the new Hub.
- Connections are migrated on a per scheme basis.
- Services are introduced incrementally such that their functionality supports only what is necessary for a given scheme.

The strategy provides the opportunity to provision business services from all the

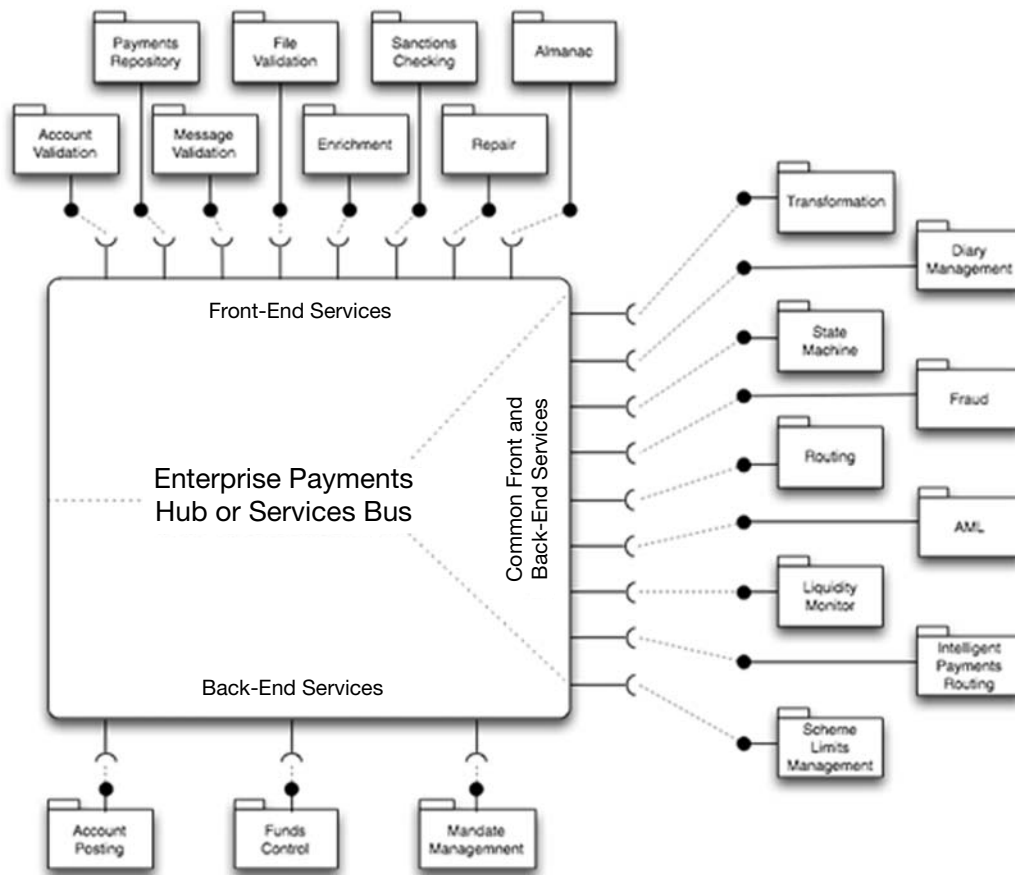


Figure 8
Enterprise
Payments Hub and
Payments Service
Bus functional
scope

defined categories — Front, Back and Shared. Furthermore, it allows for incrementally increasing the breadth and depth of functionality provided by each of the identified business services as the migration for a given scheme is undertaken, eventually resulting in fully specified services.

The sub-options identified above in 'Back-End Modernisation' are also applicable to this strategy.

An illustration of the functional scope of the Hub in the end state is shown in Figure 8.

Micro strategies

Given the business-critical nature of payments processing, the risk associated with payment systems changes is very high. To

mitigate this risk, certain micro strategies can be employed during a particular phase of execution of the overall macro strategy. Three such strategies are identified here:

- (i) Parallel Run;
- (ii) Dual Run; and
- (iii) Cornerstone.

Parallel Run

The Parallel Run micro strategy refers to the simultaneous operation of legacy components and their replacement components, introduced as part of the overall payment systems transformation. The replacement components may be any one of those identified from the Payment Business Services model.

This strategy aims to prove the operation of the new components in the live system, but with a restricted set of customers or transactions. Doing this limits the bank's risk exposure to any problems encountered with the new system components. This is an established approach within the industry when major changes are affected in a bank's payment systems.

