

A retailer's journey to digital transformation.

How to modernise legacy systems without disrupting operations

As digital technologists, we often pride ourselves on being rational beings, immune to oversimplifications and hasty conclusions. However, the truth is that we are only human, and our objectivity has its limits.

Our industry has numerous illogical beliefs and subconscious biases, which can affect our decisions. One frequent mistake is attributing the actions of others to their personal qualities and disregarding the circumstances. This tendency is also visible in how we view outdated systems.

When evaluating the people who created these legacy systems, we often rush to swift judgments and tend to underestimate the complexity of both the technological and business aspects of the domain. It's crucial to recognise that legacy systems are products of their time.

The mere fact that legacy systems become a concern indicates that **they fulfilled their purpose: they served us long enough to become old**. Only some systems get that chance. Therefore, it's essential to treat outdated systems with the respect they deserve, even if our only course of action is to replace them. This way, our transition will be smooth.

And that's where this whitepaper comes in.

Discover VirtusLab's strategy for digitally transforming critical elements of a legacy system while ensuring the continuity of daily business processes. The journey centres around our partnership with a significant retail industry player, where we revitalised an essential segment of their legacy system.

You'll discover:

- The crucial factors that influenced this digital transformation.
- An extensive overview, exploring the technical and business challenges faced.
- The implementation and the solution of the challenges.
- The outcomes three years after the migration show the key changes throughout the business process and how they've improved our partners' ability to compete effectively in the marketplace.

 $/_2$

Table of contents

Why IT systems fail: complexity and legacy	4
The 1980s: establishing the cornerstone	5
The 1990s: going relational	6
The 2000s: first approach to modernization	7
Recognising the need to modernise	9
Navigating digital transformation with surgical precision	11
How to map uncharted territory	12
The observation stage	12
The information gathering stage: interviewing people close to the system	14
The information gathering stage: checking the old system's ability to solve current challenges	14
The information gathering stage: the results	16
How to begin the digital transformation	17
A modern approach to digital transformation	20
A modern approach to digital transformation Product driven-development for modernising legacy systems	20 20
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems	20 20 21
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems Understanding data consumer challenges	20 20 21 22
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems Understanding data consumer challenges Quick steps to the initial MVP	20 20 21 22 22
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems Understanding data consumer challenges Quick steps to the initial MVP Incremental adoption and user trust	 20 21 22 22 23
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems Understanding data consumer challenges Quick steps to the initial MVP Incremental adoption and user trust The journey from proof of concept to primary source	 20 21 22 22 23 23
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems Understanding data consumer challenges Quick steps to the initial MVP Incremental adoption and user trust The journey from proof of concept to primary source The importance of legacy systems in day-to-day business	 20 21 22 23 23 26
A modern approach to digital transformation Product driven-development for modernising legacy systems Navigating digital transformation amid legacy systems Understanding data consumer challenges Quick steps to the initial MVP Incremental adoption and user trust The journey from proof of concept to primary source Legacy business processes engineered using today's technology	 20 21 22 23 23 26
A modern approach to digital transformation	 20 21 22 23 23 26 29
A modern approach to digital transformation	 20 21 22 23 23 26 29 30
A modern approach to digital transformation	 20 21 22 23 23 26 29 30 32

 \square

Why IT systems fail: complexity and legacy

To help bring clarity, we'll use an analogy. In the early twentieth century, New York's subway system grew dramatically as part of a broader surge in infrastructure development.



The subway system began with one line in 1904. By the 1920s this had grown into a vast network covering hundreds of miles.

The city's approach to public transport was hailed as an engineering marvel. It served as a model for the rest of the world, particularly in the rebuilding of war-torn European cities after the Second World War.

Since 1940, however, no new subway lines have been opened in New York, apart from a few minor extensions and connections. Why did this happen? New York's subway system had become so large and outdated that almost all of the available funds were spent each year on maintenance. As a result, there are limited resources left for major infrastructure investment.

