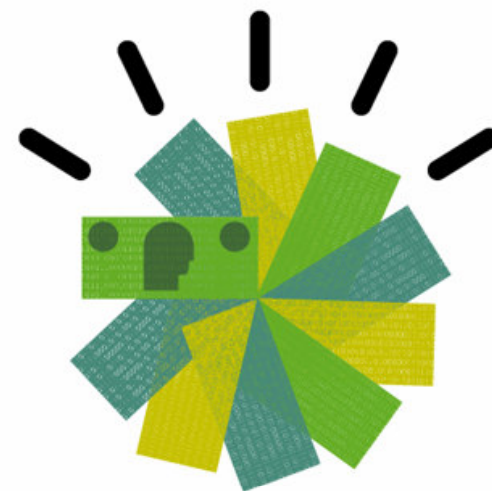


Bank Modernisation and Transformation through Core Banking

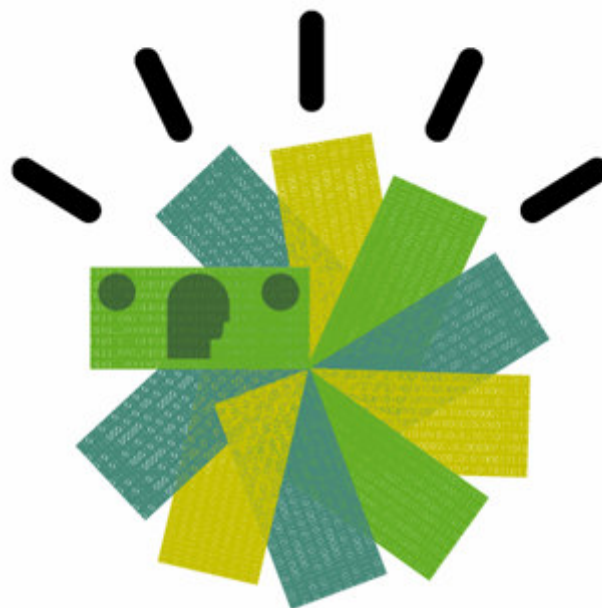


Smart Bank challenges..... And the why of our strategy

“Smart Banks” .. will require to engage into a mix of transforming their operational model , focused investment, and innovation.

This means **entrepreneurialism beyond anything the Banks have ever done before** and dreamed of.

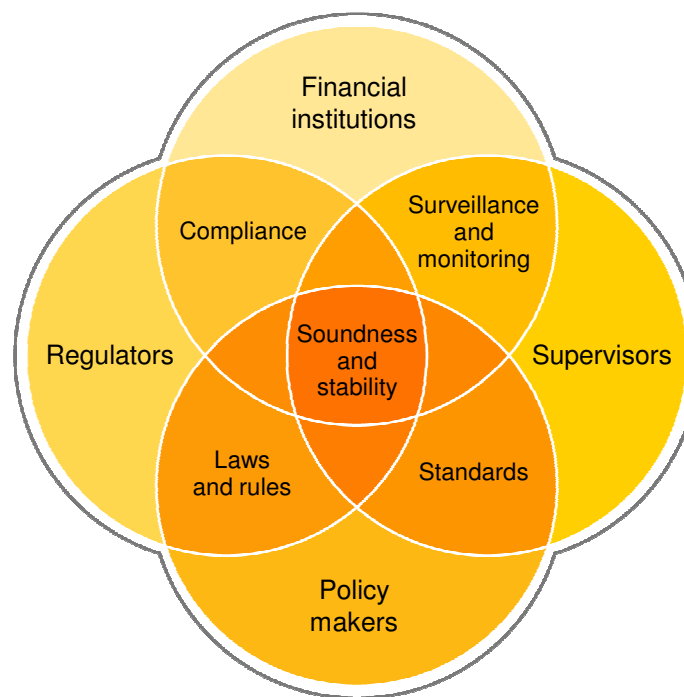
The real question is, which Financial Institution will truly build a bridge to engage this challenges of the 21st century in a way that it will strengthen their ability to **compete in the global economy**, while practicing a **better risk discipline** and at the same time truly and significantly **diminish the overall cost per transaction** so that it is reflected in the overall financial results of the Bank and **better perception from their customers**.



The reality of living in a globally integrated world is upon us.

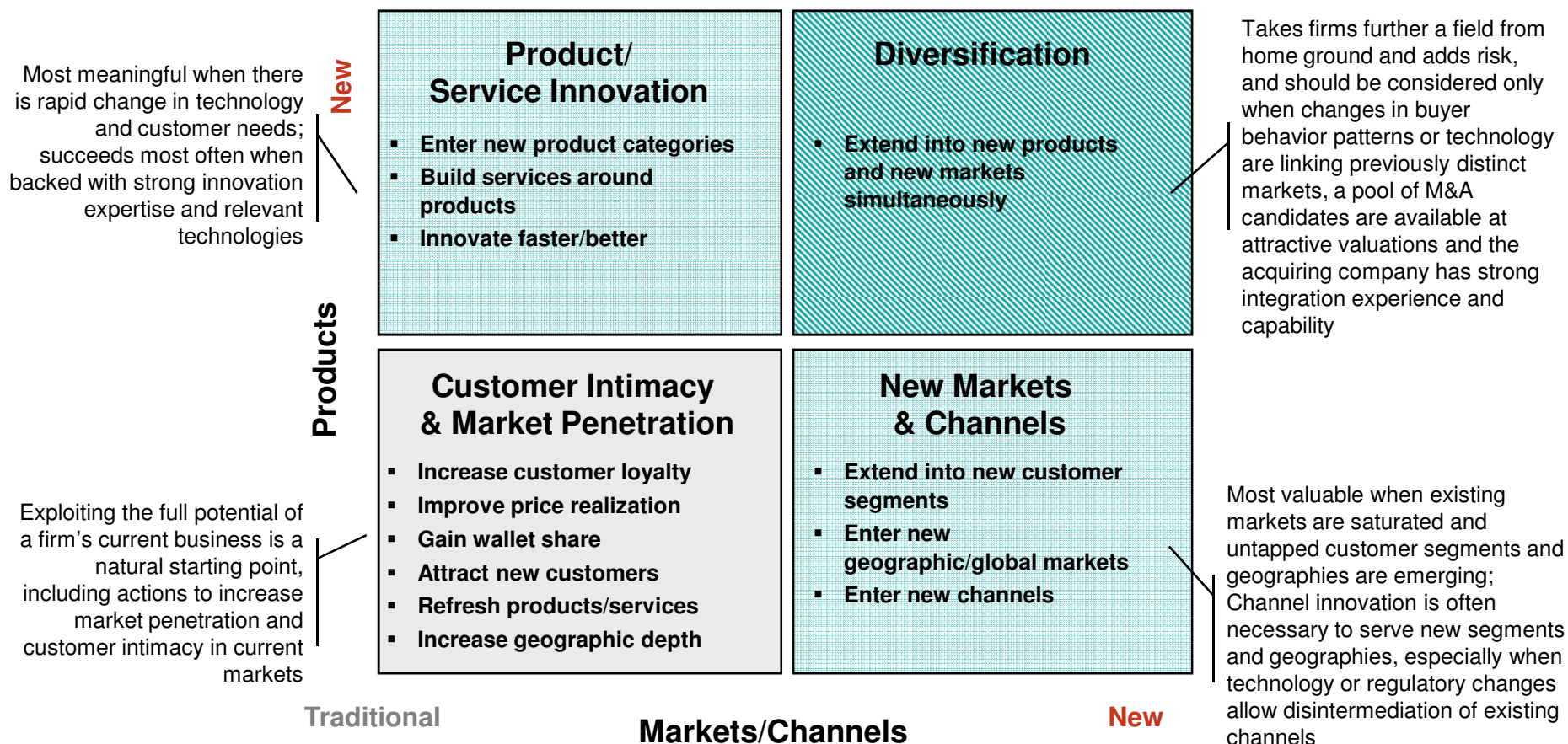
- Sophistication has outstripped our ability to handle it.
- Government and industry are working together to balance stability and innovation.
- Firms must innovate their business models in order to thrive.
- Daily interactions must deliver on brand promises.