This strategy involves routing some payment transactions to legacy systems and some to the new components. A number of ways of selecting which transactions are to be routed to legacy and new systems processing are possible. These are discussed in 'Segmentation approaches' below.

Dual Run

The Dual Run micro strategy, as per Parallel Run above, also refers to the simultaneous operation of legacy components and their replacement components. The difference is that, in Dual Run, both solutions process the same transactions for a period of time.

In the short term, the legacy solution (being the proven solution) remains the production solution. Processing outcomes of the new component(s) are compared with those of the legacy components. Once there is absolute confidence that the processing behaviour of the new components is correct, the new components are switched to become the production solution.

This approach constitutes a more risk-averse approach than the Parallel Run micro strategy.

Cornerstone

The Cornerstone micro strategy relates to an approach to de-risk the introduction of a Payments Hub. The approach is to deploy the Hub as the means of integration between the existing legacy components, but in a functionally minimal form. Recalling the three categories of service

that a Payments Hub can provide,⁷ initially no or limited business capability or process capability is deployed, only integration capability.

The key advantage of this strategy is that it deploys the key component of the payments simplification initiative — the Hub or Bus — early into the IT estate, connecting all the major payment sub-systems. It reduces risk by allowing the performance of the Hub to be monitored and tuned to meet the expected non-functional volumes in advance of increasing its functional scope. Achieving this is a major step in simplifying the payment systems complexity, as components are now all connected using the shared capability provided by the Hub.

Initially, there is no functional replacement of legacy services. In principle, it is then relatively straightforward to enhance functionality incrementally in the Hub, adding new components that provide the business services with accompanying orchestration (process) services if required. Once introduced, a new service is then available in all payments-processing contexts. This can be done with relatively small impacts on the legacy systems, as these are already integrated with the Hub.

Segmentation approaches

Segmentation approaches refer to both business and system techniques for separating payment transactions. In implementing the segmentation approach, a variety of routing rules are possible. Segmentation can be performed on a variety of attributes of the payment, tabulated in Table 2.

STRATEGY EVALUATION

The payments planning problem has been framed thus far as a pure IT problem — that of transitioning to a target architecture across the estate. In practice, however,

Table 2: Payment segmentation dimensions

<i>Segmentation dimension</i>	<i>Implementation</i>
Product	Payments for certain types of products, identified by a combination of sort code and account number.
Account number	Identified as a defined sub-set of account numbers. This may be a simple split by account number range.
Customer	There is no direct reference to a customer in a payments transaction. The relationship to a customer is fulfilled by the account number and is typically held in the Product System. Customer groupings can therefore be represented by determining the account numbers for the set of products they hold.
Brand	May refer to different product brands and be used to route transactions.
Sort codes	A simple split based on a range or table lookup of sort codes.
Split by value/high care/low care	A split based on payment value above/below a threshold.
Currency	For international payments, the currency code may be used to route payments.

any such modernisations must take place in the context of a changing business environment. A good strategy must therefore be able to accommodate internal and external shocks during its execution. In this section, the chosen target architectures and associated migration strategies are evaluated by assessing their ability to accommodate such shocks manifested as business events.

Figure 9 illustrates the variety of types of business events specific to the payments business environment. A brief description of common events is provided, and the impact on the payment sub-systems is described. This illustrates whether the event dictates predominantly a front-end issue, a back-end issue or both. This in turn determines how readily the strategy can accommodate the event and serves as a measure of its robustness.

Business events

Banking Merger

A Banking Merger trigger relates to the acquisition or merger of a bank with another financial institution. In this scenario, customers and products of the

acquired bank are transferred to the acquiring bank. From a payments-processing perspective, payments instructions from the schemes must now be routed to the acquiring bank.

Post-business merger, various states of IT integration can exist.

- *Phase 1 — Consolidation of Payment Gateways.* This is generally the first phase of IT integration. The payments scheme routes payments to a single entity (this being the acquired bank) rather than separately to both the original banks. This requires consolidation of Gateway processing. Upon consolidation, a simple interim solution is then to process payments using each respective bank's payments-processing capability. This requires a routing of payments to separate Payment Engines.
- *Phase 2 — Consolidation of Payment Engines.* A second phase of integration is to consolidate the payments processing. Payments processing for a given scheme is subsumed onto one or other of the acquiring or acquired bank's Payment Engines with the original Product Systems being retained. This activity

Figure 9 Types of business and IT events



creates a back-end integration problem to route to the system domiciling the customer account.