In addition, launching a major initiative requires broad public support and a political champion. Maintaining the status quo proves to be the easier option, even though improving public transport would benefit both the city's residents and the environment.

That's exactly what happened with our partner's legacy system.

So how did it get to this point? Let's take a look.

/ The 1980s: establishing the cornerstone

IBM mainframes

Our partner is an experienced multinational retailer. At the time, they were keen to keep pace with technological advances and recognised the transformative impact of computerisation on business. In the 1980s, they pioneered technologydriven innovation, a strategic move that put them well ahead of their competitors. These initiatives were driven by IBM's robust mainframes, which were a powerful choice at the time.



Across various departments, office tasks were computerised, such as automating shift management and optimising transport routes. Over time, the system became more complex. Some of this complexity was intentional, resulting from additional process automation. However, unintentional complexity crept into many areas.

System growth vs digital evolution

The 1980s was a period of many acquisitions for the company. As a result, the retailer had to update numerous internal systems and integrate them with other technologies. In addition, they incorporated various external vendor systems into their operations.

By adopting 1980s technology, they became pioneers with a crucial competitive advantage. But it also had its drawbacks. New competitors in the 1990s and 2000s had access to the latest technology, allowing them to start afresh without the burden of legacy systems. This meant they were able to iterate quickly to meet the evolving needs of their customer base.

/ The 1990s: going relational

Oracle database & extract-transform-load mechanisms.

The original mainframe approach required a transformation in the 1990s. As a result, our partner introduced an enterprise-wide Oracle database on top of the existing tiers to democratise access to data.

This database was backed up by managed extract-transform-load (ETL) mechanisms, which played a critical role in coordinating database updates while ensuring synchronisation with the mainframe system.



As the business expanded, practices from the original headquarters were replicated in other countries. Let's remember that **in the 1990s and early 2000s**, **there were no prominent cloud providers and no established patterns for such global expansion**. Building system architecture in new countries involved duplicating systems, a process known as 'copy-pasting'. In particular, region-specific changes were coordinated individually, leading to rapid divergence between systems.

Each country had its own **independent IT department**. However, this independence **increased costs and time-to-market**. Introducing new features became a repetitive task, requiring multiple implementations in different regions. Even within a single region, the SQL/ETL-based solution presented data ownership challenges. When multiple administrators were involved, it became difficult to determine ownership of specific tables or data fragments.

As the system's database aged, the lines of responsibility became blurred. While the original architects created clear boundaries, these divisions became blurred over the years. The introduction of new features became a complex process akin to labour negotiations. Programme managers ended up acting as mediators between the development teams involved.

Managing the system felt like walking through a minefield. Even when managers identified an obvious error in the dataset, correcting it was a risky decision. Addressing such issues required extensive research, blurring the lines between an error and a deliberate tactic.

There was an urgent need for further improvement, focusing on a clearer separation of domains.

/ The 2000s: first approach to modernization

Logical Domain Separation

Our client identified the design problem in the 2000s, prompting the first attempt to fix the architecture. To ensure full domain ownership, our partner established a 'single point of truth' with an independent team acting as an overseer.

The technology chosen was the widely used Lotus Notes. This proved to be much more adaptable and user-friendly than the mainframe systems, which required specialist knowledge to operate.

For those who never heard of it, Lotus Notes was a Microsoft Access-like tool. It combined the functionality of a simple end user interface, with a competent database, and a built-in scripting environment.



This innovative approach was undoubtedly a step in the right direction:

- The system as a whole became much more consistent.
- Updates were distributed efficiently to all relevant parties.
- A single, dedicated entity had complete and accurate knowledge of the current state of all sites.

However, the final resolution remained less than optimal. The goal of a single source of truth was never achieved, despite the original intention to completely replace the mainframe computers. The project was delayed and its scope had to be reduced.

As a result, no changes were made to the communication protocol itself. In addition, none of the old system components were completely replaced. Instead, an additional layer was introduced. In fact, this new solution added even more complexity to the overall scenario.

