Integration Strategies

HOW ITAS FITS IN ANY SOFTWARE ECO-SYSTEM

The target audience for this document are technicians and system integrators. Within it we discuss the approach we take to every integration project, contrasting traditional techniques to those required to satisfy modern-day requirements.

Moving Away from Flat Files

Traditional *flat file*-based integrations favoured for syncing data between multiple systems often require quite of bit of manual intervention. Agreements on standards can make integration projects slow and inflexible, and in turn, expensive.

The main issue is version-control. System A and System B require a set of rules in order to interpret the data and its meaning. Upgrades or changes to these rules need to be strictly coordinated otherwise mismatched data can cause issues further downstream.
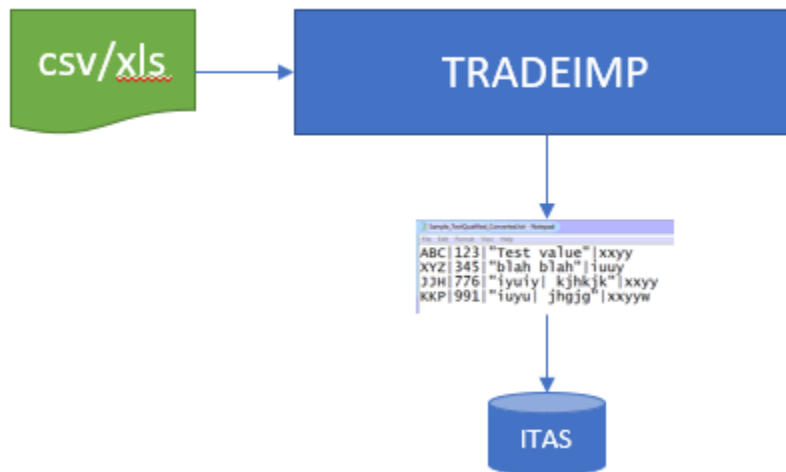

Figure 1

A typical scenario is illustrated in figure 1. A CSV contains data that ITAS requires – in this case it is contract data. Assuming ITAS recognizes the format of the flat file, the TRADEIMP process can map (transform) the data to the correct Database table and corresponding fields, and so the process will succeed. Changes applied to one side or the other (and not both) may result in a failure to map.

Maintenance of such processes is costly as it requires both sides to track changes made to either end. This results in tightly-coupled processes and deters innovation.

Avoiding Tightly-Coupled Integrations

The modern approach is to introduce middleware, which is tasked to manage the workflows between the systems requiring integration.

It understands both System A and System B but as it is responsible for the flow of data between the two Systems, it means that System A does not necessarily need to know what System B is expecting, and vice versa.
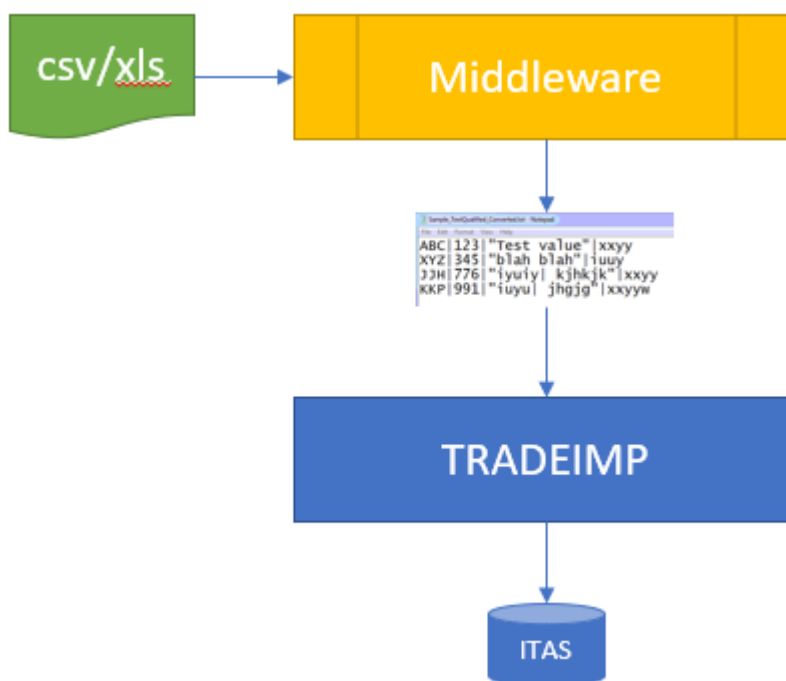

Figure 2

The above illustration now shows that the middleware has transformed the data in the CSV on behalf of TRADEIMP so import will be successful.

