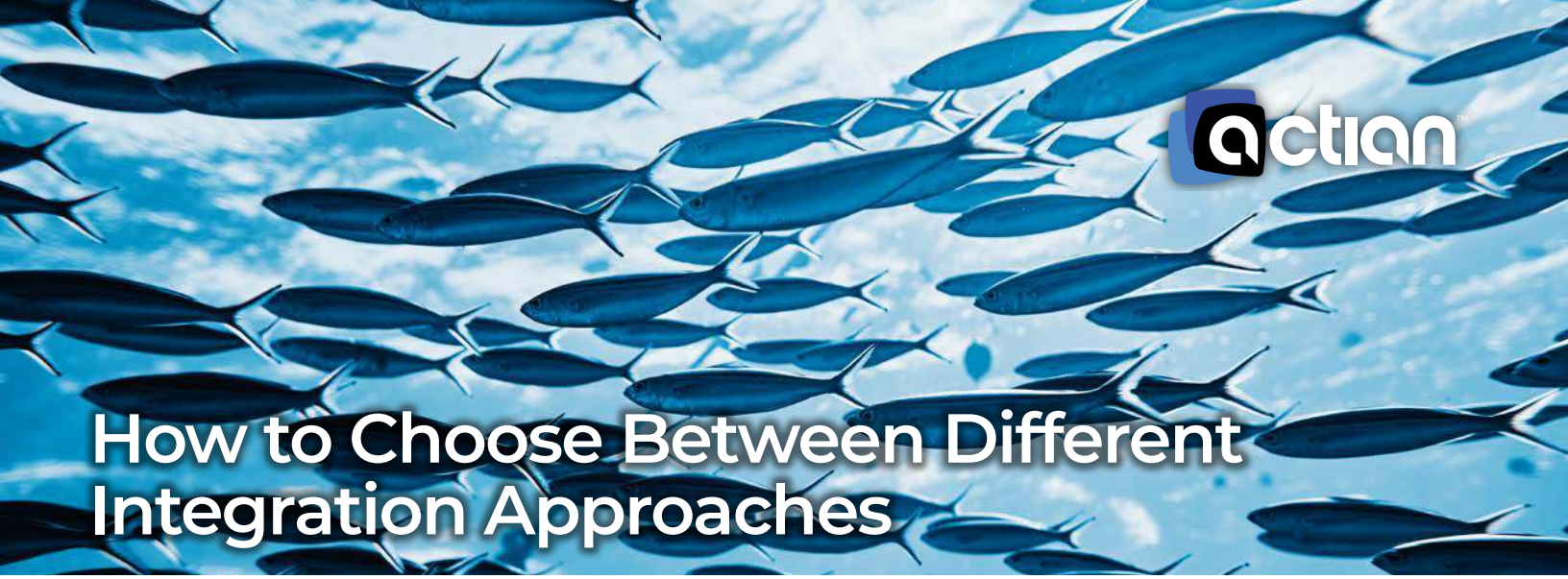


How to Choose Between Different Integration Approaches

A Guide for Application Integration leaders and Enterprise Architects

With independent teams delivering integration capabilities, the risk of tool proliferation and duplication of efforts is a reality for many organizations.

Application leaders responsible for integration need to understand where and where not to use point-to-point integration.

Integration leaders must understand the different types of integration-delivery approaches and ensure the right technologies and methods are used for each implementation. The diversity of integration requirements and the demand for timely integration cannot be fulfilled with a one-size-fits-all approach.

- Many technologies have overlapping capabilities, which can lead to duplication of integration efforts, increased support costs and a reduction in the effectiveness of a center of excellence (COE).
- Many personas are doing integration in companies, including the integration developer (application developer, specialist integration developer, Line of Business (LOB) integration developer and citizen integrator).
- Different roles and projects require different technologies to be successful.

Application Integration leaders and Enterprise Architects are challenged to bring order to the chaos to ensure business value is recognized for their companies. They must understand the types of integration delivery (point-to-point integration, embedded, managed platform, enterprise application integration, custom code, IpaaS, iSaaS, IoT integration, etc.), and decide which are appropriate for their organizations and in what situations. They must ensure project teams are using the right technology for each type of integration delivery by matching the toolset with the developer type and delivery style. They must also define and apply the correct level of governance for each type of integration delivery.

Need More information?

Visit www.actian.com/dataconnect for product, service, and solution information, papers, success stories and much more.