/ Recognising the need to modernise

We showed how our partner's initially successful changes eventually became a burden to the business. No one person made any explicit mistakes; **the system simply aged**.

During this period, old competitors disappeared, making way for new, aggressive players with technology better suited to the challenges ahead. These new players used their advantage to slowly take over our partner's market share.

Meanwhile, our partner's system resisted evolution. Implementing any change was a time-consuming process, with a software quote taking a staggering three months! Recognising the need to modernise in order to survive the next two decades, our partner sought our advice.

In the next section, we look at the modernisation process. You'll discover how we approached digital transformation from the ground up, and the techniques we used to overcome each challenge along the way.

9

In today's fast-paced business world, digital transformation is essential for companies to remain competitive. By incorporating modern digital technologies into their operations, businesses are transforming their processes, improving customer value, increasing revenue and improving planning and decision-making.

Let's explore the importance of digital transformation and how a service partner can provide valuable support.





10

Navigating digital transformation with surgical precision

Over the years, a combination of effective and ineffective choices had contributed to the system's increasing complexity. It had reached a critical juncture where the existing technology jeopardised the company's competitiveness against modern, agile rivals.

Our partner wanted to recapture the innovative spirit that had driven the company's success in the past. So they turned to VirtusLab for help.

We aimed for true modernisation rather than a superficial overhaul. Our focus was on uncovering new, sustainable opportunities for our partner. This required a deep understanding of the client's IT landscape and business processes.

Each layer of the system presented unique challenges. As a result, a thorough analysis was essential to make informed decisions about what and how much to cut.

Focusing on efficiency alone can sometimes jeopardise the entire digital transformation initiative. It can be tempting to abandon iterative approaches and preparatory steps in order to speed up the outcome. However, a 'get the job done' mentality is rarely effective in highly complex situations. When faced with numerous unknowns, the first step is to prioritise continuous learning and understanding.

What does 'digital transformation' mean?



The term 'digital transformation' refers to the widespread integration of digital technology into all facets of a business or organisation, resulting in significant changes to its operations. This transformation involves the integration of cloud computing, data analytics and artificial intelligence to improve operational efficiency and decisionmaking. It is also **characterised** by an imperative cultural shift towards adaptability and a strong focus on meeting customer needs.





Legacy systems are like a complex jigsaw puzzle: it is difficult to assess the scope of the project in advance. The architecture of the system, including the relationships between its components, is initially unclear. The confusion extends to the specific value that different parts of the system contribute to the business.

The challenges become particularly daunting when a company has merged with others or is dependent on external vendors. These integrations are the hardest to track and the easiest to overlook.

Unfortunately, there are situations where a broad understanding is critical to a successful transformation. In this case, given the client's outdated infrastructure and multiple layers of architecture, understanding the system relationships was critical to the success of our mission.

To tackle the problem, the first step was to become a cartographer and meticulously map the entire environment. At this stage, a different perspective was essential. Using research and analytical skills, we learned about the whole system through three main actions:

boserving the operation of the current system both past, and present of the system, both past, and present of the first two actions.

Although these tasks are described separately, ideally they should be carried out simultaneously as they complement and reinforce each other.

/ The observation stage

Documentation becomes outdated. Human memory is unreliable. Even if it were reliable, the current workforce didn't create the original system. So relying on the operational production system was key. However, there was a challenge: the components we needed to understand were obsolete. So the question arises:

How do you 'observe' a system that was built long before the concept of 'observability' existed?

You approach it like unearthing ancient artefacts, meticulously piecing together the puzzle of the past. To do this, we used two complementary methods that we call White Box and Black Box observations.



White Box observations

You may have guessed that our White Box and Black Box terminology is inspired by similar names used in testing strategies. Much like White Box testers inspect the internal state of a system, our White Box observations work in a similar way.

First, we observe the operation of the system from a machine perspective. While this may sound simple, this approach is critical to understanding the activity within the local machines and ETLs.

