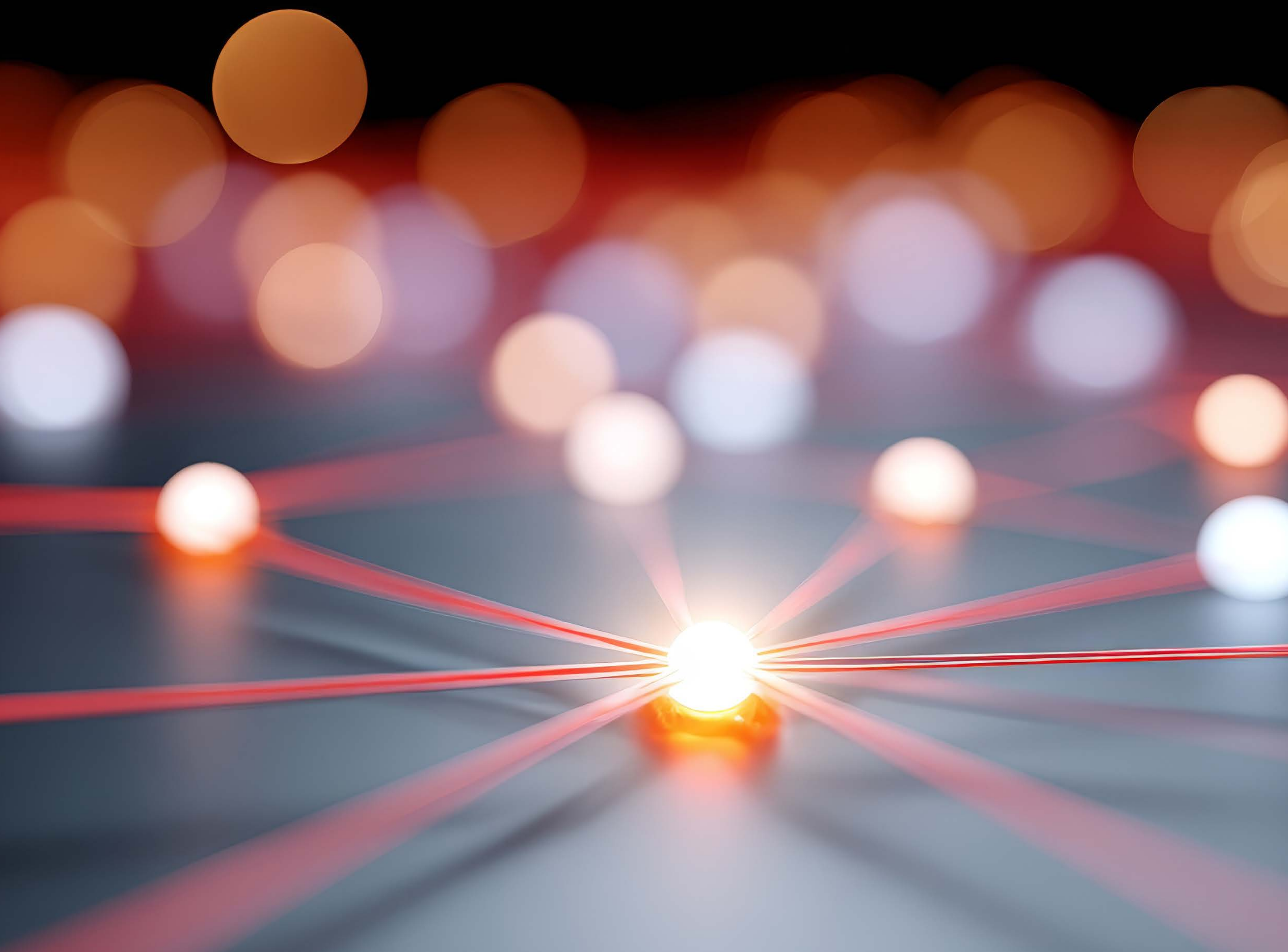
greyspark partners | valantic

MODERNIZING FINANCIAL SERVICES WITH API-ENABLED INTEGRATION

A Practical Path to Efficiency, Compliance & Futureproofing



Modernizing Financial Services with API-enabled Integration

A Practical Path to Efficiency, Compliance & Futureproofing

Most financial institutions operate using a complicated web of interlinked systems and processes. The interfaces between them are critically important to the operation of firms, especially when new systems are added into existing networks, workflows are changed, new products are added or amendments to technology are needed to comply with incoming regulation. Firms utilize a variety of systems of different vintages, including legacy 'old-world' technologies and other standalone systems that do not necessarily interact well with other systems. In this article, GreySpark Partners and valantic FSA explore how deploying interfaces that can bridge the gap between internal and external systems, 'old-world' and modern technologies, SaaS and on-premise solutions, along with many other variations, is essential to support the ever-increasing digitalization within today's financial services organizations.

