

# Migrating or Integrating Your Existing Data:

How Beacon Platform Brings  
New Functionality to Legacy Systems



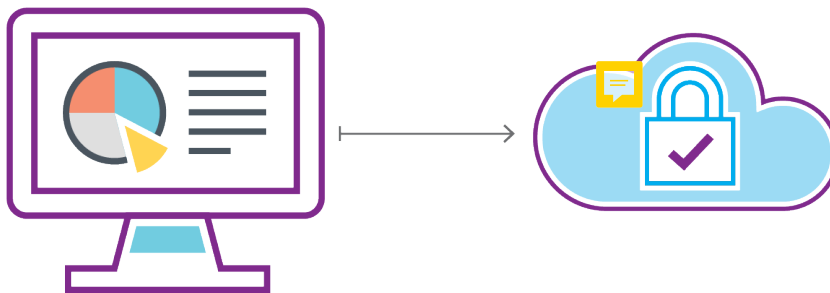


<b>You're perfect, now change</b>	<b>3</b>
But change brings risk	
<b>Migration versus integration</b>	<b>4</b>
Dealing with different types of data	
Trade data	
Reference data	
Market data	
<b>Integration: Building connectivity between new and existing systems</b>	<b>5</b>
Excel to Beacon example	
<b>Migration: Moving everything to the new platform</b>	<b>8</b>
Typical Beacon implementation	
<b>Benefits and best practices</b>	<b>9</b>

# You're perfect, now change

There comes a time in every software relationship where you want something that your existing systems cannot provide. It could be that your in-house development team cannot keep up with the demands for new functionality or higher performance from a growing user base. Or your purchased tools are a black box that you cannot see inside or change how it works. Perhaps you have an older system that is approaching end of life with limited support and upgrades. Or your Excel spreadsheets have become so complex over time that they are fragile and difficult to maintain. The best time to address these issues is before they start impacting your business by slowing your response to market changes, delaying investment decisions, or introducing process errors. The next best time is now.

Once you decide that you need to make some changes, you need to figure out how to best leverage your existing data. [This paper explores different methods of bringing modern, agile, cloud-based capabilities to your organization, without losing the wealth and variety of data and expertise that you have accumulated.](#)



## But change brings risks

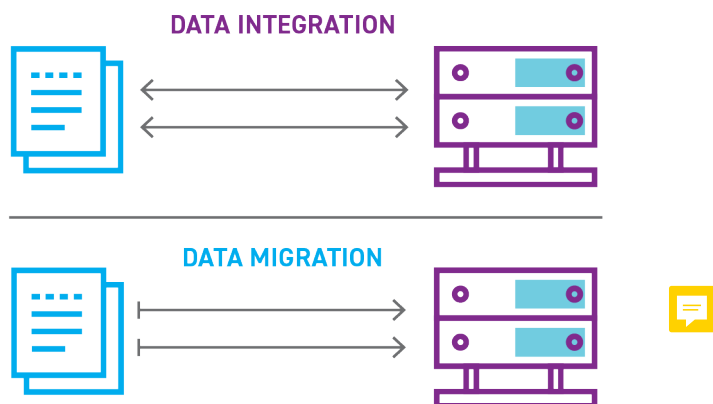
Of course, change introduces potential risks to the organization. Concerns about execution, that the project will be late or over budget, or that the investment in time and money will not produce the desired results, are probably the most common fears. Other typical system migration concerns include vendor lock-in, lack of functionality or flexibility, and suboptimal performance.

From a people perspective, some of the team may worry that moving to a new system will negatively impact their jobs, or worse, put them out of work. Users may be anxious about learning new tools and how their performance will be affected.

The answers to most of these issues are to start with a small project and scale up, clearly define the requirements, test and test again, and engage experts with relevant industry and technology experience to assist.

# Migration versus integration

It is important at this point to define the difference between data migration and data integration, and decide which method you are going to use. Integration involves building programmatic connections between the new and existing systems, and using some functionality from both systems. Migration means moving most or all of your data from the old system to the new one, and at some point turning off the old one.