The world is connected:
economically, socially
and technically.



Banks have a number of different options to consider as they evaluate which opportunities for innovative growth are right for them

Opportunities for Innovative Growth



Source: Adapted from The Growth Triathlon, IBM Institute for Business Value

How do we focus our strategy in order to help Banks think “Smart”

Growth requires smart institutions to think and act in new ways



Reinvent the business model
In order to Embody Creative Leadership



Drive a simplified and streamlined agile enterprise that balances growth, efficiency and business resiliency



Focus on the Customer
In order to Reinvent Customer Relationships



Develop enterprise wide capabilities and new intelligence to enable informed judgment, client-centricity and profitable growth



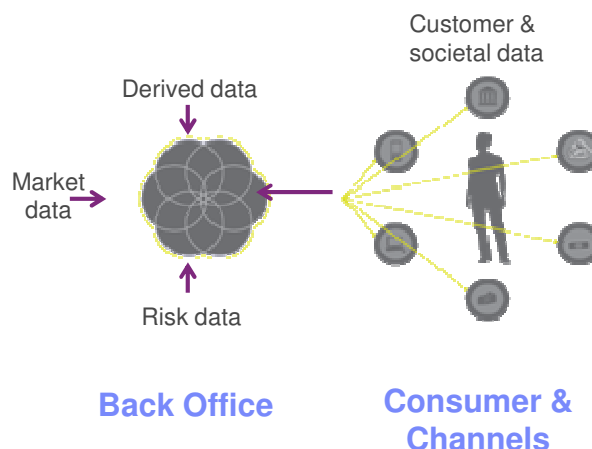
Integrate risk management
In order to Build Operating Dexterity



Achieve compliance objectives while mitigating operational risk, fighting crime and optimizing financial returns

What Banks will be asking for:

- Services future growth from a lower cost base achieved through **intelligent cost reduction**
- Is built around an **optimized and integrated back-front office** in business and technology
- Wraps **tailored products with flexible pricing** to deliver convenient offerings on demand
- Delivers through an **interconnected and ever-evolving** set of channels



- Draws on marketplace insights and engages **customers as co-developers**
- Uses analytics to **transform decision making for a competitive edge**
- **Enables a continuous flow of data** to cut costs, drive speed and innovate
- Integrates a **firm-wide view of risk and compliance in real-time**

The need for Core Banking system transformation is clear....

90%

Of financial services firms say that the **inflexibility of their core banking system is the biggest deterrent** to being innovative in the marketplace.

95%

Of the banks in the world have **product-centric core banking systems**, which is a big obstacle to becoming a customer-centric organization.

93%

Of banks **admit to using more than one application** for even basic banking processes such as opening an account.

....and the opportunities are significant....

90%
faster time to market

Faster Time to Market:

Large Asian Bank

Bank achieved 90% improvement in time to market for new products and product-bundling offers based on dynamic pricing and customer segmentation.

50%
productivity jump

Doing more with less:

Large global bank

Bank achieved a 50% productivity jump after it transformed its core system around customers and broke away from product-centric architecture.

\$640m
saved

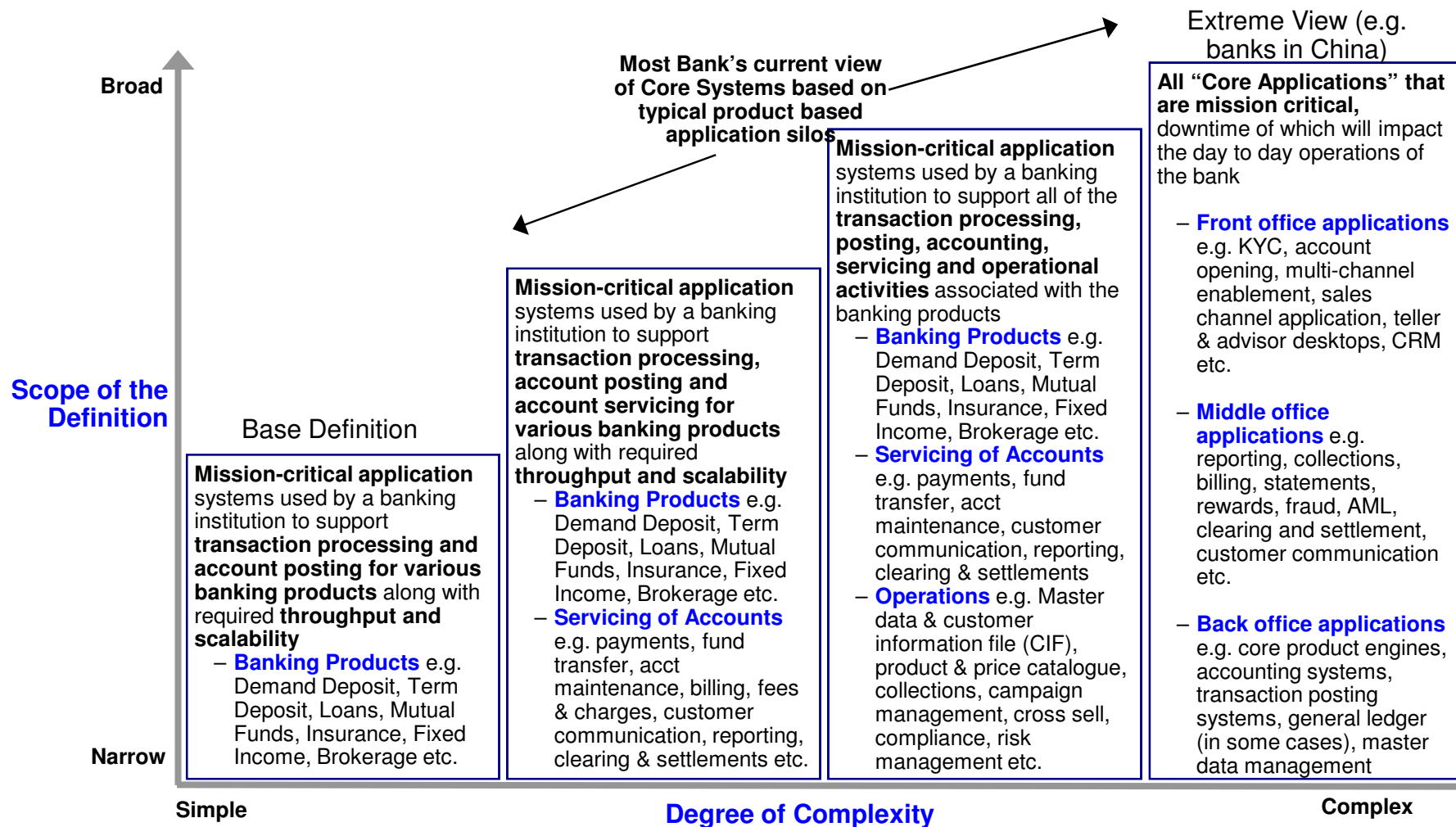
Enterprise Data Management:

Large North American Bank

Bank has identified savings and executed a major data transformation to achieve a single view of customer, product and contracts data.

What is the definition of Core Banking Systems⁽¹⁾

IBM's view of "Core Banking Systems" definition includes all mission-critical application systems used by banking institution to support transaction processing and account posting for various banking products along with required throughput and scalability



Transformation in Banking

Why Transform - Many top tier financial institutions have embarked on transformation programs driven by burning platforms around

- time to market,
- complexity,
- cost reduction
- revenue growth.

What to Transform - The focus of these transformation programs are varied and have included

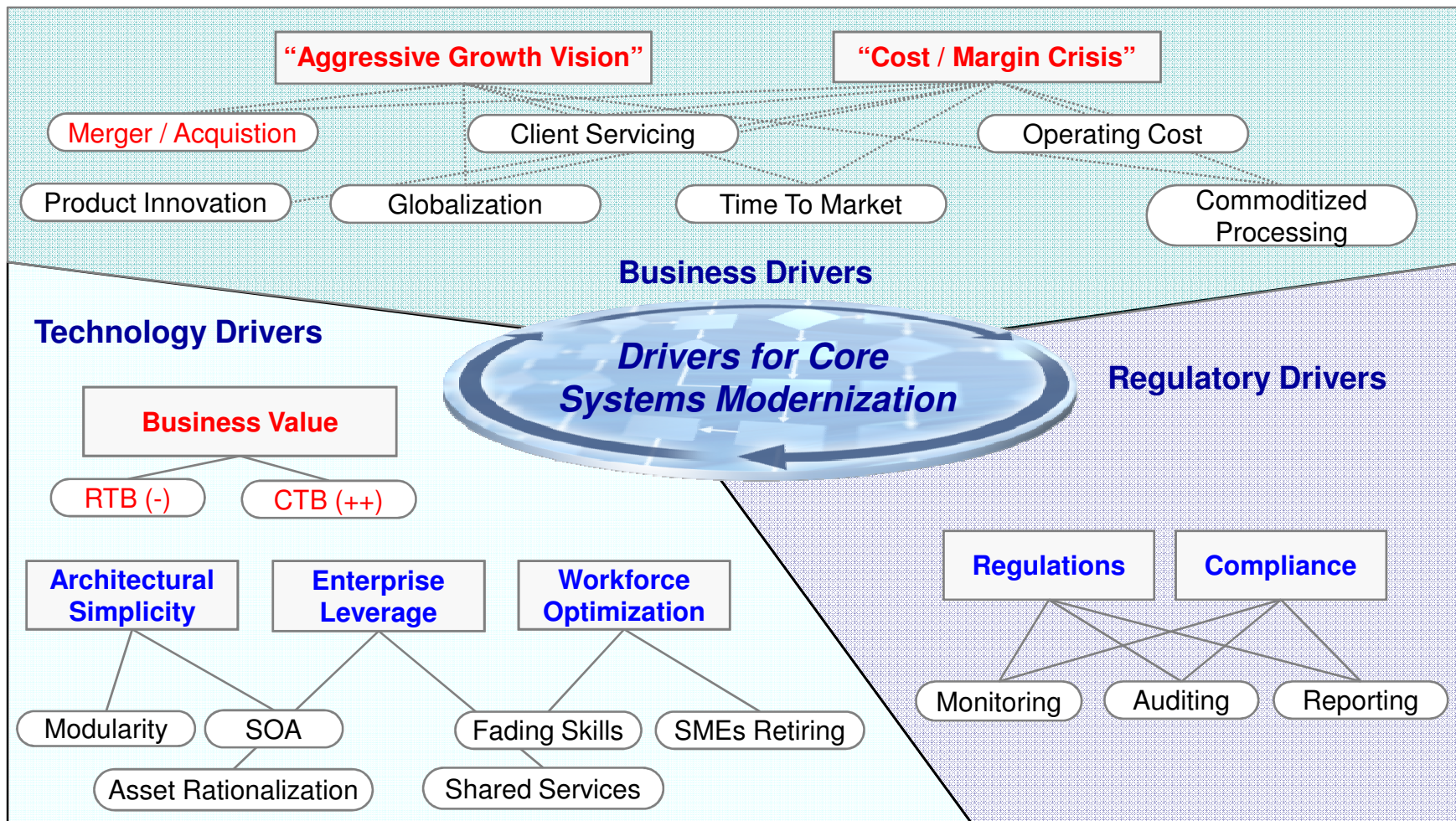
- back office operations and technology,
- the customer experience and interaction,
- data and information management,
- modernizing organizations, business processes, applications, architectures, and platforms

How to Transform - The approaches vary depending on their competitive and operating environment,