- *Phase 3 — Consolidation of Product Systems.* In the third phase of integration, payments processing and customer accounts are all consolidated onto single systems. Assuming the completion of Phase 2, this is then a straightforward activity to adapt the routing rules as the account migration takes place.

Selection of the preferred consolidated systems can be on a ‘winner takes all’ or a ‘best of breed’ basis.

Summary

Type	Impact
Front End	Ability to route to the original banks’ Payment Engines upon Gateway consolidation.
Back End	Ability to route to original banks’ Product Systems in short to medium term until Product Systems are consolidated.

Banking Divestment

A Banking Divestment trigger relates to a business decision or mandated regulatory requirement to divest part of the bank.

Specific segments of the bank’s products and customer base are transferred to the divested bank.

Should the divested business be acquired, the impact on payments processing from the acquiring bank’s perspective is covered in ‘Banking Merger’ above. From the perspective of the divesting bank, there are two key options for payments processing to consider:

- (i) the divested bank becomes a full scheme member and performs its own associated payments processing; and
- (ii) the divesting bank may act as an Agency Bank for the divested bank, processing payments on its behalf.

The Product System to domicile the divested customer’s accounts may be provided:

- by cloning the divesting bank’s systems;
- by migrating to the acquiring bank’s system; and
- by commissioning new Product Systems.

Assuming an Agency model and continued use of the divesting bank’s Product System to domicile the customer accounts, from the divesting bank’s per-

spective, the IT impact of the divestment requires a logical separation of payments processing and the customers' accounts. Assuming the divested bank provides its own Product System in the long term, maintaining the Agency model will require a new Gateway between the two banks.

If the long-term objective is for the divested bank to become a full scheme member (or if the acquiring bank already is), payments are routed directly by the scheme to the Payments Gateways of the divested bank. In these circumstances, IT impacts relate to the divested or acquiring bank as per 'Banking Merger' above.

Summary

<i>Type</i>	<i>Impact</i>
Front End	Introduction of new B2B Gateway in the event of the adoption of an Agency model.

New Automated Clearing House

A New Automated Clearing House (ACH) trigger relates to the bank becoming a member of a scheme. This may be a new scheme or an existing scheme in which the bank does not currently participate as a direct member. A New ACH is a major event, but a relatively rare occurrence, examples being the introduction of the Faster Payments scheme in the UK and Eurozone SEPA schemes.

In system terms, interaction with a new scheme usually requires the introduction of a Payments Gateway. It will typically also require a new Payment Engine or modification to an existing Engine. Product Systems must accommodate debits and credits associated with inbound/outbound payments via the scheme, affected by the Payment Engine.

Channel systems may also be affected if initiation of the payments through the new

scheme is permitted directly in the channel. The preferable approach, however, is that the responsibility for method of payment (MoP) selection logic is held outside the channel system itself within the Intelligent Payments Routing service. The channel is then isolated from the payments scheme specifics and is not affected by such a trigger. This event provides the opportunity to introduce this architectural approach.

Summary

<i>Type</i>	<i>Impact</i>
Front End	Introduction of a new Gateway and integration with the associated Payment Engine. Other channel connections to support payment initiation unless intelligent MoP is adopted.
Back End	New routing capability to route from the new Payment Engine to the existing Product Systems.

New Customer Channel

The New Customer Channel trigger relates to the introduction of a new channel system within the bank. An example might be the introduction of a mobile banking channel. The new channel system may be required to initiate payments via one or more of the payment schemes. This will require integrations from the channel to each scheme-specific Payment Engine.

As discussed in the 'New ACH' trigger, if the MoP determination is done independently of the channel, this problem is

Summary

<i>Type</i>	<i>Impact</i>
Front End	New integrations from the channel system to multiple Payment Engines in the circumstances of scheme-specific payment initiation in the channel.

lessened, as the channel needs to integrate only with the component providing the MoP.

