LEAP™

A Blueprint for Legacy Core Technology Transformation in Financial Services

Part 1 – A Case for Change

Part 2 – Taking the LEAP

Michael Sanchez
msanchez@savanainc.com
www.savanainc.com

PART 1

A Case For Change

Back in 1980, I started a software company with my brother Frank to develop a next-generation core banking system. About six years later, the CTO of Chase Manhattan Bank visited us and was totally smitten by our new real-time banking system. At the end of the meeting he said *"Boys, this system can have a profound impact on The Chase Manhattan Bank – and when you can tell me how I can go from what we have today to your new system without putting the entire bank at risk, I'll buy it!"* We went on to become a successful public company after enabling banks with our new core in emerging markets overseas, and in de novo Internet banks here in the States, but we never did answer that question posed over 30 years ago by one very astute CTO.

Now that Frank and I are at it again developing a next generation core-as-a-service for the digital age with our new company Finxact, we finally need to answer that question. The answer lies in following a process using new technologies that encapsulate legacy systems and stage them for replacement with minimal technology risk or operational impact. The tools and processes described in the second part of this paper can support the variety of business approaches under consideration by banks intent on digital transformation. In part one of this paper, I will focus on transformation imperatives faced by traditional banks servicing consumer and commercial customers. The solution described in part two also applies to other segments of the financial services industry, like Insurance and Capital Markets. In fact, it can apply to any industry where similar legacy problems persist.

What is a core system?

A core banking system, often called the System of Record or SOR, is the system that defines product behavior, processes customer transactions, stores balances, and calculates interest and fees. It sits at the heart of a bank's technology infrastructure, surrounded by an ecosystem of other applications that manage payment systems, CRM, branches, call centers, mobile banking, and internet channels. The core is analogous to the factory of a company that makes hard goods.

There are typically unique core systems for different product types like retail deposits and loans, commercial lending, mortgages, and credit cards. Some core systems date back to the 1960s with most being from the 1970s and 1980s. With the frenetic pace of technology evolution over the past few decades, these systems should have been relics of the past by now.

What is the problem?

So why haven't banks replaced them? It boils down to the perceived risk outweighing the benefits. There is some rationale behind this thinking. According to a Cognizant study from 2013, 75% of all core renewal projects in banking have failed by either being cancelled or not meeting business objectives.[i] With a record like that, it's no wonder that most bank executives would rather leave this problem to the next administration.

Over the years, large banks have added layers of technology around their cores to help compensate for deficiencies or add capabilities like mobile banking, bill pay, business intelligence, and P2P payments. This makes for a very complex and fragile IT environment that is not conducive to replacing individual components.

In the community and mid-tier bank market, three vendors (FIS, Fiserv, and Jack Henry) control most of the technology banks use in the United States. They typically offer the core system with all the surrounding systems like mobile banking, payments, check processing, online banking, branch automation, ATM software and the like, all bundled into a single long-term contract. This makes it very difficult to target individual components for replacement, even if a low risk migration approach was available.

Another factor that plays into the risk equation is that there are currently no good replacement options available that provide enough benefit. Although there has been a great deal of innovation by Fintechs, they typically provide bespoke consumer direct solutions, like alternate payment systems, social lending, or account aggregation intended to disrupt legacy institutions as opposed to arming them with new capabilities. This is changing as a number of start-ups have emerged intent on offering banks a next generation core system designed for the digital age. Despite these challenges, banks recognize the need to make a change. A study in 2016 by NTT Data Consulting showed that 80% of banks surveyed want to change their core, but only 15% intend to initiate a replacement project in the next three years.[ii]

Why change the core?

With all this complexity, vendor contract issues, and the failures in the market for those who've tried, why bother transforming legacy core technology? Frankly, there are lots of reasons and they have been accumulating for decades. The technology at the core of most banks was designed back when the only channel was the branch, and the payment system was primarily physical checks that took seven days to clear. We now live in a time when consumers expect everything to happen in real-time, on-demand, and to be available through their device or channel of choice. According to McKinsey's 2015 Global Banking Annual Review, up to 40% of banks' revenue will disappear by 2025 from consumer finance, mortgages, SME lending (funding of small and medium-sized enterprises), payments, and wealth management unless they have new tools to compete. 70% of banks surveyed stated that they don't currently have these tools.[iii] Market share is shifting to Fintechs not saddled with legacy issues that can launch better products and consumer experiences quickly. Several studies have identified that banks spend between 75-80% of their IT budget on remediating and maintaining legacy systems. This leaves very little to invest in new technology that could reverse the negative trends and fuel growth.

What is wrong with the existing cores?