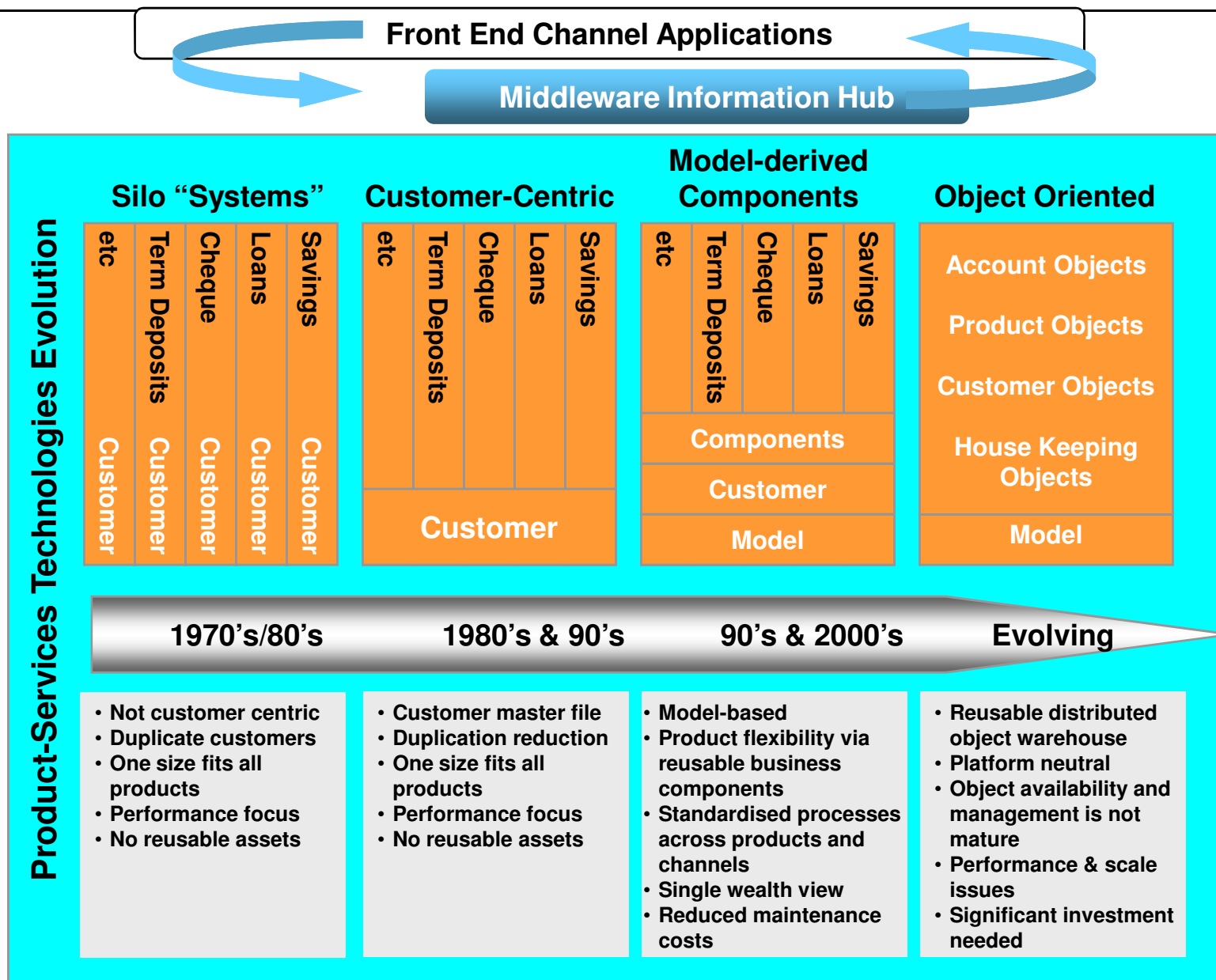
- growth aspirations and
- risk/reward appetitive

Theme: Determine the main transformation themes that applies to your bank and express your case within the themes boundary

Banks are looking at core systems modernization in response to specific business, technology and regulatory drivers