Examining the files generated on the box and the records added to the database is time-consuming, but yields surprisingly insightful results.

Consider a scenario where a system consists of discrete, unrelated steps with no central scheduling. If you want to understand the sequence of operations, having access to specific machine-generated log files allows you to compare them. It allows you to uncover temporal relationships. This involves a touch of artistry alongside scientific analysis, making it a valuable technique to have in your toolkit.



Black Box observations

In contrast, Black Box observations are invaluable when we have no access to the state of the machine,

such as when the system lacks small intermediate steps and useful logging.

In these scenarios, several components of the system are distributed in binary form without access to the source code. As a result, intricate details of their behaviour have been lost over time.

Fortunately, by observing the outputs of the system for specific inputs, we can capture some exceptional cases. However, covering all possible input-output combinations is labourintensive and challenging without a production-like testing environment. Ultimately, this practice is risky. However, it becomes necessary when other alternatives fail.

Even if certain systems don't have a true pre-production environment, there is often the option of 'dry runs' of various operations. We discovered that this dry-run technique was a standard feature of systems from the 80s and 90s, and significantly accelerated the iteration time between test waves. Having access to this functionality makes Black Box observations an invaluable asset.

/ The information gathering stage: interviewing people close to the system

Observing systems like this allows us to understand how specific components of the system work, but it often doesn't tell us much about how those specific components are connected to each other. **To learn about the subtle connections between different subsystems, you need to observe even the smallest details**.

At the start of a project, it is often unclear what to look at and what questions to ask. You start by asking simple questions, gradually refining them as your understanding of the system deepens.

Gradually you start to connect the dots by talking to people. It's valuable to seek information from different departments, as knowledge is dispersed in large organisations. In addition, digging into the corporate database, such as Sharepoint, will provide insights.

Interestingly, older systems from the 1990s and 2000s often have some form of design documentation, even if you do not have access to the source code. However, it's important to be careful: these documents can be misleading, as they may have been written long before the last changes were made to the system. It's important to check any information you find against the observations mentioned in the previous section.

/ The information gathering stage: checking the old system's ability to solve current challenges

Many believe that understanding the basics of the technology layers is sufficient. However, this understanding is only a partial and flawed representation of the real business requirements.

14

Ask the hard 'why' and 'what if' questions, such as:

//

//

Why does the system continue to collect data from different dependencies when it can be retrieved from the central data warehouse?

Is the business process that previously used this data still active? It seems we're transferring this information every day, but the component that used it was deactivated five years ago.

This system has only one consumer, right? What if we developed a simple connector? That way we can take three machines out of service and reduce the processing time from a week to two days.

There is a common belief that technology evolves faster than organisational change. However, perhaps due to the perceived complexity of software systems, IT systems exhibit surprising inertia over the long term.

As market dynamics change, organisations adapt, but their technology infrastructure still reflects outdated realities. The typical response is to patch the existing system to fit the new landscape. However, it is not enough to focus solely on modernising existing systems.

This is where an external perspective proves valuable. Consultants identify opportunities that are hidden from internal staff. They may think they are better engineers or have more experience. The simple truth is that people who deal with the system on a daily basis may have a "can't see the forest for the trees" perspective or a "we've always done it this way" mindset.

Outside operators have the advantage of approaching the problem with a clean slate. If they have any marks on their page, it's because they've solved similar problems in previous cases. With curiosity and persistence, consultants regularly uncover crucial details that inform important decisions, sometimes leading to significant leaps forward.

/ The information gathering stage: the results

Using all the actions we discussed, we created a map of unexplored territory. Note that the map is a real one, but the labels have been removed due to our NDA.



Well prepared, we entered into a thoughtful discussion with our business partner. The groundwork we had laid enabled us to work together to identify the most pressing needs. Our map served as an excellent reference tool, helping us to assess the processes we could easily address. Together, we began to define our transformation goals.