Point-to-Point Integration

Point-to-point integration is a tightly coupled integration between two or more endpoints. Point-to-point integration is the simplest form of integration, but it can introduce technical debt with a complex spaghetti architecture that is hard to manage and change.

Point-to-point integrations are the simplest form of integration, but that simplicity comes at a price. Because they are tightly coupled to the applications for which they are created, they cannot be reused, and each individual integration increases the complexity of your overall integration portfolio in a linear (or worse) fashion. In addition, point-to-point integrations can be challenging to monitor, and it is easy for these one-off solutions to become lost.

This approach is ***the right*** choice for a given integration when:

- **Proof of concept/prototypes** – Assess the strengths and limitations of various integration designs, even when purchasing integration tooling (since the tools often obscure the underlying design considerations.
- **Time to market is key,** and the project is self-contained, with integration being used between internal application components.
- **The production lifetime of the project is short-lived** and does not warrant long-term investment of expensive integration technology. The integrations are unlikely to be usable in other projects.
- **Application centric integration design** – Use point-to-point tools accompanying SaaS applications when those applications are at the center of the integration design.
- **Active development** – Use coded point-to-point integration when application developers can deliver integration as part of their application. Developers can quickly do the work and where the whole solution is easily redeployed when changes are required.

Extract, Transform and Load (ETL)

The ETL approach automates the process of moving data between systems and keeping data synchronized with minimal human intervention. As the name implies, Extract, Transform and Load (ETL) software manages three aspects of the process:

- **1. Extraction** involves getting a copy of data from a source, which could be an application, database or text file.
- **2. Transformation** translates the source data to match the format of the target system. This includes changing data types, combining or splitting fields and applying more complex formulas.
- **3. Loading** completes the process by putting the transformed data into a target system.

This approach is the ***right choice*** for a given integration when:

- **Requires skilled staff to handle it** – Use this approach when you have business analysts who can handle it.
- **Frequency** – ETL moves data in batches, often on an hourly or daily basis, so it works well for moving data behind a firewall inside an organization when the data is not time-sensitive. Many companies use ETL software to load data from different systems into a data warehouse for reporting and data analytics. ETL is not suitable for near real-time processes.
- For cloud deployments, it requires an on-premises server to run the integration software.
- **Time to market** – In most cases, integration, design and testing require less than a month.
- **Have high-data volume** – Handles very high-data volumes into the millions of records.

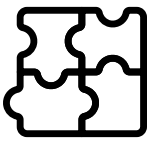
Embedded Integration

An embedded integration solution is an advanced cloud-integration mechanism that addresses the increasing challenges enterprises encounter when trying to integrate with a diverse portfolio of customer and partners. A modern embedded integration solution features the B2B tools and capabilities that can connect customer systems, whether it's on-premise or in the cloud.

Integration embedded into a SaaS Solution

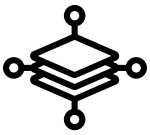
Many SaaS application providers try to remove the integration barriers to adopting their SaaS applications by embedding integration capabilities within the applications. This often takes the form of simple connectors and adapters, but, in a few cases, can actually be fully functional embedded integration technology, such as ESB, ETL and iPaaS. The limitation here is the integration technology is closely tied to the serving application, and cannot be used independently, increasing dependency on the parent application.

3rd Party Connectors



An alternative approach to using the embedded integration capabilities of the SaaS applications is the increasing range of specialist after-market adapters and connectors. In this scenario, a third party implements the adapter or connector and it is installed and configured at one end of the connection (typically as an extension of the SaaS application). Although this approach allows end-user organizations to integrate applications rapidly with little development effort, it has its limitations where applications have been heavily customized.

Embedded Integration Platform



Additionally, embedded integration creates a secure data-transfer gateway that is built directly into the (SaaS) tool's architecture. Once fully integrated within an enterprise's environment, it can begin to realize a more seamless customer interaction with faster and more reliable data exchanges. An embedded B2B integration engine helps to reinforce the integration workflows behind the scenes – the onboarding of partners, customer communications, etc. – so your valuable IT resources can focus on core business initiatives that create value.

Benefits of Embedded Integration platform:

- **All the integration connectors and protocols you need** – An embedded integration platform comes pre-loaded with application, SaaS and B2B connectors that enable an enterprise to integrate easily with any partner or customer requirement.
- **Eliminate the need to build customer code** – Without an embedded integration solution, managing all these systems to work together requires a tremendous amount of customized code and one-off integrations for each new customer. The consequence is the monumental task of maintaining each of those integrations.
- **Enable DevOps** – Embedded-integration technology allows for a quick spin-up, spin-down architecture that includes flexible licensing, database independence and complete support for inflexible infrastructure patterns.
- **Highly scalable architecture** – An embedded integration platform can support the most demanding levels of throughput and allows you to expand your customer base as well as connect to your customers as fast as you (and they) require.