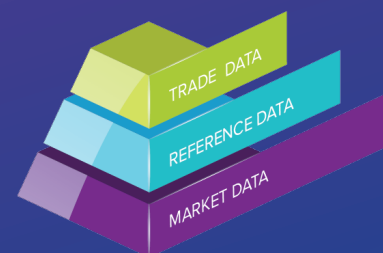


Integration is often a phase along the path to full migration. Managing a bunch of interconnected systems can be much more difficult than managing one platform, because of the complex data exchanges and mappings between all of them. Clients typically start out with some data integrations so that they can deliver new functionality that the legacy system cannot easily deliver. Over time, they add more capabilities and complexity, until the cost/benefit equation tips in favor of the new platform. A key project decision is how you are going to add new system functionality during any transition phase. It is also fundamental at this stage to separate the tasks of moving data from developing application functionality, and deal with those jobs independently.

For example, you have an in-house or vendor-purchased system for booking and tracking investments. The initial phase of a Beacon migration project adds connections to the existing data, starts mirroring the deals, and you get quick added value from **Beacon's Front Office Suite** and your own internal development. As a second phase some Beacon clients choose to migrate their data fully into Beacon, building the upstream and downstream system connections, and then retire their legacy system.

## Dealing with different types of data

- > In financial applications there are many different types of data that have distinct migration requirements, such as trades, references, and markets.



### Trade data

- > Trade or transaction data may be considered the core of what this paper is discussing. Typically proprietary, this information includes all of the firm's transactions and contractual details. What was bought or sold when, with whom, for how much, under what terms and conditions. This data is highly dynamic, and migration or integration decisions have to carefully consider timing and execute a lot of tests before going live.

### Reference data

- > Reference data is all of the additional information needed to analyze or book a trade. This typically includes definitions of units, counterparty details, instrument and security definitions, and even calendars for holidays and other notable events. This information tends to be less dynamic than trading data, and so may be easier to capture a snapshot in time for migration purposes. However, it also tends to be less structured, with many similar references. For example, "Acme Inc.", "Acme, Incorporated", "ACME incorporated", and several other variants may all refer to the same company. Cleaning up data like this can be very time consuming, and it is important to evaluate the best time to do this. There is also an opportunity to remove or skip references that are no longer relevant or valid.

### Market data

- > Market data, such as bid and ask prices, open and close details, and volumes is typically delivered as a feed, with various options for querying historical data. Many platforms include a variety of market data feed options, or can be easily modified to integrate your preferred feeds.



# Integration: Building connectivity between new and existing systems

To build data integrations between two or more systems, there are a few questions that you should consider:

- 1. How frequently do you need to move data?** Some applications may only need their data refreshed once a day, while others need to schedule multiple transfers per day, or run queries in real-time in response to user interactions.
- 2. Does data flow just one way, or is there a need for bi-directional communication that returns calculation results or updates prior values?**
- 3. What method will be used to communicate between the systems?** The good news is that there are many different methods and protocols available to meet your requirements. Most commercially available systems have a selection of data connectors, and custom adapters can typically be built in a few days.
- 4. How complex are the reconciliations between systems?** Robust reconciliation and exception reporting is a critical part of integrating production systems.

## INTEGRATION PROTOCOLS


### OPTION 1: EMBEDDING PROPRIETARY LIBRARIES WITHIN BEACON

- Client libraries in C/C++, Python, JavaScript, Java, R, and other languages
- Applications combining Beacon code and client libraries
- Develop, deploy, and support libraries with Beacon Core
- Docker images automatically deployed and scaled within Beacon

### OPTION 2: EMBEDDING BEACON WITHIN PROPRIETARY APPLICATIONS

- Remote Procedure Calls (RPC) from proprietary applications to Beacon
- Integrate C#/.NET applications using Beacon's C# API
- Call Beacon functions from within Excel with Beacon's Excel add-in
- JavaScript/JSON API for native integration with server-side frameworks like Meteor