16

How to begin the digital transformation

When you're building software, there's always more work than resources. To start our digital transformation, we had to prioritise our tasks. Our aim was to help our partner regain lost innovation capabilities. That's why we focused on improving one key area: Location Management.

Location Management



In retail, a store serves as a central hub for many processes, especially for tasks such as setting prices for goods or services. While manually managing a few stores is feasible, managing pricing for thousands of locations becomes a complex task.

The challenge arises when factors such as geographical differences, such as urban centres versus rural areas, require different pricing strategies. Therefore, the implicit knowledge held by administrators needs to be transformed into digital logic. This involves organising stores into different pricing tiers, each managed by different business departments. You might ask yourselves: Why did we choose this process to transform first?

There were three reasons:

A vital business process

Of all the challenges the business faced, the process of opening new stores was the most demanding for the team. The 'store' was a fundamental concept to our partner's business and was integrated into the IT systems from the very beginning. Each layer introduced over the years had to handle and respond to events related to the creation of new stores. As these layers evolved, complexity increased.

Streamlining location management could make a significant and lasting difference to the company. Despite the vital nature of the process, the existing software system did not meet the company's requirements.

- The only way the systems communicated was through weekly batch exports, resulting in painfully slow data circulation.
- Updating information throughout the system took weeks, resulting in costly delays.
- The rapid advance of new competitors underlined the urgency of making internal processes faster and more efficient.

Do you want a concise overview of Location Management, the challenges and the solution? Then we encourage you to read this success story:

Enterprise API providing location data for a global retailer



Using our map, we decided that the best place to start was the current entry point to the system: the Lotus Notes database. Although it was considered to be the primary source of location data following a migration in the early 2000s, it had significant issues that required urgent attention.

The Lotus Notes database, although critical, was hosted on a single server without adequate disaster recovery measures. Even minor maintenance tasks posed challenges. For example, changes to the Windows update process once led to concerns about the machine's ability to reboot. Another ongoing concern was the limited lifespan of the server's old hard drive.

Ever-changing business environment

The competitive environment alone was a strong incentive to invest in modern, flexible technology solutions. However, additional pressure came from various governments introducing regulations on how data is stored in IT systems. GDPR, in particular, proved costly to implement, especially for older systems where it was more difficult to establish appropriate policies. Of these, **Location Management** being one of the oldest systems **required significant effort to ensure compliance** with the new legislation.

Against this backdrop, we needed to modernise without breaking anything. Transforming an existing system is a complicated task. In most cases, the legacy system is the one that generates revenue. It's important to maintain the functionality of the organisation throughout the process. That's why we divided our work into two simultaneous processes:



The strategic solution

Developing a new master source that incorporates modern, cloud-enabled and non-functional requirements.



Implementing a new development process

Enabling the legacy consumer to access data from the strategic solution.

A modern approach to digital transformation

In order to modernise the system effectively, we had to demonstrate the viability of our solution. Therefore, we developed a proof of concept.

/ Product driven-development for modernising legacy systems

To create a competitive solution, we decided to use a technique that isn't usually associated with replacing legacy systems: product-driven development. Our team consisted of a few engineers and a dedicated product manager.

The product manager analysed the requirements to identify critical needs, looked for opportunities and handled communication. Meanwhile, the engineers collected system usage metrics and iterated to improve the solution. Their goal was to deliver a solution that addressed key business pain points and was flexible enough to evolve.

In discussing technology stack options, we discovered that the company's standards were Java, Vert.x and MongoDB, so we created a walking skeleton, a basic framework that was compatible with these technologies.

It could be deployed on the partner's cloud system of choice, which was Amazon Web Services. We set up all the necessary processes and created the first iteration of the new API. We quickly came up with a minimal implementation. In other words, a minimal viable product (MVP) that could be easily extended in any direction. At this stage, the API was standard CRUD (create/read/update/delete). This deliberate simplicity allowed us to engage in discussions with external teams and provide a tangible starting point for understanding their needs, rather than abstractly mentioning that we had an "API with locations".