This is fine but there is still a requirement to be able to access the application TRADEIMP. As the transformation of the data is now being undertaken by the middleware, the application is redundant beyond saving the data in the Database. Direct Database access can be achieved through the middleware - security and infrastructure aside. However, a more flexible approach is to use an Application Programming Interface (API).

The ITAS API

An API is simply a means of providing two-way communication between two systems. In terms of the earlier example, an API endpoint would provide an open definition of the *fields* (or Properties) it accepts. This is used by the *consumer* of the API (in this case the middleware workflow) to build its request.
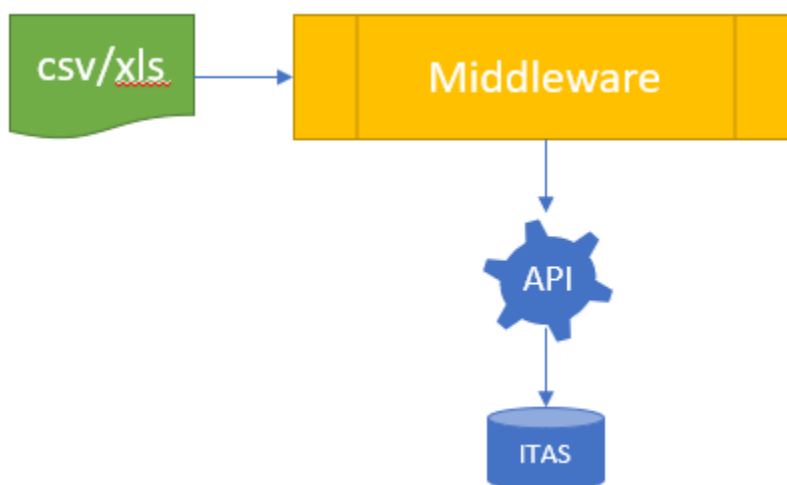

Figure 3

The above illustration reflects this change in approach.

One of the clear advantages is that communication with the API is via HTTP(S) which is the protocol used to transfer data over the web. This opens access to data and functions to a wider audience beyond your local fixed network/domain.

Security is tighter (authorization is managed through a dedicated security token), and as there is no front-end user interface (UI), the usage and rollout are invisible to the users. This allows for a wider distribution of services and adoption of other software as part of the overall architecture.

The ITAS API is scalable and extends to more complex scenarios including bespoke/dynamic data definitions through our GraphQL layer, and REST-based process subsystems for workflows around cash and inventory management.

ITAS Events and Notifications

Being able to transfer data via API is only part of any integration strategy. Triggers for making the request are essential to creating a workflow and again, this is managed through an ITAS component available through our Enterprise Services.

The Events Engine (or message queue) comprises a Database for maintaining Topics, Subscribers and Messages. The ITAS API provides endpoints to expose services that allows consumers to manage subscriber accounts and handle the flow of messages (notifications or alerts).
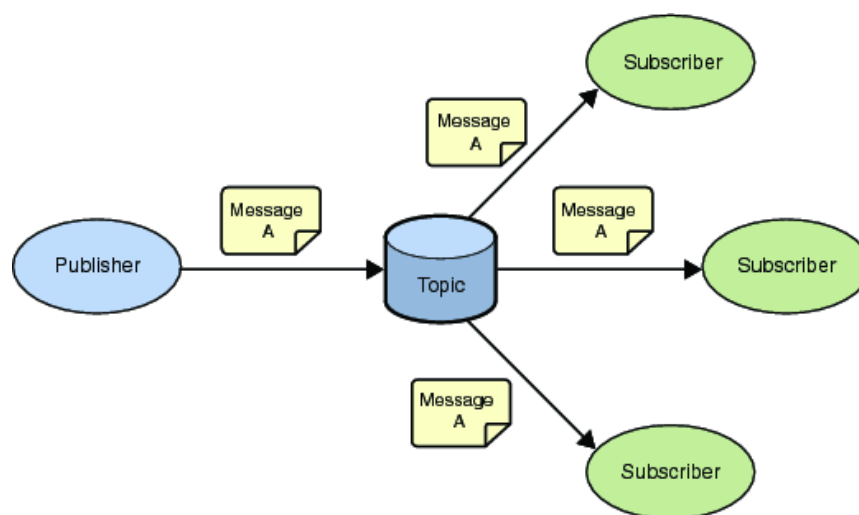