Corebanking Application Evolution Path



The Banking infrastructure challenge 1: Cost and Complexity

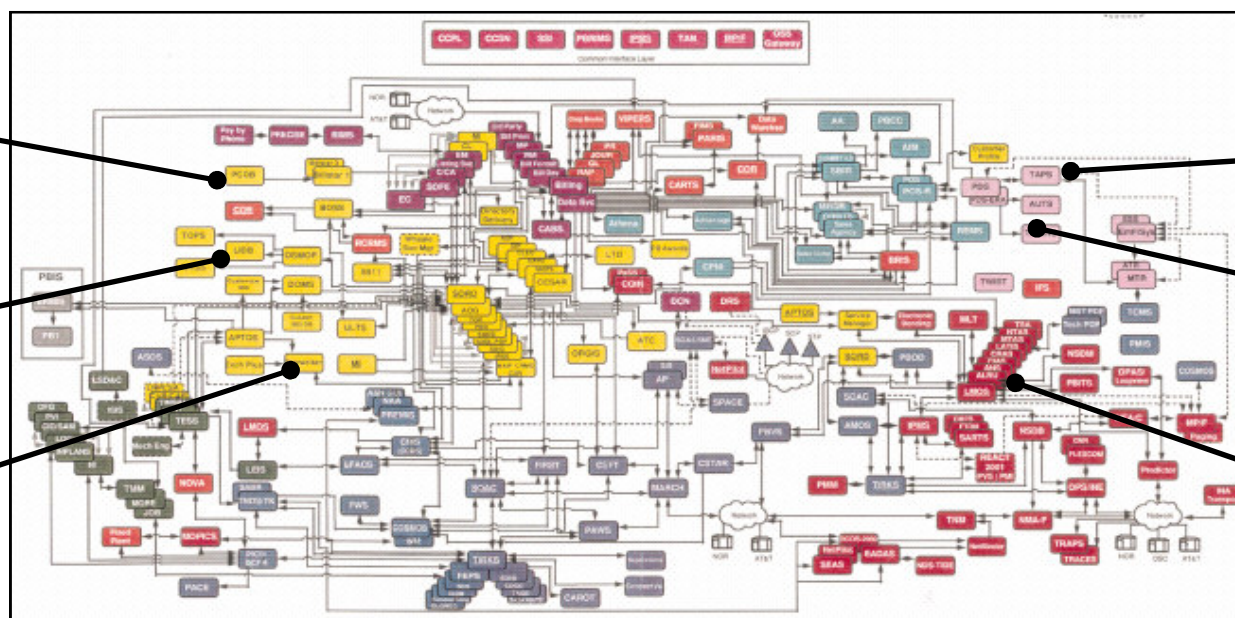
▪ **Partners**



▪ **Customers**



▪ **Channels**



▪ **Employees**



▪ **Suppliers**



▪ **3rd Party Providers**



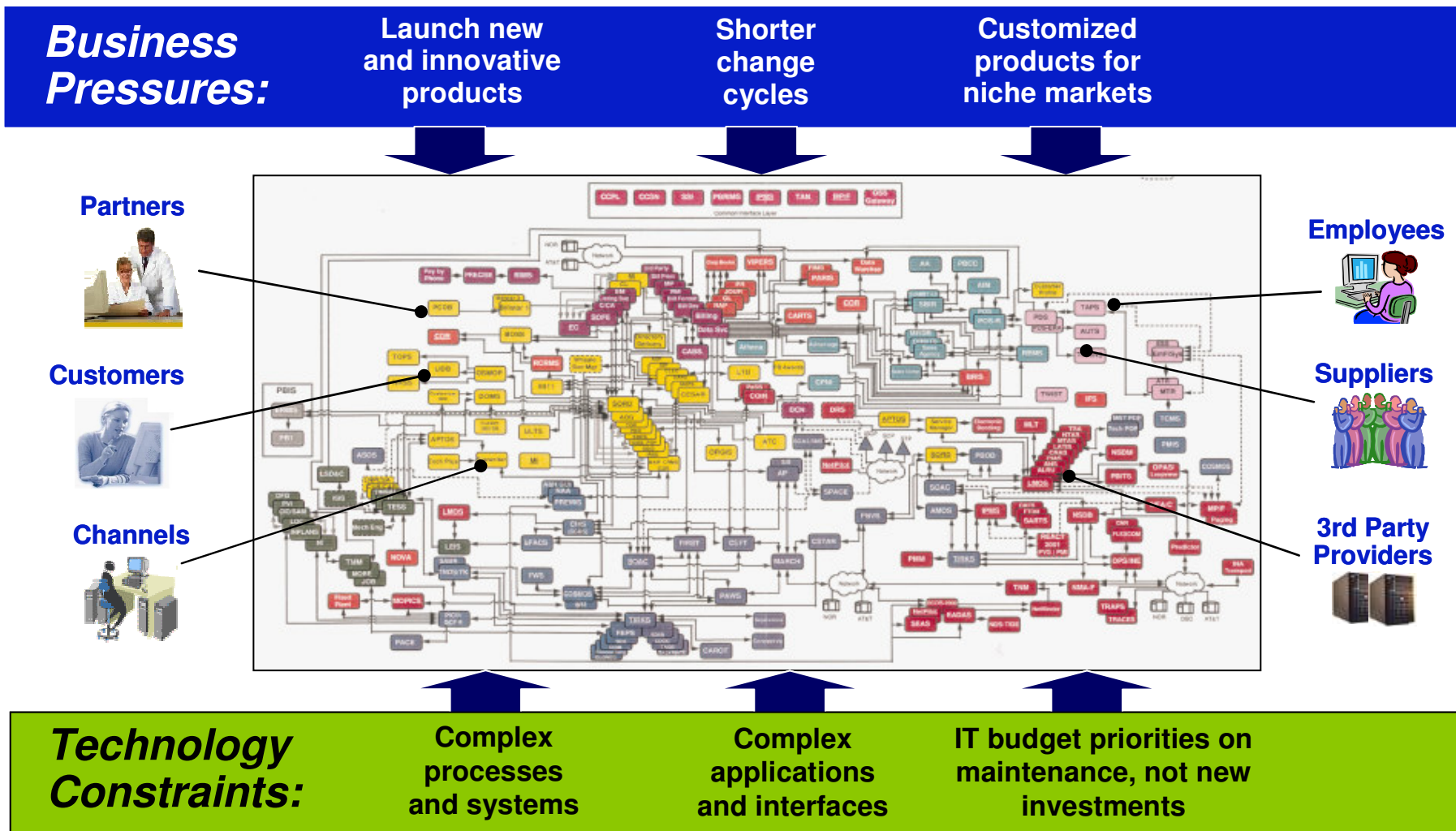
Technology Constraints:

Complex processes and systems

Complex applications and interfaces

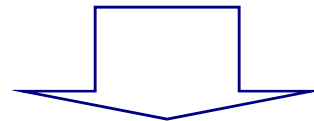
IT budget priorities on maintenance, not new investments

The Banking infrastructure challenge 2: Business and IT Alignment



Current constraints in the IT systems today

- Data Duplications to the tune of at least 15 times and multiple definitions for the same data which makes difficult to introduce new changes
- Duplication of processes at the front end and for the business end of day processing
- The current solutions has no standard architecture and most cases they are programs addressing the processing requirements
- Heterogeneous IT infrastructure with different versions of the same software and in some cases unsupported environment
- No documentation of the current IT systems, interfaces, database details, program specs etc
- Most environment developed their Corebanking solution based on the business models used in 80s and the business model is completely different at this stage
- Most interfaces are point to point and makes the current environment more complex and there is no messaging architecture or messaging hub to integrate new technologies.



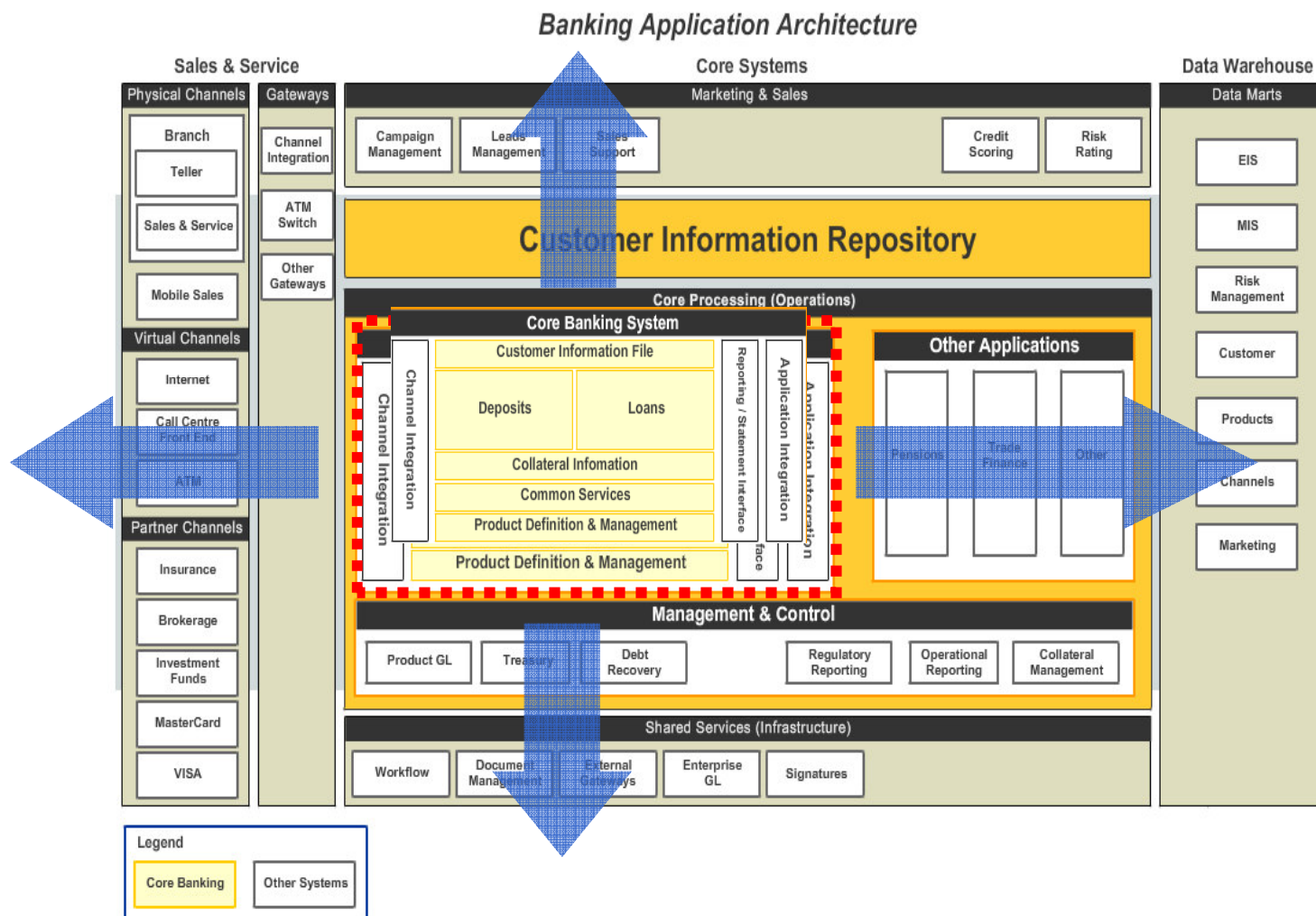
Implementing a new Corebanking solution more complex and challenging. This is the difficulties most of the bank faces today. Most Corebanking projects fails or misses the time line due to the integration and transformation issues.