### OPTION 3: SHARING DATA BETWEEN BEACON AND OTHER SYSTEMS

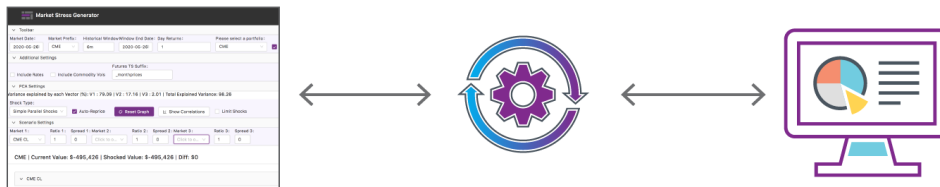
- Seamless cloud/internal data center communication via IPSec VPNs 
- Redis, RabbitMQ, RESTful APIs
- SFTP/SCP to existing PostGRES, SQLite, Mongo databases
- Internal database options MySQL, Postgres, InfluxDB, MongoDB
- Direct API access to Reuters RFA and other in-house libraries

When running data integrations, taking into account the performance effects on the legacy system is vital, especially if they are operating during business hours. As more people take advantage of the newly available functionality, it may be necessary to offload the legacy database by caching frequently used data to avoid any negative impacts.

# These are two primary data integration examples with Beacon:

- 1. Using Beacon for business-facing applications and reporting, and integrating data from the legacy system.** For example, building a real-time feed of the deals to Beacon, which enables the front office to use Beacon's risk, pricing, and visualization capabilities. An investment bank uses Beacon for research, data science, and solving various 'know your customer' questions. This includes analysis and visualization of tabular data sets over a large time span, such as account balances, and calculating growth, deposit changes, deposit to loan ratios, projections, etc. Beacon data integration enables clients to quickly deploy dashboards that help sales, financial advisors, and marketing departments offer new services, run promotions, reduce client frustrations, and improve customer retention.

Another use case is replacing portfolio analytics that are delivered to financial advisors via PDFs, with a turnaround time ranging from days to weeks. Instead, the client created a secure and scalable web portal with an interactive interface to deliver portfolio analytics to tens of thousands of financial advisors on-demand.



- 2. Continuing to use the legacy system, or spreadsheets, as the user interface, sending data to Beacon for calculations or functionality that the existing system cannot provide, and returning the results.** For example, using your existing system for deal booking, but complex pricing models and risk algorithms are run in Beacon.

Another example makes it easier to incorporate big data and machine learning into your analytics. These data science techniques are playing an increasingly important role in investment strategy, optimization, and risk management decisions, but can be difficult to add onto legacy systems. Beacon can be treated as a microservice that connects your transactions, position details, and corresponding market data to the machine learning models. The processed data is sent back to the existing system and analysts use their existing tools to review and visualize it. The elastic compute capabilities of Beacon make it easier to run these complex calculations, enabling you to run what-if queries and optimization analytics with multiple assumptions and broad scenarios to test for impacts on issues such as market, liquidity, and credit risk.