Figure 4

The above illustrates the relationship between the Publisher (the process that generates the message) and the Subscribers (those processes that are interested in the underlying Event trigger).

A Topic refers to a specific Event and Subscribers can be ITAS users (either self-subscribing through the Trader Desktop->My Alerts, or nominated through the Alerts Administrator), or external processes.

For the latter this would likely be managed in the middleware, that is, a dedicated process would hold a subscription to **every** Topic of interest and have a workflow associated with them.

Publishing Messages

In ITAS terms an Event can be configured to Publish a message to a specific Topic.

Examples of this include:

- Changes to nominated data fields, e.g. Contract Title Transfer Date
- Creation of data records, e.g. new Contract created, accounting postings
- Application-based function triggered, e.g. Contract approved
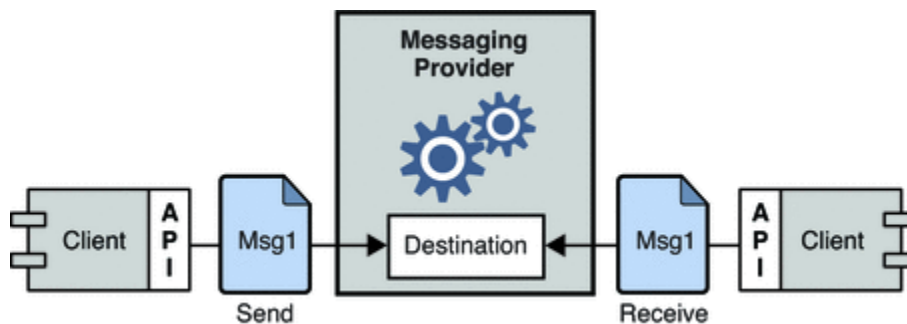- Scheduled events, e.g. data extract scripts completed

Figure 5

The above illustrates the flow of information between the Publisher and the Subscriber.

The Messaging Provider generates a 'copy' of a published message to every subscriber to the Topic.

Subscribers (clients) will pick up their messages and process them in the same sequence as they were published. Once confirmed the message has been received and processed, the subscriber will then remove the message. This will allow the next message to be picked up. This pub-sub model guarantees no messages will be lost or processed out-of-sequence.

As illustrated, all this interaction is done via API, meaning that disparate and remote systems can communicate and respond to ITAS-generated Events as easily as other local systems.

Enterprise Architecture

Integration implies workflows between multiple systems or services. ITAS Applications and Components, such as the Events Engine, can work across the Enterprise via APIs.
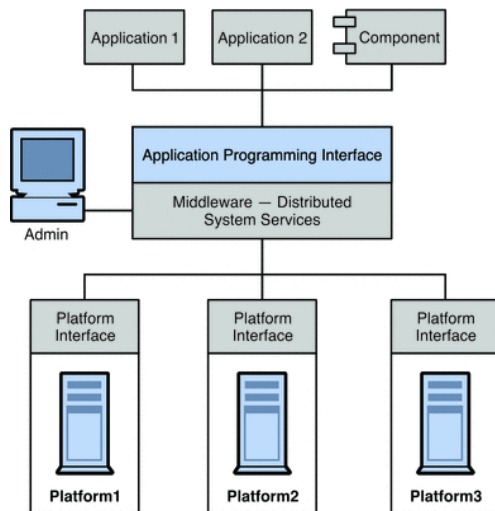