Regulatory Change

A Regulatory Change trigger relates to a mandated capability that must be incorporated into a bank’s payments processing. Regulatory Change triggers are very diverse and are not easy to predict. Typically, regulatory triggers fall into one of two categories:

- (i) the introduction of watch lists against which transactions are monitored and reported; a recent example of this category of Regulatory Change trigger is Foreign Account Tax Compliance Act (FATCA)⁸; and
- (ii) a requirement to report on liquidity or other financial attributes of the bank.

From an IT perspective, both these categories require monitoring of payment events and the introduction of logic to perform data capture and matching. Having a Payments Service Bus or Hub already in place makes such monitoring relatively easy, since this component will have visibility of all payments. Similarly, implementing a payments ‘operational data store’ is useful to capture payments transactions and constitutes a convenient data source upon which to perform necessary matching and reporting.

Summary	
Type	Impact
Front End	Capture of payment events.
Back End	New components to store data and perform matching and reporting.

IT Events

The following events relate to IT drivers to modernise payments-processing sys-

tems. These typically arise through:

- the emergence of new technologies;
- a desire to migrate from legacy systems to more flexible packaged solutions; and
- a need to migrate from applications and platforms that are no longer supported by a vendor and therefore present an operational risk to an organisation.

Core Banking Platform Replacement

Core Banking Platform Replacement refers to the introduction of a new Product System to replace the existing banking platform. Given the high business risk associated with the activity, this scenario demands a phased introduction of the new banking platform with customer segments being migrated gradually from the legacy platform to the new banking platform.

In system terms, legacy banking platforms and new platforms must operate concurrently as the migration takes place. Thus, from a payments-processing perspective, inbound payments must now be routed to one or other of the banking platforms. Similarly, outbound payments initiated from Customer Channels require debits to be posted to the Product System domiciling the customer account.

Summary	
Type	Impact
Back End	New integrations from multiple Payment Engines to Product Systems. Routing to legacy and new Product Systems.

Payment Engine Replacement

This trigger relates to the introduction of a replacement Payment Engine in the IT estate. As with core banking replacement, there is high business risk associated with the activity, and this scenario also demands

a phased introduction of the new Payment Engine, using one of the defined micro strategies.

Similarly, the legacy Engine and new Engine must be supported concurrently as the migration takes place. Thus inbound payments from a given source must now be routed to one or other of the Engines.

Summary

<i>Type</i>	<i>Impact</i>
Front End	New routing capability to route to respective legacy and new Payment Engines systems.

Industry standards refresh

This trigger relates to changes to industry standards to which a bank must respond in order to continue to interface with the scheme. The change usually infers enhancements to data transmitted to and from the scheme. The changes may involve the introduction of essential new data items and optional ones.

Using the data has an impact on the Payment Engine that must interpret the data for inbound payments or populate it for outbound payments. If the data are not business critical, the bank may elect not to process the new data. In these circumstances, data can be transformed at the Gateway to existing canonical formats employed internally within the bank. The Payments Hub/Bus should perform this

Summary

<i>Type</i>	<i>Impact</i>
Front End	Transformation of the industry data into an internal canonical format. Impact to downstream processing, primarily the Payment Engine processing if the new data is business critical.

transformation function. Adopting this approach does not affect downstream payments-processing components and is effective in quickly aligning changes to industry standards.

Strategy robustness analysis

A summary of front-end and back-end impacts of the set of events is shown in Figure 10. The majority of events have a front-end impact. This suggests that a Front-End Hub solution can readily accommodate the impact of the majority of the events. Fewer than half of the event types identified have a back-end impact. This infers that a Back-End Hub solution is less able to accommodate the impact, given the range of events that can occur. It should be noted, however, that some events have both front-end and back-end impacts, and so these particular events are less easily accommodated by either of the Front-End or Back-End Hub solutions.

Target architectures of Enterprise Payments Hub or Payments Service Bus can accommodate both front-end and back-end impacts within their scope.