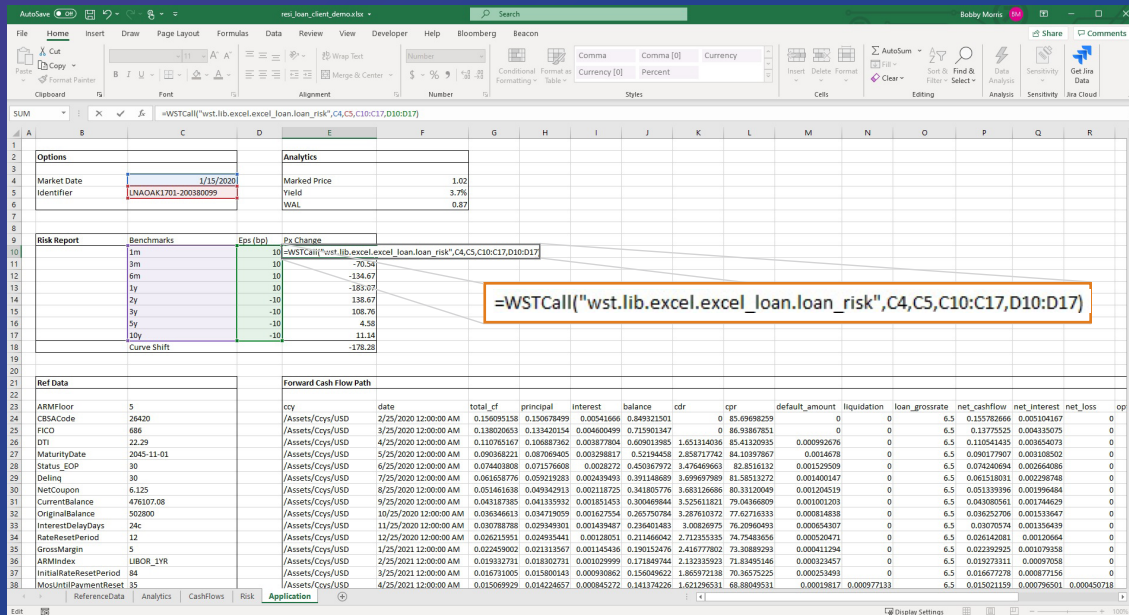


It is important to remember that integrated systems require ongoing automated reconciliation reporting, and procedures for people to manage cases where those reconciliations break. This can create a material amount of overhead, especially if there are multiple systems interacting. This ongoing overhead is a big reason why integration is typically just one step along the path to full migration.



## Excel to Beacon example

Excel spreadsheets are a useful tool for quickly creating simple models, prototypes, and running experiments, but they can struggle with large-scale projects. They also provide limited functionality for formal governance and development processes, such as version management, automated testing, and controlled deployment. Integrating Excel and Beacon addresses many of these concerns, and is a common early phase of many migration projects.



**Figure X:** Calling loan\_risk function from within Excel, passing Market Date, Loan Identifier, and an array of Benchmarks and their shifts.

Beacon provides an Excel add-in that allows users to continue using the features and user interface they are familiar with, while moving data and complex calculations to a platform that is designed for enterprise scale. Instead of storing data in tens or hundreds of separate spreadsheets, data is centralized and shared, with appropriate authorizations, improving consistency and transparency throughout the organization. Users also get the benefits of Beacon's distributed cloud computing infrastructure. 'F9' recalculations can spawn hundreds or thousands of processors, returning results for complex calculations and large scenarios in a fraction of the time it would take to run them on a desktop, completely transparent to the user.

Developers can use Python's data science capabilities for complex calculations and analytics. They get easier code sharing, formal version control, and professional testing and debug tools, improving code quality and consistency. Over time, applications can be developed directly in Beacon Platform, bringing more functionality to the organization.

# Migration: Moving everything to the new platform

Whether after a period of integration or right from the beginning, there may come a time to retire the legacy system. This means moving all data to the new platform, in a controlled and structured way. Beacon supports many different types of database, so you can choose what best matches your existing data and future requirements. Like data integration, there are some consistent steps to follow for a successful data migration:

## 1. AUDIT AND CLEANUP

Reduce errors during the migration. The older the system, the more likely that there are inconsistencies, invalid references, and forced entries because the correct category or format was not defined.



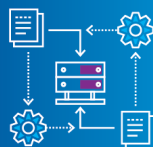
## 2. DECIDE THE SCOPE

Move everything or just certain types of data? Migrate entire trade history or only a limited date range? How will you access data that is left behind, especially if it is needed for audit or regulatory reasons?



## 3. BUILD TRANSFORMATION FUNCTIONS

Iterative process to export data from legacy and import into new. Transform uncovers data errors, outliers, and workarounds missed during step 1. Mappings to consolidate similar references, adapt instruments and trade types, or handle outdated counterparty references, among others.



## 4. EXECUTE THE MIGRATION

Run functions from step 3 and reconcile results. Compare functions or reports on both the old and new system. Mark to market reconciliations are crucial. Even 100 percent match on the trade migration can result in market value variations, due to pricing model differences or subtle discrepancies in market data.