Figure 6

Services can be distributed, meaning that through APIs, Workflows can be developed across Platforms; On-premise through to a hybrid including cloud/hosted services.
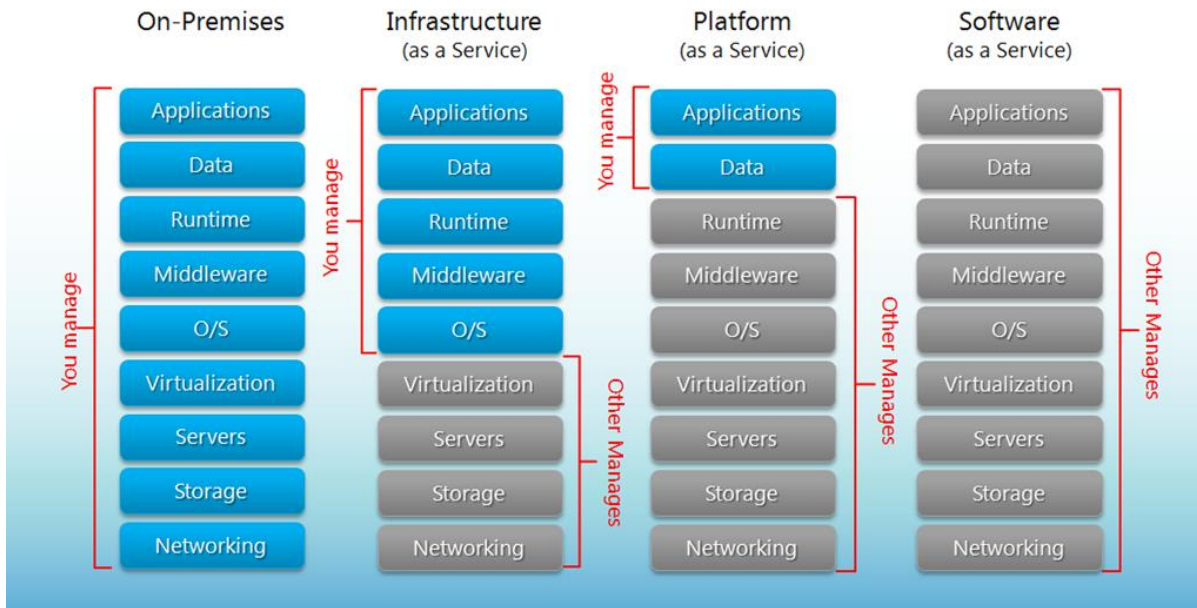

Figure 7

Integration Platform (Middleware) Options

SnapLogic (www.snaplogic.com)
Platform designed to allow customers to connect cloud-based data and applications with on-premise and cloud-based business systems. Workflows are built using "Snaps" which are re-usable pieces of logic that perform a single task. Combined they can be used to create integration tasks where the input and output can determine subsequent actions. Workflows can be batch, real-time or streaming. There are many out-of-the-box "Snaps" available for Integrations to popular systems such as SAP and Microsoft.

MuleSoft (www.mulesoft.com)
Part of the Salesforce range of products, this platform is similar in many ways to SnapLogic. Developers can build their own APIs on the (IPAAS) Anypoint Platform, as well the normal workflow and monitoring tools required to build and support Integrations. As with SnapLogic the Workflows are hosted in the cloud but connectivity to software and services on-premise is standard.

SAP PI/PO (www.sap.com)
As the integration broker of SAP NetWeaver stack, SAP PI (PO) has capabilities to integrate SAP with other legacy systems and applications. As with SnapLogic and MuleSoft, it provides a central location for an organization to build, integrate and monitor interfaces between heterogeneous systems.

ITAS WorkFlow (Data Processing Service)
Available through ITAS Services, ITAS WorkFlow provides a framework for achieving many simple integrations. Development of the Workflow scripts is undertaken by Hivedome technicians. Sitting on top of the Microsoft stack means that the regular components and workflow instances will be fully supported, with the added benefit that development of bespoke ITAS-centric activities is possible.