Existing core banking systems are monolithic, tightly-coupled applications that were hard coded to handle the functional requirements of a specific product type. Their rigid features are difficult to re-tool and create road blocks to product innovation. Not unlike what happened to the US auto industry in the late 1970s, banks with legacy cores will be unable to respond when the competition arrives with innovative products delivered from cost efficient and agile new factories.

Many banking cores still operate on expensive mainframe computers using proprietary languages and databases. This makes it virtually impossible to take advantage of the advanced features available with cloud-native computing to lower costs, provide elastic scalability, interoperability, redundancy, and enhanced data security.

The legacy cores highly-structured and inflexible databases were never designed to capture the explosion of data that has occurred over the past decades and which will multiply again with the emergence of the Internet of Things (IoT). Without capturing this data when customers transact, banks are blind to critical insights into their customers' behavior and preferences. In a BI Intelligence survey from 2017, the number one complaint of banks was their inability to make use of data available about their customers.[iv]

Core systems' inability to seamlessly interact outside a closed financial network prevents them from co-existing in a growing landscape of apps and web services that can further enhance offerings and customer knowledge. In the new world of API banking, interoperability is a core requirement that legacy systems are incapable of dealing with.

Instead of being able to take advantage of this new connected, data-rich world, many of the largest banks are operating on batch-based core systems that accumulate transactions during the day to post at night. They have teams of people in reconciliation departments manually fixing customer accounts as they try to cope with a new age of digital banking that is always on with customers transacting 24x7 in multiple time zones.

Existing core systems are extremely vulnerable in the new reality of global security threats and massive data breaches. These systems are like an aircraft carrier sitting in the middle of a navy battle group. They are the prize of any attacker and are sitting ducks if outer defenses are breached. According to the BI Intelligence report, 74% of consumers would switch banks in the event of a security breach.[iv] Yet today, the best most banks can do is extract historical files and look for anomalies long after a security event has already occurred. The new generation of cores will have the native ability to detect fraudulent or malicious activity in real-time to minimize the impact of a breach of the outer defenses.

The issues identified for each core are multiplied by the number of cores a bank has. Each one is a separate silo with a different architecture and technology. Large banks have dozens of these systems to support their different product offerings. Even the big three outsourcers for small and mid-tier banks have multiple cores for each product type, having built their business over many decades by acquiring smaller core processing companies. Because of existing cores' monolithic nature, there are dedicated teams required to support all the functions of each one a bank has. Account servicing, back office processing, regulatory compliance, security monitoring, third party integration, and system maintenance are replicated across each core. This has saddled the entire industry with the overhead of supporting these ancient relics of the past and represents a huge unrecognized cost burden that will be exposed when the new generation of cores arrive. In the new cores, a single system can support all of a bank's products. These cores can easily assemble unique product characteristics in a digital product factory, eliminating the need for multiple product silos. Having retail, commercial, mortgage, and card products all in a single new core will fundamentally change the cost and service capabilities of the industry.

Last but not least of the problems with existing cores is the fact that many of the people who maintain them and most of the people who designed them have retired or are about to soon. I was at a large bank recently with a home-grown retail banking core that still uses assembler code which pre-dates the COBOL language released in 1961. Nobody graduating college and looking for a career in technology wants to work on these dinosaurs. With no one left to support them, banks really don't have a choice but to figure out how to replace them.

The McKinsey study summarized the requirement for banking services to be on-demand, real-time, frictionless, and very inexpensive, yet existing legacy cores meet none of these requirements.[iii] 87% of banks surveyed in the 2016 Capgemini report said their existing cores could not sustain a digital ecosystem.[v]

What are the benefits of the new cores?

I've heard many people argue that there is really no need for a new core if a bank can upgrade everything surrounding the existing core to improve a bank's capabilities. In my opinion, this is a false premise driven by the fact that the industry has lived with existing cores and their limitations for so long that it's hard to see the possibilities that a new core enables. For many, it will be like driving a new Tesla for the first time – you'll feel like you have just experienced the future.

Since the core serves the role of the factory for a bank, it is where innovation needs to start in order to deliver better products and services to customers. A new core with an electronic product assembly capability allows banks to offer innovative products with features targeted to the unique preferences of any size market segment, even a market of one. In addition to dramatically lowering costs across the bank, the elimination of traditional core product silos allows easy configuration of hybrid products that combine features previously confined within individual cores.

The ability to capture vastly more data in the core from customer interactions and transaction events unlocks tremendous value in understanding customer behavior and preferences. This data is available to new artificial intelligence/business intelligence tools which can anticipate customer or segment needs. Banks can then develop and deliver new products in dramatically shorter cycle times using new product assembly tools.