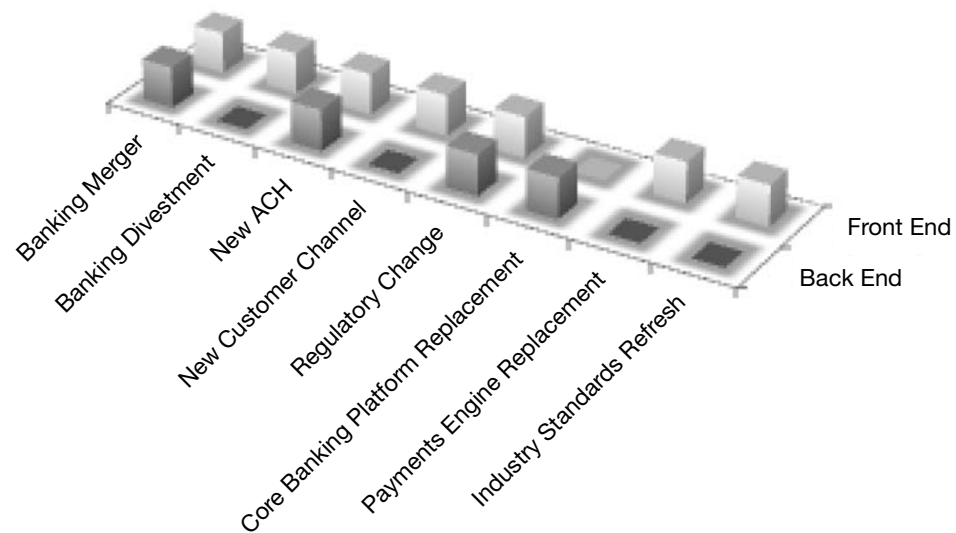
Similarly, if one is adopting a front-end migration strategy, it is likely that this can more readily accommodate impacts to the planning and execution than a back-end migration strategy can. A Scheme-by-Scheme migration will already address both front-end and back-end impacts within its scope and so is considered more robust to the range of business events than either the front-end or back-end strategies.

CASE STUDIES

Major UK retail bank

A major UK retail bank with a global presence embarked on a programme to simplify its payments processing. Over 200 separate sub-systems were to be consoli-

Figure 10
Payments
Integration Space



dated, and a common integration solution was to be adopted. The business and IT operations were primarily UK focused, but there was also significant European and North American presence. This inferred substantial UK, Eurozone and cross-border transactions.

The target architecture selected was a Payment Service Bus, the primary objective being a reduction in the Payments Integration Space complexity with its associated business benefits. This approach leveraged the existing Payment Engines, but did not preclude targeted replacement of some legacy engines in the future.

Strategies adopted include:

- *Channel by Channel Migration.* The overarching strategy was to introduce the Bus in the first instance to connect the existing channel systems and Payment Engines, replacing the existing point-to-point integrations.
- *Cornerstone Strategy.* This strategy complemented the Channel by Channel Migration strategy. Initially, integrations from Channels and Gateways to the legacy Payment Engines were to be completed. In doing this, there were to

be no functional impacts to these components. New business services were then to be added incrementally, orchestrated by the Bus. Corresponding legacy services were decommissioned in a controlled manner.

- *Back-End Modernisation.* On completion of the Channel by Channel Migration, new back-end services were to be introduced. Key to this strategy was the development of services to abstract the interface to three Product Systems and provide a sophisticated Account Posting service.

Shocks to the business environment observed during the period of the programme were:

- *Regulatory Change 1.* Introduction of mandatory US regulation, specifically FATCA. This regulation required the monitoring of financial transactions relating to a specific customer segment and reporting on their activity. Above a certain threshold value of payment, it also required the withholding of a certain percentage of the payment's value.

In systems terms, a desirable strategic

solution was to store payments such that analysis of the transactions for the specific customer segment could be performed. This in turn inferred that the Bus was required to have visibility of the relevant financial transactions and then employ a new service to store them in the short to medium term for analysis.

- *Regulatory Change 2.* Introduction of mandatory regulation relating to account switching.⁹ This regulation required the identification of payments for accounts that were closed up to one year previously and the re-routing of credits to a defined switched account. In system terms, the impact was to monitor payments and match payments against a set of known switched accounts.
- *Industry Standards Refresh.* This event represented the annual update to the SWIFT standards.

In executing the Front-End Migration strategy, it was apparent that there was significant complexity to even achieve the Cornerstone strategy. This was because legacy systems themselves were old and informally documented. It required a significant concentration of effort to analyse and scope the Payments Service Bus for a single Gateway connection.