In short, workflows can be developed to include tasks and functions specific to ITAS, as well as those provided by third-party services via API. In many cases this approach can be taken in place of traditional application development.

Development

One of the strengths of Hivedome is the ability of our technical team to analyze and propose solutions for different environments. Often this involves integrating areas of ITAS with an external system, such as SAP or AX, that may provide specific services for part(s) of a necessary workflow.

Sometimes it is a simple data synchronization; data entered in ITAS needs to be transformed and fed into another system. Others may involve more complex scenarios, such as when it is necessary to monitor the sequence of the workflow lifecycle.

Often the issue of data transformation is key; System A (ITAS) knows the accounting generated represents a Provisional Sales Invoice, System B (SAP) may handle this type of document in a specific way. While this Invoice is being processed in System B it is important that certain actions in System A are controlled, such as deleting it.

Traditionally a recognized code would be assigned and the SAP workflow developers would lock that in. System A tells System B it is working with a PSI: System A locks the Invoice: System B processes the Invoice: Down the line the Invoice is processed and System B is informed.

This is fine; however, this approach reverts to the tight-coupling model and therefore can create issues further down the line. System A now sends a new type of document and the workflow needs to be amended in line: handling of Invoice activity in System A needs more control externally. Development can be very costly.

ITAS addresses this using a more customizable model. Configurations through EVENTSMAP and DOCSTATUS applications allow the integration developers to manage their own codes which can be mapped at a granular level according to the type of document or process notification. This mapping code can be used to identify workflows within the Integration environment. Assignment of document status codes can be achieved via API so control of specific activities can be seamlessly managed as part of any workflow. The Case Studies provide examples of how this is achieved.

Case Studies

Accounting

Scenario

The client wishes to extract certain accounting documents to feed a corporate risk system. Real-time notification is required along with access to physical and transactional data. Restricted to the posting of Provisionals, Finals and all Reversal Invoices out of ITAS. Also want daily/monthly accruals for MTM Unrealised Physicals. Important to distinguish each of these documents and to mark transactions in ITAS as processed. Middleware development is required.