The ability to create an internal and external ecosystem of new apps and services through straight forward APIs opens the banking system to an entirely new set of capabilities. Like an app store, banks can offer best-of-breed third-party services to benefit their clients and capture more business. Open APIs coupled with agile product manufacturing allows banks to easily implement their own unique products, business process and customer facing layers that differentiate them in the market.

The ability of new cores to operate in cloud native environments dramatically reduces infrastructure and support costs while providing a more robust and scalable computing environment. These cost savings can exceed 70% compared to today's hosting and on-premises computing models.

Banks can also leverage an internal rules engine invoked as part of the API or during transaction processing execution to detect patterns in real-time such as fraud, AML, security breaches, and customer selling opportunities.

The availability of real-time (vs batch) banking systems has been around for decades, but in the new world of *"always on,"* core systems will continue to run 24x7 without requiring batch windows or lock downs during interest posting cycles, system upgrades, and the like.

How does the industry transform?

The original question posed to us 30 years ago is still valid, and a Deloitte study validated it again by identifying the biggest obstacle for banks wanting to make a core change as "understanding what to do and where to start."[vi] The rest of this paper will provide a blueprint for successfully performing a legacy core transformation.

The answer is called LEAP. It is an acronym for Legacy Encapsulation Abstraction Process. The combination of technology and steps in LEAP de-risks the transformation of core systems and flips the dismal failure rate into a high success rate. One of the problems with long-term transformation projects is financial and human fatigue. With no tangible benefits, often for years, organizations simply cancel the project and move on to other things that are more pressing. LEAP provides value at each step in the process and often pays for itself before the core is replaced. Even if a new “target” system has not been selected, or is not yet proven enough in the market for a bank’s risk appetite, LEAP still provides value while staging legacy systems for sunset whenever the time is right. This makes it easy to justify now.

PART 2

Taking the LEAP

LEAP Overview and Objectives

In a series of discreet steps, LEAP uses technology tools to de-couple directly connected third-party components of the core ecosystem, as well as functions of the legacy core, and abstracts them into a new API-centric servicing model. These new Process APIs encapsulate the components, business rules, workflow, and integrations required to perform specific business tasks. Banks can upgrade the individual legacy components encapsulated in a Process API over time, while simultaneously preparing the bank's operations and legacy core for future replacement.

In modern systems, the databases, business rules, bank processes and user experience are all separated into distinct layers that communicate with each other through a messaging architecture or APIs. A key LEAP objective is to externalize some of the vertically integrated functions of legacy cores into new layers.

Creating a new servicing layer makes the legacy core "headless" by replacing the native servicing screens with new UI tools.   This allows the implementation of bank-facing servicing technology before converting the core. When the bank replaces the core, the user interface will not change, significantly reducing operational risk because the bank's personnel will already be familiar with the new interface. In addition to creating a new servicing solution that is used by internal bank staff, this step standardizes business functions and exposes their Process APIs to web applications and other digital channels like mobile banking, personal assistants, and wearables. These Process APIs remain in place through the entire transformation, significantly reducing integration risks during the core conversion.

The objective of all the steps performed in advance of the core conversion is to create incremental value while mitigating transformation risk by reducing the number of variables to the absolute minimum when the day comes for the core conversion.

There are a number of consulting companies evangelizing different approaches to the challenge of legacy transformation. These include wrapping and freezing cores to launching parallel or digital banks using all new technology. The LEAP technology and process supports each of these approaches. Later, I'll discuss a hybrid approach using the LEAP tools along with a parallel bank that can provide the fastest route and lowest risk to legacy transformation.

The LEAP Technology

LEAP requires tightly-integrated technology tools that work together seamlessly at scale. Otherwise, the effort to combine the components becomes a large non-value-added project in and of itself. All of the new software components should be cloud-native applications to facilitate the simultaneous transformation of legacy applications and technical computing infrastructure. Below is a list of the functionality required for each of the integrated components of LEAP.

Domain Data Model

Creating a comprehensive domain data model (DDM) is foundational to implementing a homogenous operational model, particularly when performing common functions across multiple core systems or business units. A DDM is not a data warehouse, data mart, or any other form of replicating data into a single data repository. A domain data model provides the new technology components with a common language, definitions, and relationships of customer and account information across the organization. Without a common data model and definition language, it is nearly impossible to ensure accuracy of information needed to support business processes, decisions, or execution across disparate systems. The defined model should be agnostic of the idiosyncrasies or deficiencies of the underlying core(s). Nearly every existing core system has a definition of a customer, though no one would argue that any of these systems provide a truly enterprise-wide view of the customer.