Addressing the Business Drivers and Obstacles requires more than an ISV package decision.

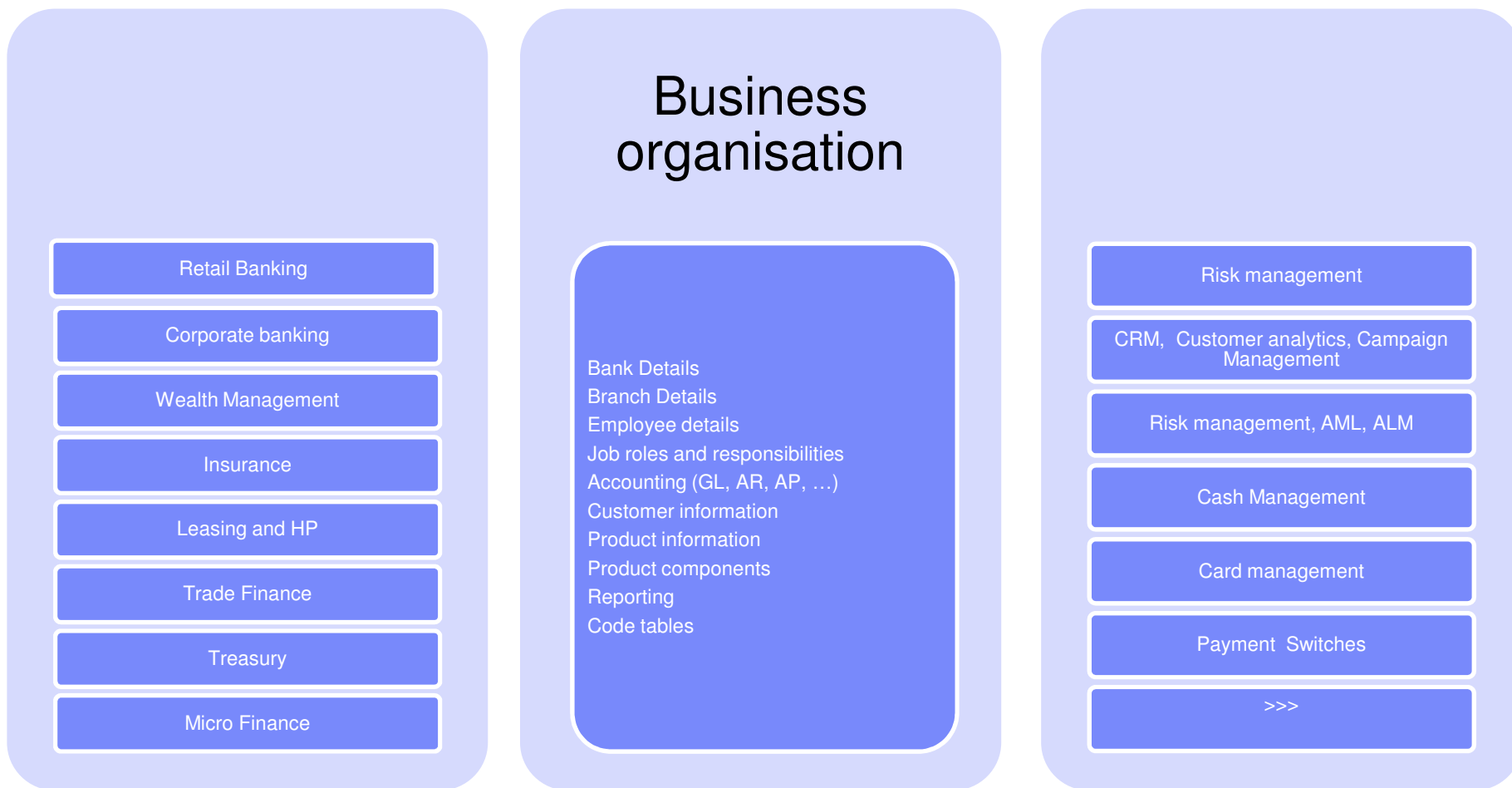
- Creating and gaining acceptance for the compelling event business case,
- Determining the partnering strategy,
- Determining the 'to-be' state for business model and infrastructure,
- Making the Build vs. Buy vs. Rent decision,
- Developing the transformation strategy to achieve the 'to-be' state for the business
- Developing the migration strategy to achieve the 'to-be' state for the infrastructure,
- Designing the project plan and identifying the right core resources required,
- Creating and determining the project business case and governance structure,
- Implementing a project risk mitigation plan,
- Creating a end to end test strategy for the total initiative
- Executing the project

Core Systems Transformation projects are complex and easily become troubled projects

A bank's core system play a center stage role in all aspects of its operations

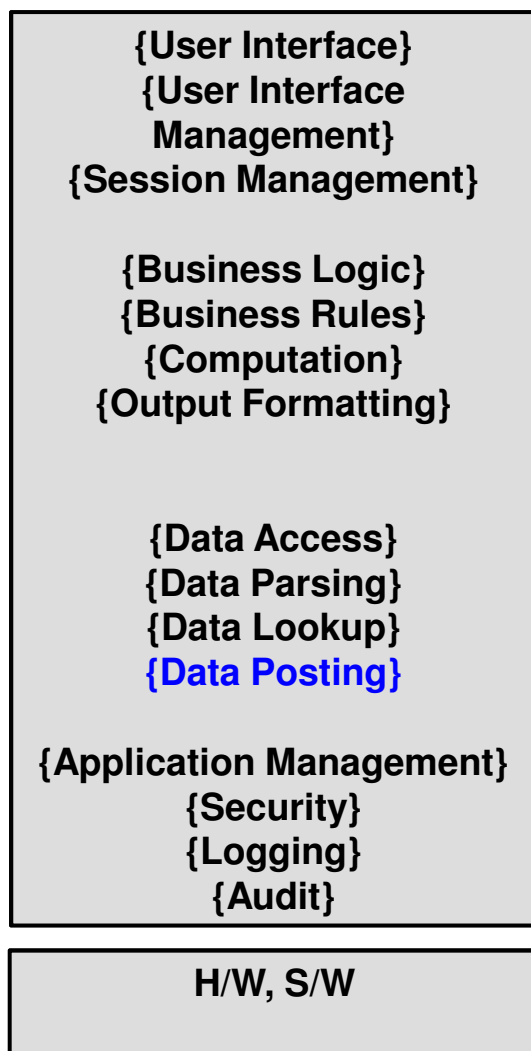


Business Solution structuring



So what aspects of core systems really manifests the problem?