/ Navigating digital transformation amid legacy systems

We looked for teams that were facing specific challenges with legacy systems in order to drive digital transformation. Relying on legacy systems as a primary source of data can hinder organisational initiatives in a number of ways:

Opportunity costs

Launching new initiatives with legacy data can be challenging. When data is difficult to access, teams often delay or avoid initiatives that could add value for customers.



Redundant work

Creating new solutions, such as using the cloud to work with legacy systems, requires building adapter layers to access the data. This adds unnecessary cost without significant value to the end user. This approach also needs to be repeated for each system, leading to duplication of effort.

Data inconsistency

In some cases, teams may choose to replicate data. It's a common practice for creating independent services, but it needs to be managed carefully. This method works well if the primary source is carefully structured and informs downstream services of changes. However, without proper preparation, there is a significant risk of data inconsistencies resulting from this approach.

/ Understanding data consumer challenges

Our solution helped manage the fragile integration with the on-premises system, enabling our partner to achieve several objectives:
Eliminate the technical debt associated with duplicate adapters
Eliminate the source of inconsistency
 Open up opportunities for future initiatives

Although our initial solution contained only placeholder data, it served as a starting point for discussions with potential early adopters. Each meeting provided valuable insights, and the placeholder data could be easily updated following exploratory sessions. Through this iterative process, we developed an API contract that addressed the basic requirements of these users and gained sufficient knowledge to propose the first substantial iteration.

/ Quick steps to the initial MVP

While the original specification highlighted the need for a complex enterprise system, showcasing easily accessible information in a user-friendly format effectively conveyed the solution's worth. We could share this information without implementing cumbersome authorization and access procedures. This approach also aligned with the project objectives.

This quick prototyping method let us exclude expensive features like authentication and authorisation. Even though these features would be necessary for more complex future usage, avoiding them initially meant we could speed up our time to market. At this point, our focus was on enhancing the learning process and showing a return on investment for our customer.

Once the first version of the solution was ready, data was replicated from the Lotus Notes database to the new system. Administrators began implementing changes in both systems simultaneously: the new strategic system (using a MongoDB database) and the Lotus Notes database serving existing legacy users. This process was challenging, but necessary to thoroughly test compatibility with the legacy systems - details of which we will discuss shortly.

During this phase, we developed a simple user interface using React.js that mirrored the layout of the old Lotus Notes interface. During the development we considered several ways to improve the outdated form structure. However, we decided to postpone these improvements.

/ Incremental adoption and user trust

Introducing new tools and updating the visual design is a significant change for the people who use the system. They tend to resist changes to an already established user experience. New user interfaces can also lead to confusion and increased workload for the administration team. Therefore, we wanted to build familiarity and trust before proposing a redesign.

Our new service started with just three users, which may seem small. However, our strategy was to start with a limited user base. This approach allowed us to identify infrastructure and solution availability issues early on, without disrupting essential business operations. This process helped us to strengthen our solution and prepare it for future growth.

/ The journey from proof of concept to primary source

As we shipped the first production release and implemented key administrative functionality, our focus shifted to the future.

Our product manager began to engage with additional users and explore potential new applications. On the analytical and programming front, we deepened our understanding of the domain and introduced new data types and entities. Through these efforts, we began to surpass the capabilities of the longstanding legacy systems.

The challenge of the legacy code went beyond the Location Management domain; it was a company-wide issue. Leveraging our programming expertise and recent experience with the Location Management system, we seized the opportunity to help our client modernise their platforms. By bringing other specialists into the team, we worked together to develop a modern alternative to replace the legacy systems. This alternative was designed to be:



We seamlessly integrated with other services within the corporate network, including an authentication API for centralised permissions management and a geolocalisation API that allowed us to offload the complexities associated with this domain. These integrations became integral parts of our interconnected system, consistently delivering new business value on a weekly rather than quarterly basis.

In the process, we gained trust and confidence and established our service as the primary source of location information.