Financial institutions' networks and processes very often include a multitude of workarounds, in the form of manual or semi-automated processes based around end-user-defined applications, databases and spreadsheets. Many of these workarounds are used to bridge gaps between different systems and business requirements and are often tactical amendments in response to changing workflows. The resultant data flows and dependencies can be complex, non-linear and, from a risk and compliance perspective, not well documented or audited.

Application Programming Interfaces (APIs) are increasingly used to streamline the integration of different platforms, and can be thought of as a 'bridge' between systems. However, it is common that each API connecting one system to another is unique, which means that when introducing a new system, new interfaces need to be reprogrammed with every system with which it interacts; in effect, it must be interwoven into the financial institution's enterprise tapestry.

There is now an expectation from purchasers that modern software vendors offer products with out-of-the-box, standardized and easily configurable APIs. However, even though new systems often do provide this capability, further integration work is almost always necessary to connect them with legacy, in-house or less commonly used vendor-provided systems.

Application Programming Interfaces (APIs) are modularized interfaces between systems that allow different applications and services to communicate with each other.

They implement a consistent set of routines and protocols that enable software to exchange data, translating it from one system's language into a language that can be understood by another system with which it must interact.

Examples include:

- Market data APIs that enable live prices to be streamed directly into a bank's trading platform.
- Financial Information Exchange (FIX) APIs that allow financial institutions to connect to exchanges and client order management systems to execute and manage orders and trades.
- APIs offered by financial institutions to their clients to allow the latter to query data programmatically rather than having to rely on standardised reports.

APIs can be relatively minimal, providing basic data or actions to simple requests, or they can be highly complex, connecting together multiple large systems comprised of many users and large, complex datasets, involving complex calculations and requiring strict security, high performance, data privacy and integration with other legacy systems.

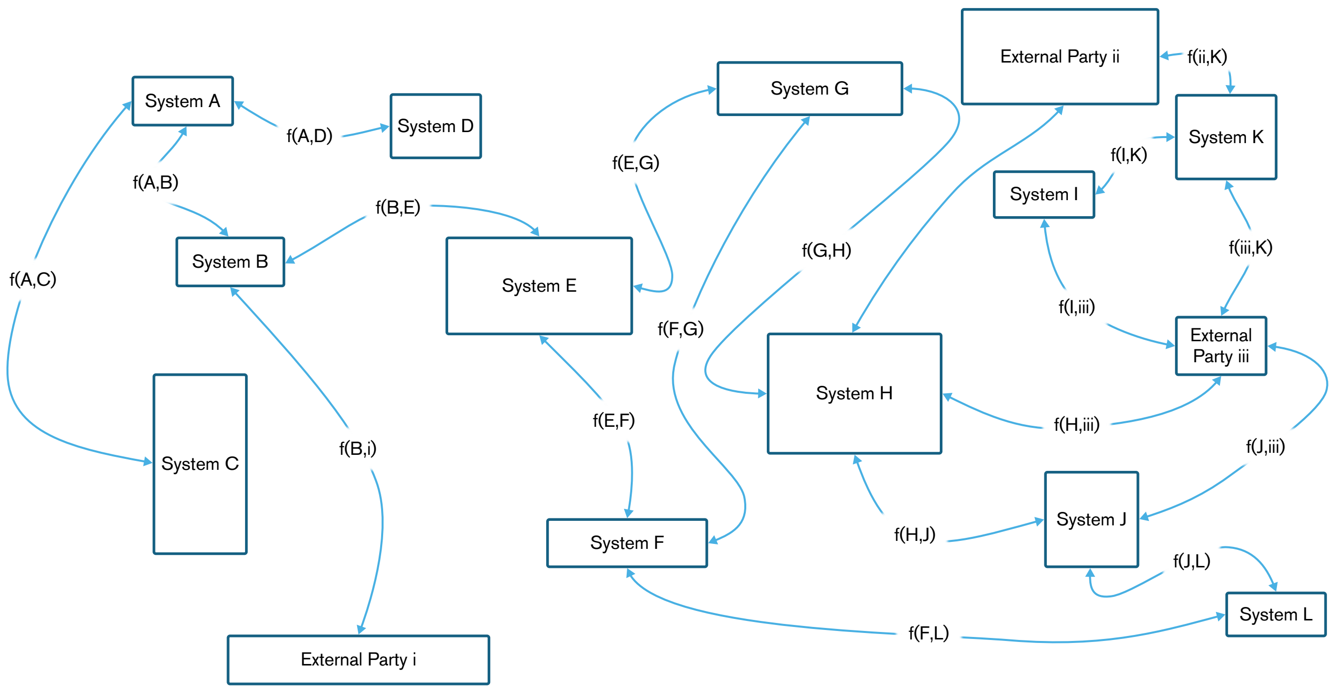
The Challenges for Financial Institutions