Core Systems (e.g. Deposit) Architecture



- Green Screens
- Need to open multiple applications
- Limited ability to address customer needs
- Difficult to change application functionality to hard coding
- Duplication of business capability across many applications
- Multiple point-to-point connections
- Application code are interconnected top to bottom making changes very difficult
- Code duplication
- Data Duplication
- Multiple definition of the same data
- Dead code pool
- Multiple versions of COBOL compilers
- Old Platforms

Impact

Time to Market

- Development time for new business capabilities are long
- Difficult to address market opportunities in timely manner

Complexity

- Difficult to manage
- High Risk
- Performance & Scalability Issues

Cost

- High cost of maintenance
- High cost of development
- Longer and complex integration testing
- More people and more time to make change happen

Revenue

- Difficult to address customer centricity
- Difficult to increase average revenue per customer

Most of the legacy core banking systems exhibit common characteristics. They are product siloed.

Legacy Deposit System
(millions of line of code,
100's of programs)

UI	~ 5% of code
Business, Process, Logic and Rules (code that manages business processes)	~ 45% of code
Application Management (code that manages security, logging, audit etc.)	~ 30% of code
Data File (code that manages customer data)	~ 10% of code
Product Engine (code that writes and posts transaction to GL)	~ 10% of code

- Legacy deposit systems were designed to be stand alone, self contained applications, Often dedicating more than 50% code to non product related features
- Complexity is caused by multiple point to point interfaces
- Codes are often duplicated over no. of years
- No defined and easy interface to make changes to code
- Dependencies between codes and programs cascades in both vertical and horizontal direction

ISV's core banking packages are no different than legacy systems as they also have vertical structure around a single product except for....

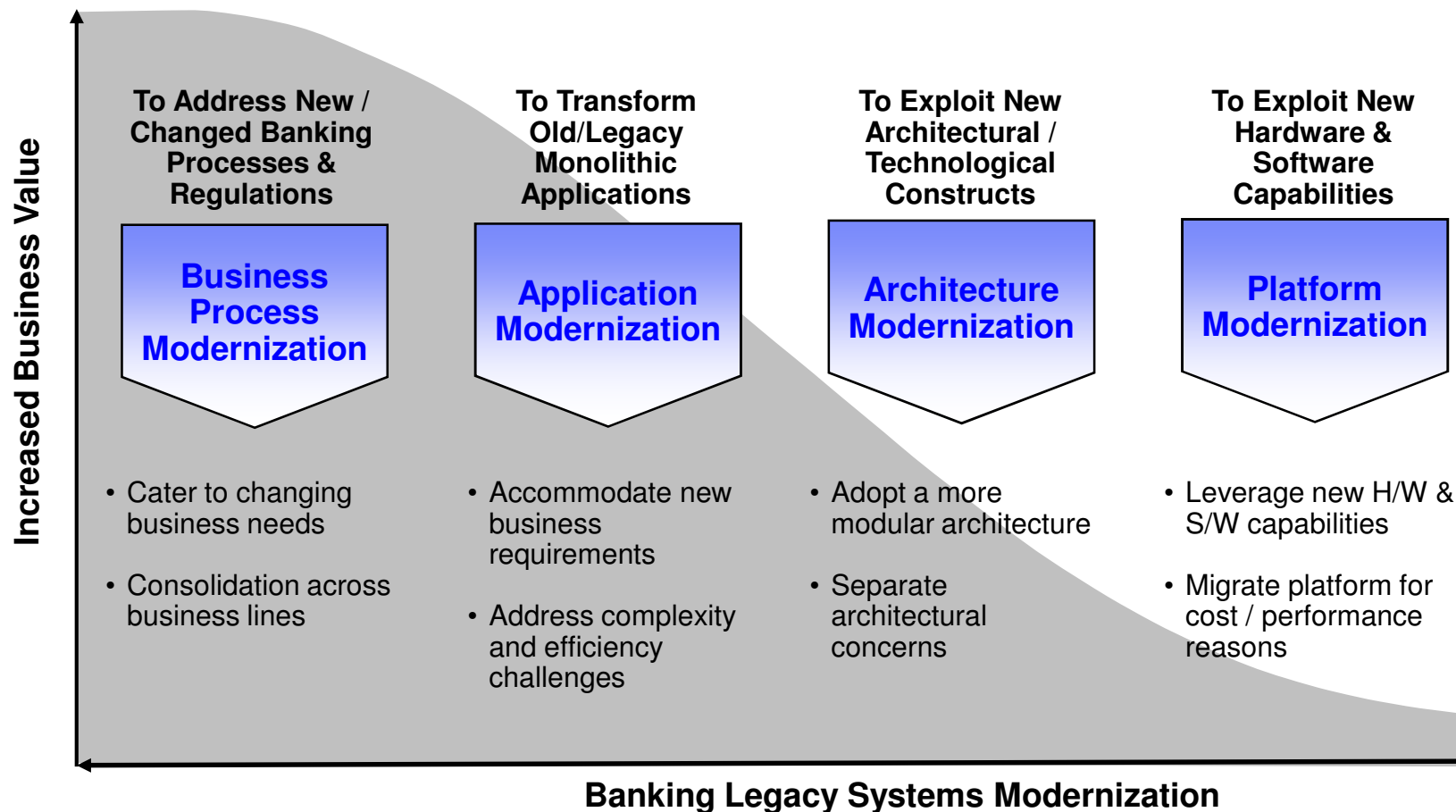
**ISV Deposit System
(millions of line of code)**

UI	~ 10% of code
Business Process, Logic and Rules (code that manages business processes)	~ 40% of code
Application Management (code that manages security, logging, audit etc.)	~ 20% of code
Message Broker (Code to manage various pieces of application)	~ 10% of code
Data File (code that manages customer data)	~ 10% of code
Product Engine (code that writes and posts transaction to GL)	~ 10% of code

- Package applications have improved interfaces to make changes to deposit applications (improved configurability and parameters)
- Message broker based architecture provides some degree of modularity and less cascading dependency