As a result of our enhancements, we needed to launch a new version of our service, which required a transition from the existing service. Transitioning customers from one service version to another is often a challenge. However, as we were continually adding valuable features, there was plenty of incentive to upgrade quickly.

An unexpected benefit was that these new features proved valuable even to clients who initially thought the original version met all their needs.

In today's fast-paced business world, digital transformation is essential for companies to remain competitive. By incorporating modern digital technologies into their operations, businesses are transforming their processes, improving customer value, increasing revenue and improving planning and decision-making.

Let's explore the importance of digital transformation and how a service partner can provide valuable support.





25

The importance of legacy systems in day-to-day business

The thrill of building "a whole new world" was tempered by the realisation that legacy systems could resist transformation. The legacy system would remain the backbone of the business until our new system achieved feature parity with the existing software. At this point, migrating all existing clients to the Cloud API was not feasible due to the significant investment required.

The Lotus Notes application served as the primary data entry point for the legacy systems, using protocols from the early 2000s. Its exports were critical for clients in the on-premises data centre, which were primarily accessible via the corporate intranet and difficult to access from cloud environments.

To avoid a "split brain" scenario and data inconsistencies, we recognised the importance of properly maintaining these legacy systems. Treating them as a priority was essential to establishing them as a true master source of location data. We therefore opted for a hybrid approach, which led to the development of an application called Phoenix.

/ Legacy business processes engineered using today's technology

The Phoenix system served as a new version of the on-premises box in the data centre - in other words, it served as a rebirth of the system it was designed to replace. Phoenix was to handle the imports and exports that had been handled by the old Lotus Notes system for the previous decade. It replicated the behaviour of Lotus Notes, which operated as a pull-based system, generating export files following predetermined paths.

Phoenix functioned as a sophisticated and well-monitored CRON system. It included essential technical features such as:

- Exponential backoff with retries
- Logging to the global monitoring system with appropriate alerts
- Disaster recovery

Although built with modern technologies, Phoenix still maintained links to the legacy world as it was deployed in the on-premises data centre. At the same time, we mirrored the logic of Lotus Notes within the strategic solution. This presented some interesting challenges.

Unforeseen challenges with large file sizes



When attempting to swap a file for

a backup, specifically a large binary file exceeding 100MB, all went well during our trial assessment. However, upon shifting the program to the cloud staging, calls from Phoenix to the cloud API often timed out. The cause of this issue was a global timeout set at the API gateway level, a parameter that we were unable to adjust.

For context, the previous system used a simpler approach: it created large files beforehand and transferred them at set times using FTP. This method worked well, as FTP is more appropriate for large files than HTTP.

We couldn't modify the timeout value at the data centre, so we adopted this approach and designed an offline cloud-based solution. This resolution involved making the necessary files beforehand and keeping them in an external BLOB storage. We specifically used Amazon S3 for this. Although this temporary fix solved the issue, it revealed how a seemingly uncomplicated process in the old system led to unforeseen communication difficulties in the modern one.

We designed Phoenix to use legacy communication protocols such as XCOM and basic FTP (not FTPS or SFTP). Despite its legacy connections, Phoenix also acted as a proxy with Internet access. This proxy capability allowed Phoenix to retrieve export files from the strategic solution.



In essence, Phoenix performed the following tasks:



A major benefit was how the creation of files was completely transparent for users, which proved to be unexpectedly crucial while we tested our solution.

/ Rigorous testing to ensure harmony between new and old

As mentioned earlier, legacy systems often contain workarounds and quick fixes. This meant that rewriting the Lotus Notes export code line by line meant introducing errors by replicating these hacks. To identify and fix these issues before the system went live, we asked the administration team to simultaneously update system entities in both Lotus Notes and our strategic solution using the aforementioned React.js GUI.



Every day, the exports generated by Phoenix were compared with those generated by Lotus Notes. By continuously analysing the exports generated by Phoenix and the Lotus Notes exports, we were able to identify potential discrepancies. If there were any problems, Phoenix would log an error, which was then reflected in our monitoring system. While there were occasional false positives, mainly due to data entry errors, this testing phase played a crucial role in strengthening the solution.