Even prior to completion of the channel migration, there were immediate and significant demands to increase the functionality of the Bus and provide services for operational storage and to perform analytics on the payment transactions to support the FATCA requirements. So, while this requirement could be considered a front-end impact and could be accommodated within the chosen strategy, in practice it required much problem analysis and created many cross-programme dependencies. It was difficult to accommodate these within the timescales of the regulatory requirement.

The regulatory change relating to account switching inferred a back-end impact to mark switched accounts and trigger specific processing when payments to these accounts were attempted. Addressing this requirement inferred bringing forward solutions relating to back-end systems processing. This again detracted from the ability to execute the initial front-end migration strategy.

Finally, the standards refresh should have been an event that was readily accommodated within the initial phase of the strategy, given it had a front-end impact. Achieving this, however, was dependent on the early fulfilment of the Cornerstone strategy, such that the Bus with its associated transformation capability was available. This approach could have been used to buffer downstream systems from changes. Timescales for the execution of the Front-End Migration strategy were affected by the initial complexity and implementation of solutions to accommodate the regulatory requirements. This meant that this event was unable to be accommodated using the new strategic solution.

In summary, the ambitious scope of the Payments Service Bus and the significant impact of the business events that occurred meant that the modernisation programme was in a continual state of planning and problem solving. The Front-End Migration strategy was seen as a 'quick win' to deploy the Bus and de-risk the technology solution. In practice, even the first scheduled phase of a single Gateway integration using the Bus proved to be a large undertaking. It is too early to comment on the overall success of the programme, and this was ongoing as of late 2012.

UK high street bank

A UK retail bank embarked on a programme to replace its legacy products system with a new core banking platform.

The bank was a member of all UK payments schemes and a major provider of wholesale payments for corporates and banking partners. The bank had limited international payment transactions.

An Enterprise Payments Hub pattern was identified as the target end state, broadly occupying a mid-spectrum position in terms of its required functionality.¹⁰ A roadmap was defined to introduce the new core banking platform and Payments Hub, this being delivered over a period of several years in three major phases of work.

Strategies adopted included:

- *Scheme-by-Scheme macro strategy.* The overarching strategy was to introduce the Hub on a per scheme basis.
- *Product Segmentation.* The initial approach was to introduce the new core banking platform to support a specific savings product with payments being undertaken using the UK BACS scheme.
- *Cornerstone Micro Strategy.* A strategy of deploying the Hub early within the IT estate was employed. Since the overall approach was a Scheme-by-Scheme strategy, initially the Hub connected to the BACS Gateway with routing and transformation capabilities to integrate to both legacy and new Product Systems. The initial deployment constituted a Cornerstone strategy as the Hub processed transactions, but all were routed to the original legacy system.
- *Parallel Run.* In the subsequent phase of delivery, the savings products were to be introduced in the new Product System, and BACS transactions only for these products were then to be routed by the Hub to the new Product System. This constituted a Parallel Run, as both legacy and new processing occurred simultaneously, but for a specific segment of the BACS transactions.

Shocks to the business environment observed during the period of the programme:

- *New Channel.* As part of the overall IT modernisation, a replacement internet banking system was to be introduced. The Payments Hub could be employed to integrate this channel to the Product System to support payment initiation.
- *Banking Merger.* The bank acquired another financial institution that sold a restricted range of financial products and had limited payments-processing capability. This merger could be accommodated by subsuming the payments processing, with the bank acting initially as an Agency for the acquired bank.
- *New Product System.* A replacement credit card system was also commissioned as part of the modernisation. The Payments Hub could be employed to undertake integration of this new Product System with card scheme Gateways.

In summary, the strategies adopted were broadly able to de-risk the business programme for the initial planned phase. They also proved robust to significant shocks in the business environment, these being an acquisition and two technology refreshes. It should be noted, however, that the programme is still ongoing, and the original timescales for the work were extended. This is indicative of the complexity of the planning activity and of the significant impact related to the business events that occurred.

CONCLUSION

Achieving modernisation and simplification of payment-systems processing is a hugely complex undertaking. In migrating to a new target systems architecture, a structured approach to the planning is

considered essential. Two types of strategies for guiding the migration have been presented:

- macro strategies, to guide the overall systems planning; and
- micro strategies, to provide localised de-risking of the delivery phases.