Fragmented and complex network architectures have come into being organically in financial institutions, sometimes over multiple decades, as systems were added over time to introduce or enhance functionality, or in response to changes in market convention or regulation. Due to the often urgent nature of the required changes, many are not made as part of a well-developed strategic plan. Most institutions have a degree of manual processing and a correspondingly low level of straight-through processing (STP). Furthermore, the interactions between different systems, many of which are executed manually or are only semi-automated, are often unaudited, representing a compliance risk.

In general, firms have tended to focus a large part of their investment on bigger and better front-office systems, as the decision-making and budget holders are primarily the front-office revenue generators and, as a result, integration issues between middle- and back-office post-trade systems have frequently been left unaddressed, and may be less than optimal as a result. Adding to the chaos is a lack of system hygiene where documentation is often missing or incomplete. Staff turnover over time can lead to the loss of knowledge about custom-built systems, and legacy software can still be in operation, despite the lack of continued support by the vendor. With the multitude of systems comes an ocean of data, which is almost certainly inconsistent in format and quality. For example, recycled or duplicated order IDs may be prevalent. Reconciliation and remediation between systems is, therefore, crucial and the data flowing between them will need normalizing at the very least. Many of the data formats and messaging protocols used are specific to financial services.

Workflows traverse many systems, and they need effective interfaces at every point of data entry and egress for every platform. Each point-to-point interface must be written accounting for the specific system at each end (see **Figure 1**).

Figure 1: An Abstract View of Point-to-Point Data Flows as a Function of Interfacing Systems
Source: GreySpark analysis



Efficiency of trade and other process flows is paramount to organizations which strive to be cost effective. However, a lack of efficient and robust interfaces between systems can cause bottlenecks that impact system response times, which can lead to outages that impact revenues as well as increase costs. Regulatory pressure is another factor to consider. Aggressive trade reporting deadlines require efficient trade flows and the upcoming T+1 settlement in the EU will demand even more.

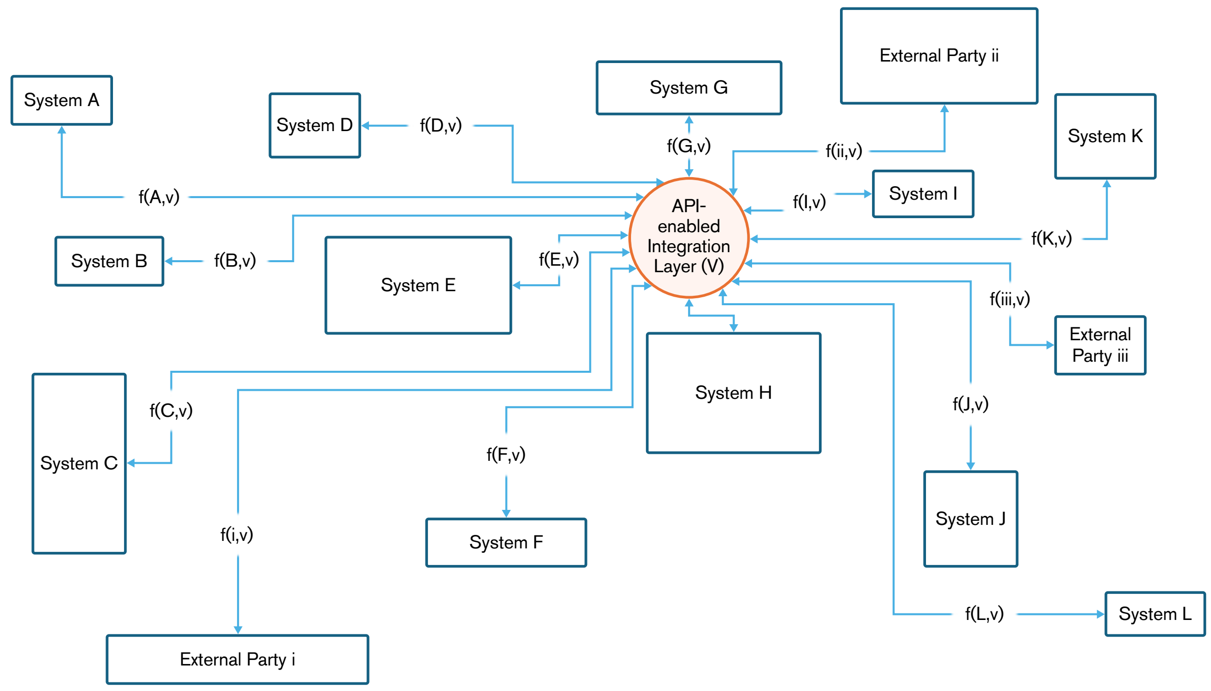
All of this points toward a compelling need to improve post-trade architecture in capital markets organizations and modernize fragmented legacy infrastructure in order to effectively increase operational efficiency.