After defining the DDM, the next step is building data adapters to translate the legacy system data into the DDM's entity and business object shapes. This enables business processes, business rules, user experiences and channel applications to have a uniform and homogenous way to address the underlying core data. Together, the DDM and data adaptors become the "Rosetta Stone" that translates and normalizes data definitions from multiple systems participating in business processes.

A key benefit of LEAP is that the DDM remains constant through legacy component replacements so the new servicing layer does not need to change or require re-coding for any new component, including a core. Instead, new components only require a new data adaptor and/or API. Additional data elements that become available from any new component are added to the DDM to expose them in the LEAP servicing layer. Since the servicing layer uses modeling tools, exposing new data or business processes is not a "coding" effort and can be accomplished very quickly.

Data and Integration Adapters

Interfaces and adapters that map and connect the underlying core system to the DDM are required to allow the new servicing layers and APIs to uniformly address legacy systems. Modeling technology is commercially available to rapidly construct these data adapters using low-code and visual design approaches without traditional and expensive software development efforts. Providing the core has some addressable integration interfaces, even direct to database access, these modern modeling tools allow for the rapid delivery of bi-directional interfaces to the core data. It is important that these are direct, real-time interfaces where possible versus simply adapters to data warehouses where core data is periodically staged. Real-time interfaces enable better rule-based decisioning, workflow and automation based on current operating state of the customer or account data.

Process Application Program Interface (Process API)

An API provides the ability to abstract and expose business functions and discreet services from code for use by other applications. In LEAP, the Process API architecture combines the integration layers for the cores and ancillary systems with the workflow, process, and business rules. Since the Process API encapsulates and orchestrates legacy components of the ecosystem, upgrades of those components do not affect the external-facing Process API, connected channel applications and servicing layers. Traditionally, legacy cores do not share the same underlying technology or services architecture with other channel-specific applications, making business processes inconsistent. The Process APIs serve as a library of discreet customer and account servicing functions that are re-usable across the enterprise. Encapsulating these functions into APIs allows their use by applications serving all banking channels; unifying business processes and creating a consistent OMNI- channel processing environment. The Process APIs enable the transition from assisted to consumer-direct servicing and straight-through processing.

Another benefit of the Process API is that some of the business logic, regulatory controls, and customization historically done in the core can be pulled into this layer. This approach reduces the development efforts on the legacy core to support changing business processes and regulatory requirements. It also separates some of the business logic from the transaction engine in the core, making the core transformation much less complex.

User Interface (UI) Framework

UI authoring tools create the interface between human users and computer applications. The LEAP UI is another layer of abstraction that uses APIs to communicate with underlying services. LEAP uses a low-code UI framework that minimizes the development effort and allows for more rapid evolution of user experiences. Particularity with mid and back-office user experiences, these new tools provide a way to circumvent lengthy development efforts to create a new comprehensive UI. Much like Business Process Management (BPM), user applications are built on modeled, configuration-driven frameworks, and integrated with a robust API to create the most efficient and agile solution.

The new UI will eventually become the single user experience in the mid and back-office. Traditional approaches of supplementing servicing with independent applications and UIs has resulted in more siloed applications simply connected to the core. In some cases, call center and back office users must use ten or more applications on the desktop to perform their job. For LEAP, the new UI serves as the foundation to creating a core agnostic, independent set of user experiences to perform servicing functions using the Process API architecture. The UI for the new servicing solution shares the same Process API with other banking channel applications, ensuring a singular and consistent process for servicing across all channels.

Business Process Management (BPM)

BPM provides a robust set of tools to configure and model business processes. BPM orchestrates and automates human and systems functions, creating repetitive, consistent, and efficient execution of tasks. BPM allows organizations to model processes and rules in such a way that they can be rapidly changed to reflect new operating and regulatory requirements.

The BPM layer needs to be contextually aware of the underlying domain data model. This typically happens in the BPM modeling architecture, business rules modeling, and data object modeling.

BPM provides a new architecture to re-integrate third-party applications currently directly tied the core itself. These include AML/KYC functions, statements, CRM, correspondence production, and other services. BPM links and orchestrates third-party applications and micro services that participate in a business process through direct integration or by utilizing their APIs. Since the goal of LEAP is to eventually replace the core, a critical step is to reduce the direct integration dependencies of the current ecosystem. The LEAP BPM provides configuration-driven orchestration of the ecosystem versus hard-coded integration to the core.

BPM also provides an abstraction layer to handle the idiosyncrasies of each core. If a bank has to access more than one core for a specific service, the BPM layer can create a single, homogenous process and orchestration, but still deal with unique core-level requirements as needed using parameter-driven business rules and process logic.