This approach is the ***right choice*** for a given integration when:

- Time to market is important.
- Application developers will be delivering integration, although there may be some LOB integrators involved for larger projects, especially when using federated traditional integration technology.
- There are plans to reuse the integrations for other closely related projects, either for geographic, divisional or departmental reasons.
- There are no plans to expose shared services to other projects outside the application domain based on embedded integration.
- There is a strong desire not to depend on central shared services to create the integrations.

IPaaS-Integration Platform-as-a-service

Integration platform-as-a-service (iPaaS) is another hub-and-spoke approach, but it provides the capabilities of the other approaches as a multi-tenant cloud service. Companies can purchase the service directly, and other SaaS companies can embed the iPaaS in their products instead of building their own point-to-point integrations. Integration is no exception to the trend of moving on-premises applications to the cloud in pursuit of new functionality and cost savings.

The iPaaS approach differs from the other approaches to integration in several important ways:

- **Rapid time to design, deploy, monitor and repair integrations**
  An iPaaS allows you to build and deploy even complex integration projects during a much shorter time than the other approaches. Implement a hybrid iPaaS solution for most data-integration requirements, because this enables rapid deployment of complex, multipoint integrations. Scenarios requiring strict data governance or high-volume local processing still require an on-premises solution.
- **Web-based design and monitoring**
  Like other cloud applications, an iPaaS has a Web-based interface that you can access wherever you are. You can design and maintain connections and integrations, monitor results and resolve errors.
- **Broad array of application and technology connectors**
  An iPaaS will include connectors to the main SaaS applications in multiple categories, such as CRM, ERP and marketing automation. It will also provide technology connectors for common databases, text files and protocols, such as OData and ODBC.
- **Design once and deploy anywhere**
  Hybrid Integrate between cloud and on-premise systems. In addition to cloud-to-cloud integration, an iPaaS manages integration with on-premises systems, usually by installing agent software behind the corporate firewall.
- **Distributed processing for scalability**
  The cloud portion of an iPaaS is a shared, multi-tenant environment, but it distributes integration workloads to agent resources that are dedicated to a single tenant, so that one tenant's integration jobs do not affect the other tenants' jobs. The system scales by adding agent resources, whether in the cloud or on premises.
- **Cost**
  Like most SaaS applications, an iPaaS is a subscription service with pricing based on the capabilities a customer needs. Service tiers are based on number of connections, connectors available and access to advanced features. There may be add-on services for development and test environments. Pricing is usually straightforward and available on the provider's Website.

- **Published method for building new connectors**
  Many iPaaS platforms have a software developer's kit that enables customers and partners to build their own connectors. Some also provide a marketplace to make certified third-party connectors available to other users of the platform.

- **API for programmatic control**
  Other SaaS companies can embed the iPaaS in their products, allowing them to offer integration to other SaaS products. An application-programming interface (API) allows another SaaS application to control the iPaaS remotely without the user interface.

The iPaaS vendor keeps connectors up to date as SaaS companies update their products, testing new versions and making changes as necessary. This allows you to escape the endless regression-test cycle associated with point-to-point integration. SaaS companies benefit doubly when using iPaaS instead of point-to-point integration or an ESB. When you build a connector for your own application, you can use it for the embedded integration you offer, but you can also publish it on the iPaaS vendor's marketplace, making it available to any of your customers who buy an integration service directly from the iPaaS vendor. This also makes it easier for end users to migrate their integrations from the embedded solution to a direct relationship with the iPaaS vendor.

When do you need to consider IPaaS:

1. Deploy an iPaaS solution when starting from "greenfield" with no existing data-integration solution. iPaaS allows future-proofing and flexibility by leveraging cloud-based, vendor-hosted products.

1. Evaluate iPaaS when you are struggling to integrate new types of cloud apps or data stores, connected ecosystems, platforms and 'things." iPaaS caters to these emerging needs for which project requirements are frequently changing and thus faster time to value is crucial.

2. Use iPaaS as an extension of an organization's data integration infrastructure if on-premises tooling exists, because iPaaS can augment and evolve your integration portfolio toward a hybrid integration platform (HIP).