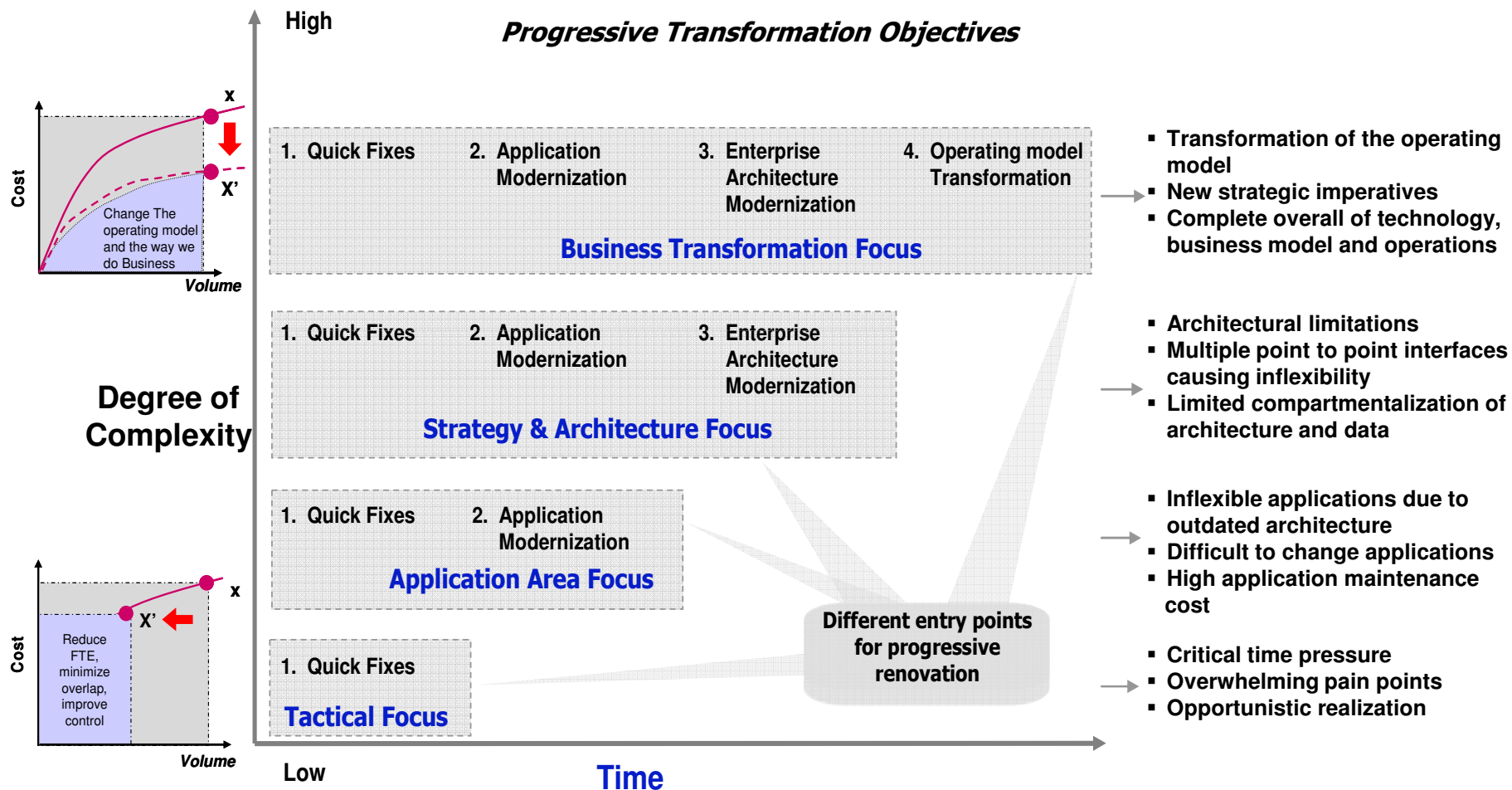
What does Core Banking Modernization⁽¹⁾ mean to Banks?

Transformation is more than just modernizing or replacing the old legacy systems



When banks embark on legacy system modernization they may include one or more areas in their modernization focus

Any renovation approach should provide multiple entry points to banks to pursue modernization of their core banking systems



Transformation: the operation of changing one configuration or expression into another in accordance to specific rules

What approaches should be pursued by banks for modernization?

Core Banking System Modernization Approach

Key Characteristics

<p>Package Implementation</p>	<p>Rip & Replace</p> <p>Progressively Replace</p>	<ul style="list-style-type: none"> • Lack of knowledge, skill sets and documentation • More than 80% business requirements met by a package solution • Bank is willing to compromise on day 1 functionalities <hr/> <ul style="list-style-type: none"> • Really old systems. Lack of knowledge and skill set • Rip and replace is not an option as bank cannot compromise on day 1 functionality offered via package • Package solution can meet upto 80% of business requirements 	<p>Korean Banking examples</p> <p>CCB, NAB, CBA, BoC, DBS, MayBank,...</p>
<p>Re-Write</p>	<ul style="list-style-type: none"> • Systems are too complex to be replaced by package solutions or undertaking legacy modernization. Best option is to re-write again • Lost documentation • Legacy analysis proves too cumbersome and expensive 	<p>Large Global Banks like ICBC, ABC, HSBC,</p>	
<p>Progressively Modernize</p>	<p>Legacy Applications</p>	<ul style="list-style-type: none"> • Legacy systems old, has many cascading dependencies and contains years of organic development and differentiated capabilities that are not available in packages • Package solutions can only meet upto 50% business requirements • Banks want to undertake a step orderly modernization driven by solving priority pain points • Banks wants to harvest its legacy and the differentiated capabilities 	<p>Preferred Choice By Many Banks in the past. The time has come for them to replace the core in order stay competitive in the market</p>
<p>Hybrid</p>	<ul style="list-style-type: none"> • A combination of other approaches • Usually targets a contained module for package replacement e.g. trade finance if business requirements are met in package and to address time to market issues • Choosing package for new markets and modernizing legacy for core • Tries to mix best of legacy leverage with capabilities from package 		

In What Context Banks Want To Modernize Legacy Applications

There are many flavors of application modernization, but the key problem that banks are trying to solve is to get a better handle of their applications, reduce complexities, reduce maintenance cost and improve time to market for meeting business needs

Application Modernization

- Code conversion from Cobol to Java/Cobol
- Compiler standardization
- Code cleanup (dead code removal, duplicity removals, method level interface cleanups)
- **Some applications can be running on a unsupported software version**

Application Modularization


- Break legacy code into smaller manageable modules
- Extract legacy functionality as sharable services to reduce duplicity
- Break hardened code through use of enterprise integration
- **Example ; Bank specific business rules, Reporting, product specific code, ...**

Application Rationalization

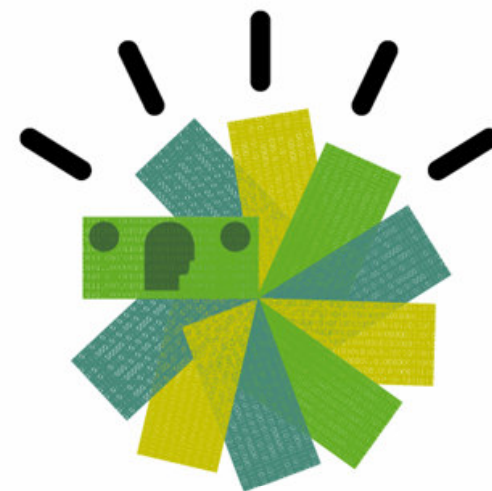
- Rationalize applications across the enterprise
- Remove duplicate and redundant applications
- Rationalize functionalities offered by applications
- **Generally very complex and not many has ventured in this space**

Application Retrofit

- Retrofit legacy applications to use new architectural constructs with an aim to clean up the applications
- Retrofit applications to use common shared services
- **Some of the application and modules can be converted to support the functions in the new architecture environment**

 ***Banks often try to attempt doing one or more from the above. but lack of technological capabilities, underlying complexity of existing systems and lack of skills and resources makes them consider replacement options***

Core Banking Transformation



R