Proposed Solution

- Use EVENTSMAP to assign Mapping Codes for Topic 3-8 against Event ID 1 (Provisional), 5 (Finals), 22 (Reversals) and 255 (MTM Unrealised Physicals Accruals) to generate notifications through Posting Process
- Middleware service to subscribe to these notifications (Topic **3-8**)
- The Message provides **DocumentReference** to be used as part of data interrogation via ITAS API
- The Message also provides **DocumentType** containing associated Mapping Code
- Read Endpoints available for Posted Transactions (txx) and Headers (xtx01)
- Pass data to external system via its own API (or equivalent ETL process)
- Mark document (transaction) as 'sent to accounting' via API
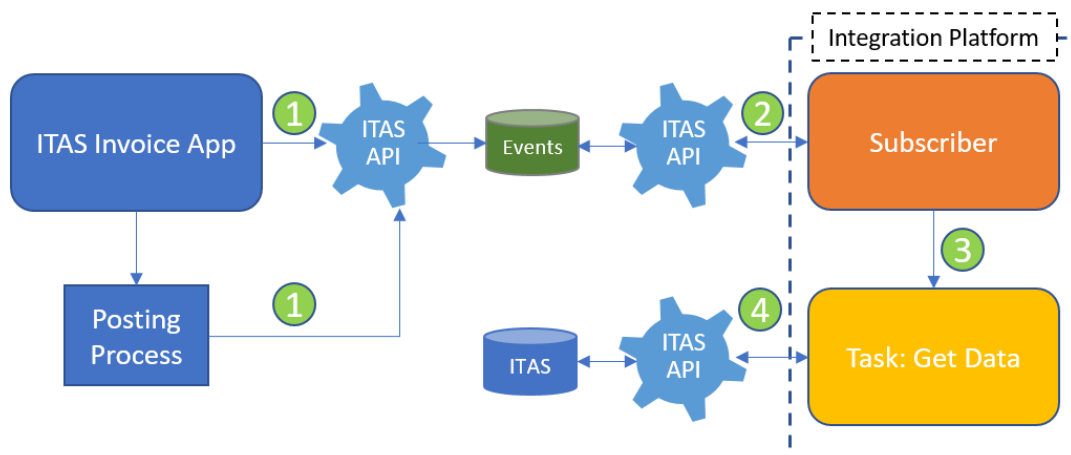
Workflow
1. Create Invoice Notification (Auth/Not Auth)
2. Check for Notification
3. Extract Invoice Document Reference
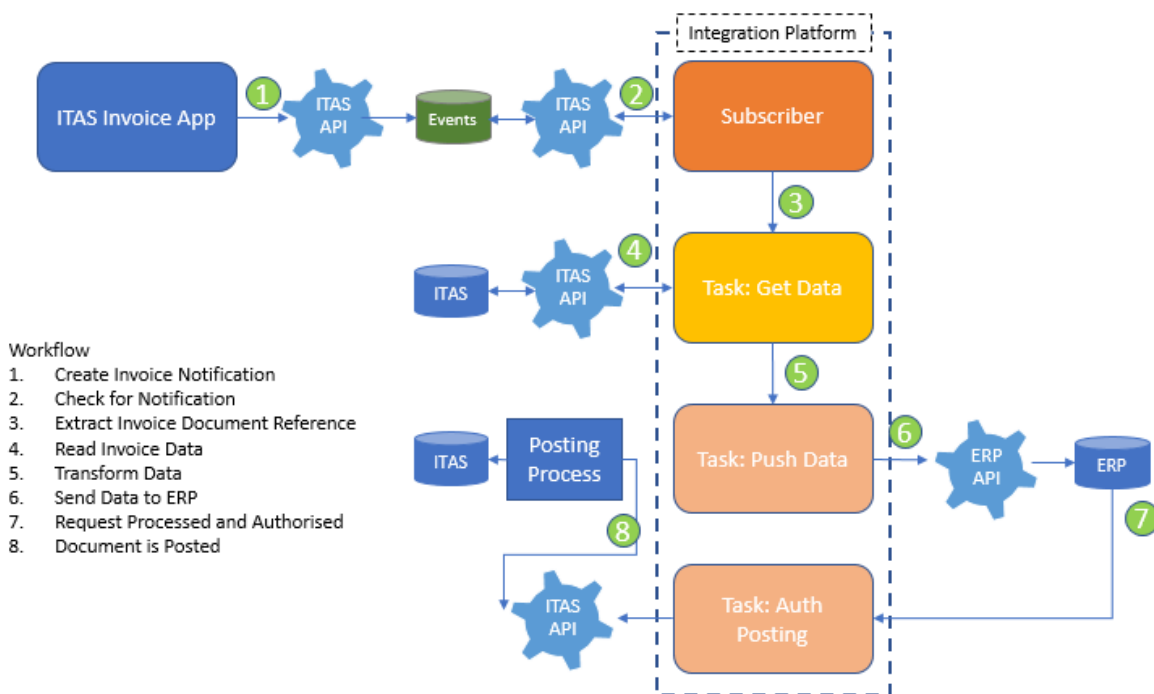4. Read Invoice Data

Finance

Scenario

Client wishes to manage Pre-Payment requests through an in-house treasury system. Accounting is fully managed through ITAS. Middleware development is required.

Proposed Solution

- Use EVENTSMAP to assign Mapping Code for Topic 3-9 against Event ID 46 (Pre-Payments) and 52 (Pre-Payments Reversals) to generate notifications
- Also use EVENTSMAP to assign Mapping Codes for Topic 3-8 against Event ID 52 (Pre-Payments Reversals) to generate notifications through Posting Process
- Middleware service to subscribe to these notifications (Topics 3-8 and 3-9)
- Message provides **DocumentReference** and **DocumentType** containing associated Mapping Code
- Topic 3-9 will identify Pre-Payments created (not authorised for posting) or deleted (pre-posting)
- Topic 3-8 will identify Pre-Payments that represent Reversals (generated when reversing posted Pre-Payment), and Pre-Payments that have subsequently been authorised for posting
- Read Endpoints available for Posted/Unposted Transactions and Headers
- Pass data to external system via its own API (or equivalent ETL process)
- Put Endpoint available to Authorise Pre Payment for posting in ITAS

Workflow
1. Create Invoice Notification
2. Check for Notification
3. Extract Invoice Document Reference
4. Read Invoice Data
5. Transform Data
6. Send Data to ERP
7. Request Processed and Authorised
8. Document is Posted

11

Summary

ITAS Services offer an array of components to support the development of Integrations in any corporate environment, across any landscape.