BPM provides the outward facing API of the new architecture. The Process API is available to all channels to create consistent processes as part of the digital transformation. For example, invoking a consumer complaint or Reg E dispute from any banking channel only requires development of a simple user experience for the channel application that calls the Process API. Behind the scenes, the Process API and BPM component routes work to the appropriate systems and people to fulfill the service request. It also exposes the status and expected outcomes, creating process transparency to the channel originating the request or any other interested party.

Enterprise Content Management (ECM)

In legacy environments, much of the customer and account servicing process is dependent on chasing paper. The primary reason is that the cores have historically lacked underlying workflow and servicing process management. Forms and documents have historically provided the required audit trail and details when capturing the data and information to create customer requests and handle exceptions. Even in cases where various document and content repositories are in place, there are often multiples of them across the enterprise serving specific purposes (statements, notices, correspondence).

The LEAP ECM provides a complete repository of content related to customers and accounts that is accessible by any stakeholder or process within the enterprise. A singular, unified store of information creates a seamless aggregation of information to support customer servicing across all business processes and channels. The content is tied to a specific action to provide "content in context" of a business process, as opposed to another disjointed information silo. If using a single content store is not possible, the individual content stores should be logically unified by connecting all of them under a single API.

Password Management

One of the major integration challenges when implementing a new servicing layer around the core and its ecosystem relates to user password management, authentication, and entitlements. Building APIs, services, and user experiences that effectively eliminate the direct use of native legacy screens requires visibility of a user's entitlements. Every core and ancillary system has internalized the user identity, password management, role authorizations and entitlements. No single technology or vendor solution provides a holistic solution to solve the SSO and entitlement management across the technical environments of the cores and related systems in the enterprise. There are a number of different approaches to extract user entitlement functions and manage it outside the core(s). This does not strip the core of its entitlement management feature, but merely begins the process of externalizing the user's rights and authority to create a new operating model. If the core exposes user management APIs, this process is straight forward.

What about middleware?

I'm going to go out on a limb here and say, *"you don't need traditional middleware to make LEAP work."* Since the mid-1980s, middleware has been seen by the IT community as one of the key technologies that provide easier integration between systems and allow the much-touted Service Oriented Architecture (SOA) to be enabled. Although beneficial, middleware projects can be very complex and time consuming with little to show for the efforts since it is "plumbing" and usually does not include value-added applications. In most large organizations, the business lines pay for projects delivered by IT and they have resisted paying for enterprise-wide initiatives that do not show tangible benefits to their customers. Although well intentioned, CTOs have struggled to convince the business owners to continue paying the toll for these long-running initiatives and many are cancelled or only partially implemented. This is a classic example of project fatigue that LEAP avoids by performing smaller, more manageable projects that add measurable value as the primary goal, yet still provide integration benefits to the enterprise. Middleware can be replaced with modern APIs that routes data between systems to provide a more open and standardized approach to exposing services that individual applications can consume directly. For this reason, middleware is not an absolute pre-requisite for LEAP, but it is good to have if the bank already invested.

The LEAP Phases

Although the steps are similar for any size bank, the sequencing and time it takes will vary depending upon a bank's existing conditions, business priorities, number of components requiring upgrades, and transformation approach. It is worth the journey though, since banks who take the LEAP will have removed a major impediment to profitable growth and be positioned to compete and prosper in the age of digital banking.

Planning Phase

The planning phase is similar to other IT projects, and includes requirements gathering, developing budgets, timelines, program governance, and expected business outcomes. After completing the overall roadmap, detailed plans for accomplishing LEAP in multiple phases are developed. It is important to identify dependencies between the different phases and any undertakings requiring extended time periods. These "long poles" can include internal development activities or identifying and acquiring third-party solutions to replace existing components of the legacy ecosystem. Longer-running activities are launched in parallel with other discreet project phases so they don't hold up the timeline and they converge at the right time in the overall project roadmap. The specific project planning tools and methodologies used are not critical to LEAP so long as there is effective management and visibility for stakeholders. Like any project, the most important success factor is to have the business and IT owners aligned, committed, and capable of executing the tasks.

Document Existing Conditions and Future State

This LEAP-specific step documents the data elements, process flows, business rules, product features and integration details of the existing core and supporting systems. The same documentation is then developed for the future state, including new third-party integrations. It is helpful to create visual diagrams where possible, similar to a blueprint. Like remodeling a house, it is important to know what your existing conditions are before you begin, and to know what the desired outcome looks like. Before beginning this step, identify the tools that you will use to document and maintain the data model, process model, product features, and integration requirements. There are a number commercial tools available for these purposes.