/ The final migration

Once we had achieved a satisfactory level of certainty, we initiated the migration process in two phases:

Phase 1 Initially, during Phase 1, we guided users to obtain exports from the Phoenix server rather than the Lotus Notes server. These files were extracted directly from Lotus Notes and then stored in a designated location on the Phoenix server.



Although it may seem unconventional, our aim was to assess the connectivity of the system and ensure that the necessary permissions were in place. Anyone familiar with legacy systems will understand the unexpected challenges that the network layer presents. This phase allowed us to identify issues relating to data centre firewalls, amongst other things. By the end of Phase 1, we were confident that successful connectivity was in place.

Phase 2

Phase 2 saw the start of the generation of export files using the strategic solution on a client-by-client basis. The export files were now served from the strategic solution, accessible via the agreed path on the Phoenix system. During this period, export files continued to be downloaded from Lotus Notes to Phoenix on a daily basis.



This dual approach provided a safety net: in the event of a failure, users could quickly revert to the Lotus Notes version. Having this fallback option available allowed us to troubleshoot without the pressure that often leads to hastily implemented system hacks.

Phase 2 took about two to three months, mainly because our downstream process had to test everything with our consumer downstream processes. But in the end, we migrated all our clients to the data feed from the new cloud solution.

Decommissioning the legacy system

Upon completion of the two migration phases, all legacy users were transitioned to using data from the new system. Once our client was satisfied with the outcome, we decommissioned the Lotus Notes database, making the strategic cloud system the sole source of location data for the entire organisation.

By implementing the new system, we created a single master data source that met the needs of modern consumers while supporting nearly one hundred legacy consumers through the Phoenix environment.

This approach enabled hard-to-rewrite solutions to access data through a system with strong disaster recovery and advanced monitoring. Most importantly, it ensured consistent data for both the existing system and new projects, creating new opportunities that were previously unavailable.

Conclusion: digital transformation changed the mindset

The process described began three years ago, marking the start of a significant transformation for the retailer.

We started with three consumers and only a few transactions per second (TPS). Within three years, this number had grown significantly. The Strategic Location Management system is now used by over a hundred business processes, handling dozens of TPS on average and seamlessly managing up to 500 TPS at peak times. It has become a vital component of the entire platform, with its high availability and 99.99% uptime being of paramount importance.

While Phoenix continues to serve some legacy consumers, their demand has declined significantly since its introduction. Our partner, an international retailer, faced similar challenges in its regional centres. The approach we developed at the head office inspired these regional centres to replace their legacy systems. Our method of connecting the strategic and legacy worlds proved versatile and enabled similar initiatives in these regional centres.

Over the past three years, we've shared knowledge, exchanged experiences and transparently addressed common challenges across the business. As each country regains control of its data and technology, we are preparing for the next big step: creating a versatile solution that can meet the needs of international clients.

Modern, cloud-native architectures significantly reduce support costs and speed time to market for innovation, which is critical in today's hyper-competitive world. These are some of the tangible benefits of applying product management techniques to digital transformation. Soon, any improvement made in one country will be available to the entire organisation.

However, the most significant benefit of this digital transformation journey has been a mental shift. Working with the legacy system was complex and error-prone, and people were reluctant to take bold steps despite our partner's history of innovation.

Today, of the one hundred users of our system, only a handful are working to change existing processes. Most are creating new integrations, each representing a unique business opportunity, many of which might not have started without democratised access to data. Providing a reliable set of tools has empowered individuals across the organisation with promising business ideas.

There is a common assumption in the industry that large companies are slow and lack innovation. We want to dispel that notion. Large organisations have skilled people who are eager to drive business development - provided there's an accessible technology layer to support them. With a robust level of trust between business and software engineering, flexibility can be restored even in a long-established, large organisation.