The business environment for banking, and payments in particular, is subject to significant shocks, especially in the current climate. The types of business events that can occur have been identified, and their potential impact on payment systems processing has been determined. Some business events infer primarily a front-end impact, whereas others infer a back-end systems impact or both. The robustness of the target architectures and strategies has been evaluated in terms of their ability to accommodate these events.

To provide maximum robustness to shocks to the business environment, banks should adopt either the Enterprise Payments Hub or Payments Service Bus architecture. Both these architectures are considered to provide the greatest robustness, since they can more readily accommodate both front-end and back-end impacts. A Front-End Hub is considered more robust than the Back-End Hub, since the business events are shown to be more likely to affect the front-end processing.

The overall system's robustness is strongly determined by the choice of target architecture, but the choice of migration strategy also plays a role. A Scheme-by-Scheme migration strategy is considered most robust to possible business events, since the planning activity is already addressing improvements to both front-end and back-end processing. A Channel by Channel Migration strategy is considered more robust than a Back-End Modernisation, again because of the more likely impact of business events to front-

end than back-end processing.

Regarding the micro strategies, the case studies highlight a potential drawback of the Cornerstone strategy in that there is a significant amount of integration work in 're-wiring' the existing legacy components. Given the undocumented nature of most legacy systems, there is a risk that the programme becomes entrenched in technical integration work. Several delivery cycles could complete without any significant realisation of business benefit. Furthermore, the approach may incur significant integration effort for a legacy system that is in fact due for replacement. This integration effort would be repeated when the replacement system was introduced.

The experience highlighted by the case study in undertaking a Channel by Channel Migration suggests that known or emerging business events can soon disrupt the strategy execution. If such events begin to dominate the planning and execution, a modified strategy of prioritising and aligning improvement phases to the emerging business events is considered effective. Using this approach, each phase is aligned to the specific requirements of a business event and delivers only the minimal functionality necessary to support that event. Furthermore, legacy components are only decommissioned if they are affected by the particular event.

Adopting this approach means that effort is focused on priority business imperatives. The downside, however, is that even a series of business events is unlikely to affect all the payment sub-systems, and so modernisation of some components would not be triggered. This implies that the target end state of a fully integrated Hub cannot be achieved by this approach alone. This contrasts with the macro strategies that are designed to achieve the complete migration of all components to the desired target architecture.

In conclusion, it is clear that all the highlighted strategies must always be complemented with planning to accommodate the emerging business events. The scale of impact of the business events, their visibility on the planning horizon and often challenging timescales will continue to make the payments planning problem extremely challenging. The structured approaches presented here help address that challenge.

REFERENCES

- (1) Farrow, G. S. D. (2012) 'Patterns for Payment Systems Integration', *Journal of Payments Strategy and Systems*, Vol. 6, No. 1, pp. 15–36.
- (2) *Ibid.*
- (3) The Open Group Architecture Forum (2009) 'TOGAF 9', ISBN 978-90-8753-230-7. Available at: <http://www.opengroup.org/togaf> (accessed 2nd March, 2012).
- (4) Porter, M. E. (2004) 'Competitive Advantage', Free Press, New York.
- (5) Farrow, G. S. D. (2011) 'The Payments Hub Spectrum: A Model for the Design of Payment Hubs', *Journal of Payments Systems and Strategy*, Vol. 5, No. 1, pp. 23–24.
- (6) Hayden, R. Akash, L. Ledford, S. and Nunn, C. (2010) 'Payments Hubs: Redefining the Industry's Infrastructure', *McKinsey Quarterly*, No. 8, pp. 16–21. Available at: http://www.mckinsey.com/client-service/Financial_Services/Knowledge_Highlights/Recent_Reports/~media/Reports/Financial_Services/MoP8_Payments_hubs.ashx (accessed 12th February, 2012).
- (7) Farrow, ref. 5 above.
- (8) Foreign Account Tax Compliance Act 2011. Available at: [http://www.irs.gov/Businesses/Corporations/Foreign-Account-Tax-Compliance-Act-\(FATCA\)](http://www.irs.gov/Businesses/Corporations/Foreign-Account-Tax-Compliance-Act-(FATCA)) (accessed 20th November, 2012).
- (9) Payments Council (2011) 'Accounts Switching', Payments Council, London. Available at: http://www.paymentscouncil.org.uk/current_projects/account_switching/ (accessed 20th November, 2012).
- (10) Farrow, ref. 5 above.