This is the time to start building a comprehensive data model to account for data from all sources in a common definition. A good place to start is to look at the various data sources available that the industry uses for data exchange (IFX, BIAN, Open API, MISMO etc.). Building the DDM is a dynamic process so having a good tool to manage it is important.

These processes and data models will be used when creating new layers around the existing core. The layers simplify the replacement of the core and remain in place after the transition is complete. If part of the technology migration includes moving all infrastructure to the cloud (highly recommended), create a staged plan to execute this transition in parallel to the component upgrades outlined in the LEAP process below.

Wrap the Core and Related Servicing Components

The next step in LEAP is to create a new servicing layer for the legacy core. Legacy cores generally have antiquated user experiences that lack efficiency and automated workflow. Most of the call center and back office functions remain on a multitude of pieced together disparate applications, resulting in a complex and inefficient environment for workers. In this step, the BPM connects and orchestrates all the disjointed systems and human tasks to unify and automate the servicing process. These are encapsulated into new Process APIs for each unique servicing function. The new UI for the banks internal servicing staff in call centers and the back office consumes the servicing tasks exposed through the Process APIs. This provides an early win with measurable value to the organization. The automation of all servicing tasks provides tangible cost savings by decreasing average handling times and automating post-call activities. It also provides a better client experience through faster service execution and process transparency.

This step also helps the bank transition to an OMNI-channel servicing model and a consumer self-service model by allowing mobile banking and consumer web applications to call servicing functions and inquire on the status of service requests directly through the Process API. For LEAP, the process of abstracting and putting a standardized wrapper around the core and its ecosystem sets the stage for low risk migration of the core and other legacy components.

Upgrade the Ecosystem

The next step in LEAP is to replace or retire selected legacy applications in the core ecosystem that are now orchestrated by the new servicing layer. Some legacy applications may become redundant, since their functions are easily replaced by modeling them in the BPM, content management, and UI tools that were deployed in the previous core wrapping phase. This consolidation of legacy components directly to the servicing layer can be part of the core wrapping project (above) or in this subsequent step.

In this step, new third-party applications selected to replace legacy components are introduced by using data adaptors or APIs. After mapping the data from these new applications to the DDM, nothing in the servicing layer needs to change since it is already using DDM data shapes. Since the component targeted for replacement is already encapsulated into one or more Process APIs, the core and ecosystem do not have to be re-integrated to the new component. If the component requires new data elements, they are easily exposed by adding them to the DDM and modeling them into the servicing layer. Similarly, if the new application will drive a new business process, it can be modelled using the configuration tools of the BPM and new UI. The new upgraded process solution around the legacy core provides direct benefits for the bank and its customers while staging for a low risk core transformation.

Multi-Core Process Unification

The process of wrapping a single core and upgrading its ecosystem can be extended to additional cores using the LEAP tools. This is accomplished by creating a new data adaptor to the second core, and using the modeling tools in the BPM, UI, and ECM to incorporate any unique data, account servicing, and business process requirements of the additional core. The Process API now incorporates the integration layer for the additional cores, allowing common functions to be operationally consolidated. Standardizing business processes across multiple cores begins to "break the silos" by enabling the establishment of centers-of-excellence for banking functions like account maintenance, regulatory compliance, CRM, audit, fraud detection, customer complaints, and loss mitigation. It is now possible to automate much of the work to update multiple cores behind the scenes after business process initiation, and do it without human intervention. Tasks like address changes, customer onboarding, or account holds are performed systemically for all the relevant systems attached by multi-core unification. Unifying cores also exposes data and business processes for all the cores connected to any channel through the Process APIs, enabling OMNI-product and OMNI-channel banking. Connecting the product silos and their associated servicing operations, ecosystems, and customer channels creates significant value for the bank and its customers. It also stages the bank for core and ecosystem consolidation to the new target systems.

Prepare for Core Conversion

The final efforts to prepare for the core conversion become technical tasks accomplished behind the scenes. With the core encapsulated and the ecosystem upgraded to its future state, the core conversion will not impact the majority of the bank operational staff. This reduces the remaining tasks to preparing for the conversion of data and interfaces for those systems that will remain after the transition. Most of the efforts up to this point have been in front of the core, upgrading customer-facing and bank operations applications. Now is the time to work behind the core, where the payment systems connect and various data feeds are delivered for financial and reporting purposes. Most payment systems communicate through industry-standard protocols that all cores support, making this task a simple process to ensure the new core is communicating and posting transactions properly. Replacing custom data feeds should also be straight forward since any new core will support standard query languages that can output data with the selected elements in the format required by the receiving application.

