Connecting the dots of a software ecosystem.

PIDX Conference - April 2021
About me

• CEO of Payload Technologies
• 20+ Years of Software Development Experience
  • O&G, Medical, Forestry, and Real Estate industries
• Started from Developer to Architect to Executive
• Passion for Software Development, Gaming and Muscle Cars (and Star Wars).
Introduction

A Software Ecosystem is a collection of software projects, which are deployed, integrated and evolve in a connected software infrastructure.

• Application integration is the process of enabling independently designed applications to work together.

• Integrations are key consideration in how effectively an organization can use its applications, processes and data.

• Few, if any, software applications and data sources will live in isolation for a typical business.
  • Even if they are only connected via manual process today.

• According to a recent study, 99% of SMBs that use digital tools say these tools help them improve business performance and remain competitive.
What does this mean?

Integrations are key to a modern business's success.
Single vendor vs. best of breed.

- Meant to be an all-in-one, goal is to eliminate the need for integrations.
- A unified solution that houses all data and functionality into one place.
- Services and customization are part of the overall solution ecosystem.
Single vendor vs. Best of breed.

- Independent solutions with depth of functionality in a specific area.
- Presents depth of functionality and deeper capabilities in their domain.
- Enterprise level solutions will be designed for integration at code and data levels.
Realities of single vendor and best of breed solutions.

Single Vendor

- Ties you to a single platform, vendor and price increases.
- Always involves integration which often requires more specialized and expensive resources.
- Does not evolve as quickly in every area, tends to focus more on the platform than the solution.
- Customization can be required to get depth of solution for your business needs.

Best of Breed

- Rate of change can be a positive but also creates more effort to keep integrated, easily.
- Can be less customizable.
- Infrastructure requirements and integrations can be more complex and fragile.
Key takeaway

"In either case, integration is inevitable as your software ecosystem evolves."
1. Data

- Done through ETL, Data Dumps or Data Connectors.
- Cloud Data warehousing has simplified this through pre-built connectors and shared DMZ’s.
- Enables data aggregates to form a Source of Truth useful in AI/ML and Data Analytics.

2. Process

- Can be automatic or manual integrations between different workflows in an organization.
- Consider opportunities for optimization through synergies and/or redundant aspects of two separate workflows.
- Process Maker, Azure Logic Apps or AWS Simple Workflow (SWF) many more options.
  - Evaluate each option to see if they are at the right level for your needs (high/low level).
3. Application

• If supported, can enable two applications to impact each other's state directly.

• Additional logic may be required to handle unavailable services (dead letter queue).
  • This also enables advanced options/alternate processing for these cases.

• Typically achieved at the WebAPI level through REST, GraphQL or SOAP.
  • REST has 70% adoption, easier, flexible and more performant than SOAP.
  • SOAP provides more security and better suited for distributed environments.
Process and application integrations.

Each of these integration types typically involve some aspect of the other.

- Data integrations rely on applications connecting to one another, even just a db connection can be considered an application integration (e.g., JDBC driver).
- Process integrations rely on applications and data to store workflow state and pass system commands.
- Application integrations need to send or receive data to for a typical use case.
Many different approaches to data integrations, each has its use but for system-to-system integration less is more, seek to avoid data at rest when unnecessary and reduce layers to keep things simple.

**File drops**

With in-app export functions (automated or scripted).
- Good for data extraction to allow users flexibility with spread sheets and manual imports.
- Bad for system-to-system integrations - less secure (data at rest), more infrastructure (SFTP).
- Recommend auditing for use of this approach to removing where possible.
Data integrations - part one.

Most Process and Application integrations involve data exchange at some level.

Data Loads
Through custom scripts and SQL triggers.

- Highly fragile, create a complex integration environment and costly to maintain.
- Recommend replacing with ETL or Data Connector.

WebAPI's used for Data Extraction>Loading

- Can be used in Data Virtualization implementations (e.g., Denodo).
- API changes impact data structure, system availability considerations and costly to maintain.
- Recommend replacing with ETL or Data Connector if possible.
ETL/ELT or Extract Transform Load

- Many tools to streamline much of the complexity to move data.
- Consider costs involved, many ELT SaaS providers are charging "by the row".
- Well known technique that has been improved over time to be relatively low cost.

Data integrations - part two.

Most Process and Application integrations involve data exchange at some level.
Data integrations - part two.

Most Process and Application integrations involve data exchange at some level.

Data Connectors

- Can be implemented over application or data level integrations, consider for your needs.
- Even if built over web services, should provide architectural abstraction for data ingestion.
  - For instance, data model changes can be captured with default or placeholder values.
- Many 3rd Party sources to assist with connecting to supported systems (Hubspot, Salesforce, etc.).
Data integrations - part two.

Most Process and Application integrations involve data exchange at some level.

Shared Dataspace

• A DMZ between two different data ecosystems.
• Provider can create a view layer to enable direct integration of BI tools and applications.
• Very simple to use for the consumer, often just a database connection.
• Often allows storage to be handled by provider for a "JIT" approach to queries.
  • e.g., You may not want to store millions of geo-coordinates but may need to build ML models from time to time.
As shown previously, there are many different mechanisms that can be used to integrate systems in your software ecosystem.

With choices, comes complexity in two forms:
- How do we enable these integrations on a Technology level?
- How do we understand the impact of these integrations on a Business level?

PIDX provides clear direction for both questions, which is key when integrations can happen in so many forms and across so many applications.

When considering your options, understand that standards do not need to be rigid and constrain your own solution(s).
- Sometimes we use standards to bridge the gap between systems only.
"Integrations should be viewed as an opportunity to synergize your existing and new investments. Integrations are a foundation to enable aggregate solutions of the future so your company can transcend a myopic and localized understanding of its potential."
Questions?

www.payload.com