Through the combination of Events Management and an extensive range of REST and GraphQL-based APIs, we are confident that we have the tools to enable the most complex of system-to-system workflows to be developed and supported.
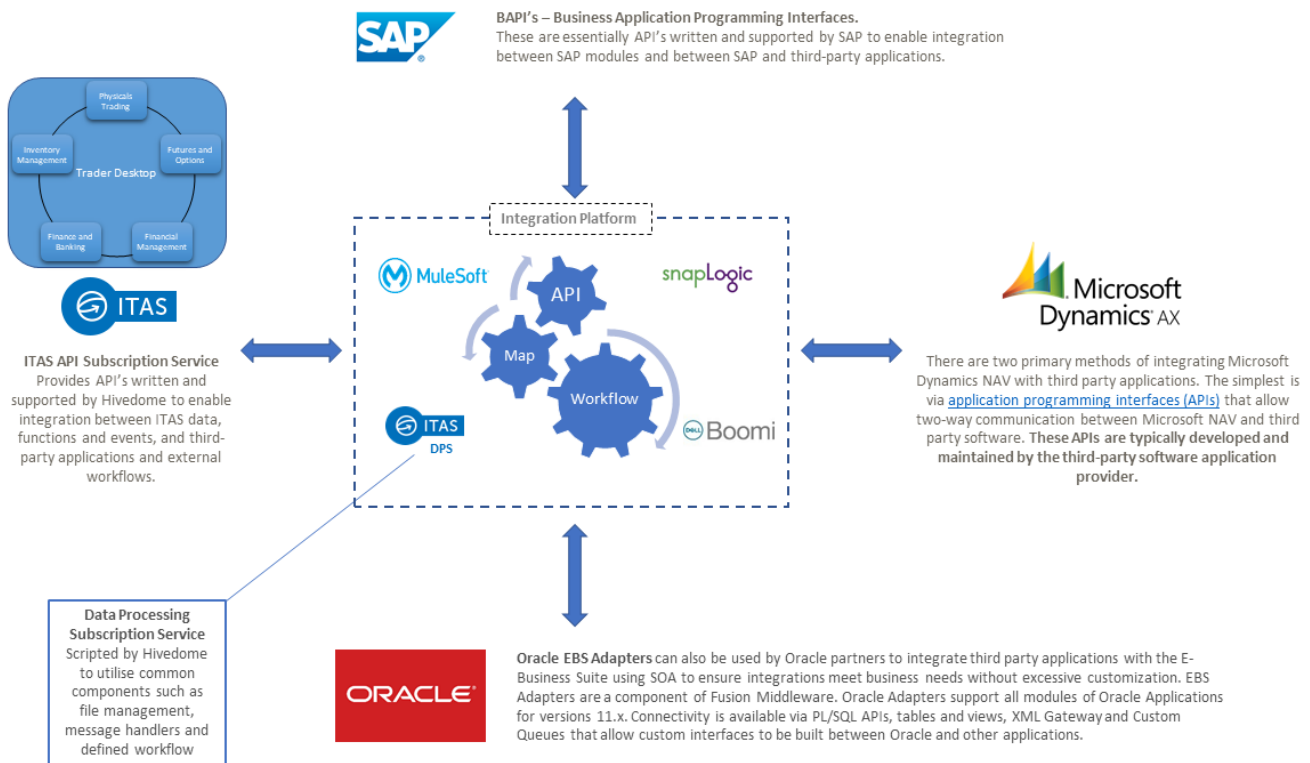
Please contact Hivedome to discuss your specific requirements:

Finance House
20-21 Aviation Way
Southend-on-Sea
Essex SS2 6UN
United Kingdom

+44(0)1702 433030
salesemea@hivedome.net