The new core will have a native data schema that needs to be translated into the DDM used in the previous phases. After testing the data adaptors and APIs, the new core is connected to a copy of the servicing layer and ecosystem to verify functionality. If new data items are available in the target core that are not yet in the DDM, this is the time to expand the DDM to incorporate these for future use by the servicing and ecosystem components.

The new core needs to be configured with the products and their behavioral characteristics for the customer accounts targeted for conversion. After the new core is set-up, transactions and time-triggered cycles are tested to ensure the calculations are correct for interest, fees, and the like.

The next task is to prepare a data conversion program for moving existing accounts, transaction history, and other files to the new core. Since DDM understands the native schema of both cores, using it as a conversion reference not only makes this task easier, but also verifies the accuracy of the data adaptors for the DDM.

The final step in preparing for the core conversion is running a series of end-to-end tests and following a test script for all connected applications, payment systems, data feeds, etc. to ensure that the results meet expectations. Use account balances and transaction files from the legacy core to test the new core and verify that it replicates the current production results accurately.   Perform trial conversions to test all systems, processes, and contingency plans. After checking all the boxes, it is time for the conversion itself. This usually occurs over a slow period like a weekend. The process is repeated for each core that is being sunset and consolidated or replaced with a new core.

Replace the Core

The last step is the actual conversion to the new core using live cut-over data, pre-tested data adaptors, payment system interfaces, and APIs. Use a go/no-go checklist covering all relevant areas to manage all the tasks and get the appropriate sign-offs required to perform the conversion. It is important to resist the temptation to enable new core capabilities or introduce other systems at the same time as the core conversion. It is also not necessary since the LEAP technology put in place earlier can quickly expose advanced capabilities of the core and its ecosystem *after replacing the core*.

Hybrid Approach Using LEAP

For a large bank, combining the benefits of a parallel bank with multi-core unification can potentially provide the fastest path to innovation and low risk legacy transformation.

A parallel bank is a "bank within a bank," and is very similar to the approach many large banks originally used when launching direct (Internet) banks back in the early 2000s. Bank One's Wingspan Bank was the first domestic bank to launch a new Internet bank using all new technology, including the core. Citibank's Citifi and ING Direct were also notable advocates of this approach. Aside from ING Direct, most of these initiatives were ultimately merged back into the parent bank and converted to the banks' legacy core technology. In the case of ING Direct, they flipped the equation and ended up converting their main banks in Amsterdam to the new Sanchez PROFILE core used to build their global network of direct banks.

Now, nearly 20 years later, banks are again trying to innovate around their legacy technology by launching parallel banks that offer a unique value proposition. A parallel bank is the fastest and easiest way to test new technologies, but the challenge remains how to convert the parent bank to the new technology even if it proves to be better. The parent bank has a more complex technology infrastructure with many more customers as well as a recognized brand that they don't want to put at risk in a botched conversion. The LEAP process and technology can enable the transition of the parent bank to the new technology of the parallel bank in low-risk, high-value stages. Launching a parallel bank using a new core coupled with the LEAP technology provides the foundation to later connect the parent bank to the parallel bank in a subsequent phase using the multi-core unification process. This allows a bank to innovate quickly with an entirely new technology stack. All of the benefits of multi-core unification described earlier exist after connecting the two bank systems. The resulting unification of the cores and ecosystems creates a digital bridge between the parent bank's legacy system and the new technologies of the parallel bank. This allows the bank to consolidate legacy components to a target technology at a low risk and without significant disruption or effort. Banks can likewise consolidate customer accounts on the target cores with low operational or technology risk

Shadow Core

Another transformation approach uses a concept that many large banks already have in place but with a major twist. Banks often use an Operational Data Store (ODS) that is a shadow database of one or even multiple cores to authorize and accumulate transactions from multiple banking channels. The database is updated daily to reflect the starting balances and other key information from the SOR. For banks with batch-based (non-real-time) cores this database does not negatively impact the timeliness of transaction updates since these systems update overnight and the ODS has accurate intraday running balances required for transaction authorizations. The twist that makes this approach viable as a core transformation strategy is to use the target core with its native data schema as the ODS prior to migration. The legacy core remains the SOR for the bank, but now the new core can be configured to not only authorize and accumulate OMNI channel banking activity within its' native database, it can also be configured to mirror the product behavior of the legacy core that is being replaced. The new core can use the same transaction information as the production core to run in parallel for testing purposes through multiple time cycles.