A Practical Solution

It is not always necessary, and is rarely desirable, to undertake a wholesale 'rip-and-replace' approach towards legacy or in-house systems. Such an endeavor can be expensive, time-consuming and disruptive, and may not always be the optimal option for the business. Indeed, in many cases, the existing system might already be the best choice for the use case in question. Consequently, the answer may not be to replace systems, but to simplify and enhance their interoperability with other systems in the network, taking a structured and efficient approach. A centralized integration platform with configurable APIs (along with other integration methods) can be an effective way to achieve this. Standardized, automated and audited ingestion, transformation, enrichment, validation and routing of data via configurable APIs should be an objective as well as support for different integration methods and timeframes (real-time, near-real-time and batches) depending on use case. Additionally, the ability to support diverse data and message processing flows, including financial services industry-standard message formats such as Swift MT, ISO 20022, SEPA and FIX, plus other standards-based formats, as well as proprietary in-house data formats, will enable the firm to futureproof their network architecture.

Figure 2 shows the proposed API-enabled Integration Layer, which allows data to seamlessly flow between platforms by streamlining and automating processes. It offers mandatory features such as security and error handling, as well as being able to handle large volumes of data, requests and events, while ensuring data privacy, appropriate audit trails and archiving are maintained. It would ingest, validate, transform, normalize and enrich data before routing it to its destination system or systems through abstraction and loose coupling of applications and services.

Figure 2: APIs as a Function of Each System and the API-enabled Integration Layer
Source: GreySpark analysis



If architected appropriately, the Integration Layer could increase the robustness and transparency of processes as well as the efficiency, as it would allow the firm to retire redundant processes and workarounds, while retaining essential functionality within existing systems.

Ideally, a solution such as this would be robust yet sufficiently flexible to address whatever mixture of applications and architectures an organization has in place, including legacy technologies, data and processes. A mixture of standard, financial services-specific, configurable and bespoke APIs is likely to be required. Indeed, a platform which can easily be configured to optimize specific workflows would also help to future-proof architectures.

Case Study: Middle-office Systems and Post-trade Infrastructure Integration for a Large European Financial Services Provider

A large European financial services provider recently engaged valantic FSA to build an integration solution, using its x-gen platform, which sits between middle-office systems and post-trade infrastructure. The engagement was driven in part by the upcoming move to T+1 settlement in the EU and UK.