## 5. RUN IN PARALLEL

Ensure that everything is working satisfactorily. Start with legacy system as primary and Beacon as the mirror. Then switch to Beacon as primary and legacy system as the mirror as a final check that everything is operating as expected



## 6. ARCHIVE LEGACY SYSTEM

Finally, it is time to archive or turn off the legacy system and complete the project.



## Typical Beacon implementation

A typical Beacon implementation goes through the following phases:

1. Install Beacon Platform in a client's cloud account, such as Amazon AWS or Microsoft Azure
2. Integrate to the client's internal systems interface, such as FIX, RPC, and XML
3. Integrate to desired market data to pull data and connectivity for initial runs in parallel run
4. Connect Beacon into market data vendor for prices (static, on demand, real time) and into the internal system (e.g. IBOR or CTRM) for real-time trade event feed
5. Mirror trades as they happen into Beacon's internal deal model. At this point, you probably stop adding new functionality to the legacy system.
6. While running in parallel, integrate Beacon to any other desired upstream and downstream systems
7. Scope, build, and test the data migration functionality
8. After the parallel systems have been running consistently, complete the one-time migration of historical trades
9. Run reconciliations of the full trade set each day to make sure the real-time feed did not miss anything

Successful data migration project management starts with a clearly defined scope of data and functionality of both the source and target systems. Test frequently before executing the formal migration, to ensure that you have caught and corrected all of the errors. Since the legacy system is continuing to operate during this phase, new issues and errors can be introduced at any time. Plan for potentially large amounts of data. While cloud storage and compute are highly scalable, it will take some time to move very large databases.



# Benefits and best practices

The benefits of a modern trading and risk management system are substantial. Modern code development frameworks boost your capacity for innovation, adding new functionality more easily and reducing the time to market for commercially valuable tools. Leveraging cloud-based infrastructure, you can significantly reduce the time to run reports and complex analytics while also decreasing operating costs with scalable computing resources. The wealth of available code libraries and utilities means you can buy most of what you need and increase your internal ROI by focusing on key areas of expertise and competitive advantage. Transparent and open-source software licenses give you clear visibility and ownership of everything running, and the agility to adapt as desired.

Once you have decided that a new system is worth the migration effort, success is all about effective project management. Experienced consultants bring a wealth of existing tools and techniques to the job, so that you do not have to reinvent them. Import and export connectors, data cleanup and conversion scripts, and knowledge of key metrics to track are three of the most important benefits of bringing in outside help. Like many other specialized jobs, your team may be quite capable of doing this, but is it the best use of their time and resources for your business?

## Migration project best practices can be summarized as:

- > **Clearly define the scope, source, and target, and be prepared to iterate through this as issues are uncovered**
- > **Plan and test, then test again and again, tracking percent completion across multiple criteria**
- > **Start small and scale up, building experience and reaping the benefits of the initial phases**
- > **Integrate then migrate to deliver business value quickest and pay down the integration overhead over time by retiring legacy systems**

## About Beacon Platform, Inc.

Beacon Platform moves financial markets into the future with secure and scalable quant technology, elastic cloud infrastructure, and front-office applications. The integrated development environment and web application framework make developing applications faster and easier. Born from decades of collective experience developing trading and risk management systems for Bank of America Merrill Lynch, JPMorgan Chase, and Goldman Sachs, Beacon Platform gives you the robustness and flexibility of these powerful tools. See how you can increase the return on research and development investments from Beacon's next-generation platform for enterprise-scale analytics, trading, risk management, and more at [www.beacon.io](http://www.beacon.io).

## About Essentia



Essentia is a leading energy and commodities consulting firm headquartered in Houston with offices in New York City and Mexico City.

Essentia's depth of experience in the energy industry and other commodities has made it a trusted advisory source for clients ranging from high-growth emerging companies to the Fortune 500. The Essentia consultants have worked together for more than 20 years with backgrounds in public accounting, consulting and industry. The depth of experience brings unmatched understanding of complex issues and needs which translates into refined cost-effective solutions for Essentia's customers.

[essentiaap.com](http://essentiaap.com)





# Migrating or Integrating Your Existing Data:

How Beacon Platform Brings  
New Functionality to Legacy Systems