In addition, some of the functionality of the legacy core can be abstracted into the services layer of the target core (or it's supporting BPM/ECM layer) prior to conversion of the legacy core. New product launches can likewise be supported by the new core prior to conversion of the legacy core since it is already connected to the banking channels as the authorization engine. Back-end payment systems and data feeds can easily be connected to the new core to support new product types or for testing purposes without disrupting the legacy production environment. Once all the calculations and product attributes are confirmed to be in lockstep, the legacy core is "turned off" and the target core that has already been standing-in between the legacy core and the banking channels takes over as the new SOR.

.

LEAP and Outsourced Banks

Most of the larger outsourcers of core banking technology have announced plans or already offer some of their solutions through a new API architecture. This allows banks more choice and easier integration of ecosystem components from both within these supplier's inventory of offerings or from other third-party providers. The outsourcers could use the LEAP tools and process to standardize the layers around the cores with Process APIs and a new UI to standardize their client bank's internal servicing operations for all the similar cores. Using the multi-core unification process, the outsourcers could converge their own cores to a new target core without disrupting their banks. A pre-configured LEAP solution with pre-tested data adaptors to the DDM, Process APIs, upgraded servicing UI and a broad selection of ecosystem components would provide a very efficient and low-risk solution for these vendors to help their clients meet their transformation challenges.

The Last Word

I hope the processes and technology described in this paper provide some insight and answers for those seeking to undertake a core transformation project. LEAP unwinds the complexity of a core transformation by peeling back and replacing layers of the old technology with the next-generation technology in manageable and valuable steps. This process encapsulates the core and de-couples the surrounding legacy ecosystem, making replacements easier and less disruptive. The LEAP tools become the modern go-to servicing model for people and systems throughout the process. The operational risk is reduced since the bank's staff and its customers will already be familiar with the new ecosystem and servicing platform before the core is replaced. With this approach, the core itself becomes just another component that can be replaced without risking the entire bank. Taking the LEAP using the tools and processes described finally answers that burning question posed over 30 years ago by the CTO of The Chase Manhattan Bank.

About the Author

Michael Sanchez is the Chairman and CEO of Savana, Inc., a technology company specializing in enabling the financial services industry to transform their legacy technology. He also serves as President of Finxact, a next generation core-as-a-service company he founded with his brother Frank. Prior to Savana, he was the President of the International Division of FIS Global, which experienced significant growth during his tenure. He was the Founder and Chairman of Sanchez Computer Associates, the developer of the PROFILE banking system that is in use in over 40 countries and is the predominant system for Internet and Direct banks in the United States. He is the recipient of the 2001 Ernst & Young's Entrepreneur of the Year award for technology in the mid-Atlantic region.

About Savana Inc.

Savana is a technology company specializing in enabling the financial services industry to transform their legacy technology. Savana's bank transformation technologies include tightly coupled BPM, ECM, UI authoring tools along with bank-specific domain data models and bank servicing applications to help banks enhance existing legacy technology while positioning for future transformation. Learn more at www.savanainc.com.

About Finxact, LLC.

Finxact is the first enterprise class public cloud/private data Core-as-a-Service platform with a completely open core API. The Finxact core will allow banks to seamlessly deliver new services for today's omni-channel customer. The Finxact team has been at the forefront of banking software for 35+ years, revolutionizing the industry in the 1990s with the first real-time core banking solutions for Internet banks, and is now creating a new cloud-based core banking solution for today's real-time, digital-first world.
Learn more at www.finxact.com.

References

[i] Cognizant. (June 2013). Understanding Failed Core Banking Projects. Retrieved from https://www.cognizant.com/whitepapers/Understanding-Failed-Core-Banking-Projects.pdf.

[ii] NTT Data Consulting. (August 2016). Don't Fear Modernizing Your Core. Retrieved from https://us.nttdata.com/en/-/media/assets/whitepapers/apps-mod-banking-modernizing-your-core-whitepaper.pdf.

[iii] McKinsey & Company. (September 2015). The Fight for the Customer: McKinsey Global Banking Annual Review 2015. Retrieved from https://www.mckinsey.com/industries/financial-services/our-insights/the-fight-for-the-customer-mckinsey-global-banking-annual-review-2015.

[iv] BI Intelligence. (June 2017). Core Banking System Overhauls.

[v] Capgemini Consulting. (April 2016). The World Retail Banking Report 2016. Retrieved from https://www.capgemini.com/consulting/resources/world-retail-banking-report-2016/.

[vi] Deloitte. (2008). When legacy is not enough: Understanding the high stakes game of replacing a bank's core systems. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Financial-Services/gx-fsi-us-why-legacy-is-not-enough-2008.pdf