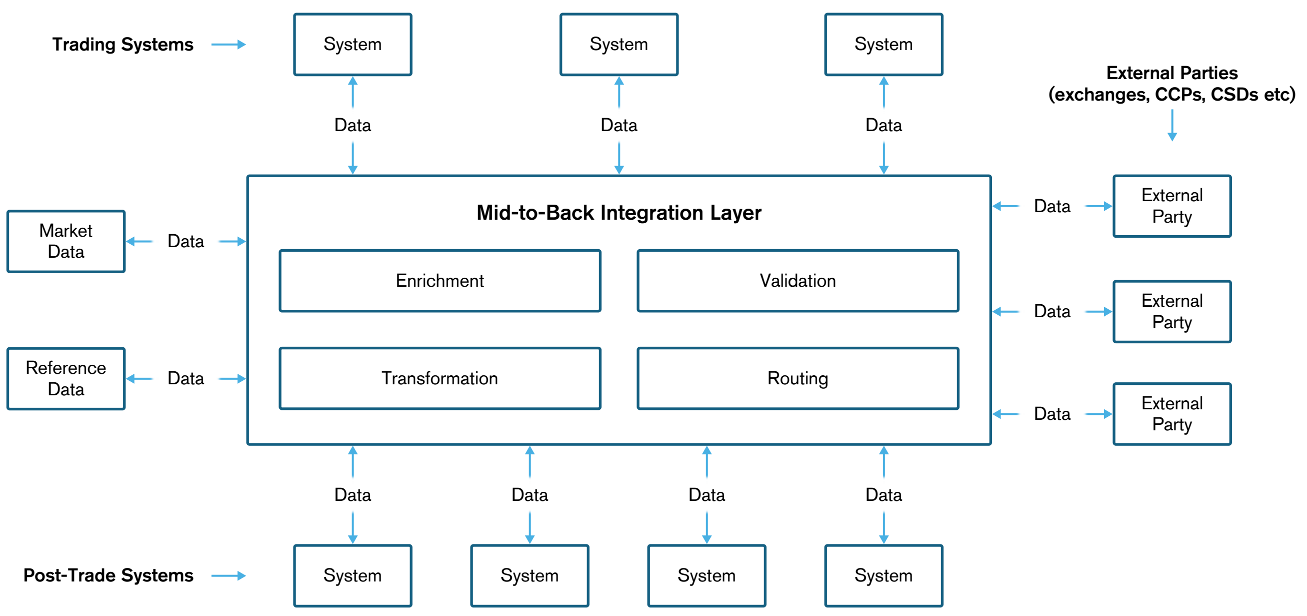
valantic FSA's x-gen (low-code/no-code) platform is a processing engine that simplifies and accelerates the creation of interfaces, the automation of data flows and workflows, and the digitalization of manual or semi-automated processes.

It supports data and message format parsing, syntactic and semantic validation, transformation and orchestration via a workflow engine that enables business process automation.

x-gen is designed to maximize automation and straight-through processing (STP) across disparate applications within the customer's systems landscape and can also be used to create complete business applications.

Out-of-the-box, x-gen supports financial services message standards such as ISO 20022 (and various standards based on ISO 20022), legacy Swift ISO 15022 (MT) and FIX formats. Proprietary or customer-specific data or message formats can also be easily supported.

Figure 3: Middle-to-Back-office Centralized Integration Layer
Source: GreySpark analysis



valantic FSA's x-gen platform, in conjunction with a streamlined dev-ops approach, were utilized to build and automate the deployment of an integration solution which sits between incumbent trading systems and post-trade infrastructure. This integration middleware layer employs a variety of configurable APIs to ingest, transform, enrich, validate and route the trade data, using several transport mechanisms including MQ-Series, REST, Kafka, SQL and WebServices, all of which are supported natively by x-gen out-of-the-box. Integration touchpoints were loosely coupled rather than hard-coded, point-to-point, to ensure maximum interoperability and abstraction between connected systems and services. This approach ensures that any future changes that may be required to a given interface have minimal impact on the remainder of the solution, since connections and associated processing flows are not hard-wired together end-to-end.

Despite a few challenges with the legacy incumbents – a lack of documentation, recycled reference data such as Order IDs and poor system hygiene among others – and some issues with latency, the solution was able to onboard the undocumented systems rapidly and successfully using its out-of-the-box, no-code mapping approach. The valantic FSA philosophy – a robust base platform, dev-ops, low-code, automated testing and deployment – allowed rapid prototyping and streamlined implementation, resulting in increased efficiency and support for the client's T+1 settlement requirements, as well allowing them to decommission a number of their now-redundant legacy systems.

The ability to integrate multiple, heterogenous applications, including business-critical legacy applications, alongside modern API-enabled platforms, with support for the required diversity of data and workflows, was critical to the successful implementation of this solution.

An API-enabled Integration Layer as a Transformation Catalyst

The learnings from the deployment in the case study described above lead to the conclusion that, as well as using the platform for this specific middle- to back-office integration solution, it could straightforwardly be configured to benefit other institutions, particularly regional banks, asset managers, smaller private banks and savings banks, many of whom have in-house post-trade operations and may be looking to address their T+1 regulatory obligations without necessarily replacing their incumbent systems. Technology and operational decision makers in these firms should consider a platform like this to address their post-trade infrastructural needs.

This integration middleware layer employs a variety of configurable APIs to ingest, transform, enrich, validate and route the trade data, using several transport mechanisms including MQ-Series, REST, Kafka, SQL and WebServices, all of which are supported natively by x-gen out-of-the-box.