This approach is the ***right choice*** for a given integration when:

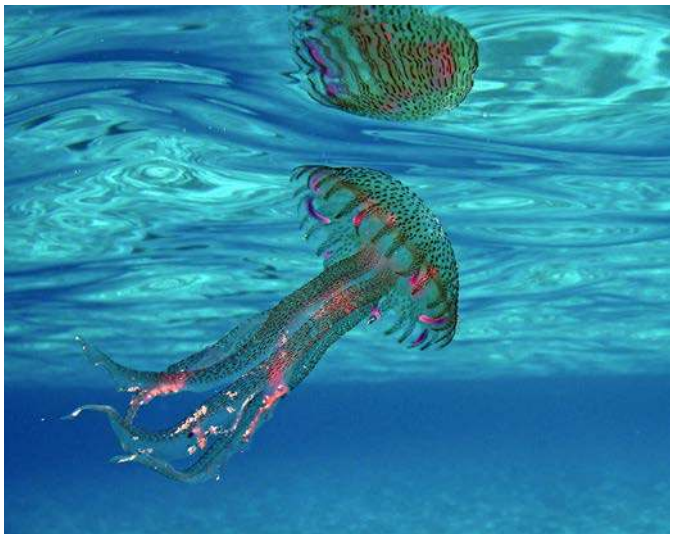
- You have a large portfolio of integrations to manage.
- You need to transform or modify the data being passed between the applications to make the integration work.
- You need to reuse the exposed data or functionality for other integrations.
- You want to enable ad-hoc integrators to assist in implementing integrations.
- You want to outsource the operational aspects of your integration middleware.
- Your application portfolio is pivoting to the cloud.

This approach is a ***poor choice*** when:

- You have multiple custom-made or niche applications.
- Most or all of the applications you want to connect are in your data center.
- You want a high degree of control of the inner workings of your integrations.

iSaaS – Integration Software as a Service

Integration software-as-a-service (iSaaS) packages are a nascent market of cloud-based integration tools that provides either prepackaged or easily configurable integration flows aimed at helping non-IT business users, and even consumers, to address simple application and data-integration issues. iSaaS offerings are wholly focused on citizen integrators. They provide only very basic core integration capabilities, often do not support on-premises endpoints and, in most instances, do not enable on-premises or hybrid deployment models. iSaaS offerings are typically:



Cloud-hosted – Easing adoption by removing the need for IT to manage the application.

Easy to use – Enabling users to configure, run and manage quickly the life cycle of simple integration flows (called "cloudstreams" or "recipes") without support from IT professionals.

Empower citizen integrators – Who have no professional IT experience, but are increasingly technology-savvy. These users may be either business users or consumers. iSaaS enables any business user to become a "citizen integrator," customizing available cloudstreams or applying integration recipes and prebuilt flows.

Configure prepackaged/from-the-box integrations – Create new integration flows by mapping predefined endpoints (SaaS applications, cloud services, abstracted APIs, social networks, mobile apps and "things") via simple "if/then"- type modeling.

iSaaS tools cover a wide range of cloud-based and mobile endpoints: CRM and ERP applications, spreadsheets, data visualization, digital marketing, social media, project planning, file sharing and others. These tools very rarely support on-premises business applications. Generally, users apply their personal credentials (rather than a central "superuser" account provided by IT) when authenticating to the endpoints. This tends to be a limiting factor to the scope of impact of the integration flows. Often, the result of the integration is the automation of steps that an individual had previously done manually using a "swivel-chair" approach (copying and pasting and/or manually reentering data).

This approach is the ***right choice*** for a given integration when:

Empower Citizen Integrators – Embrace iSaaS to enable business users to support their integration (or "automation") ambitions and goals in a do-it-yourself fashion, thus becoming citizen integrators. Develop a strategic plan to empower citizen integrators as part of an enterprise-wide "pervasive integration" competency that spreads the integration workload across many personas.

Reduce the time to value of new digital initiatives, particularly those that incorporate new mobile apps and SaaS applications. Rapidly react to "digital business moments," such as customer complaints on social media.

Increase integration competency within the organization. This will allow citizen integrators (as well as application developers and SaaS administrators) to self-service simple tasks and should allow specialists to focus on higher-value and more-complex integration work. Application leaders should select your iSaaS with the guidance of your business leaders and influential business users, but also factor your technical requirements into the selection process. This way it will be possible for central IT to monitor, manage, administer and govern your citizen integrators.

Embed or incorporate an iSaaS offering into its hybrid integration platform to complement its established, integration-specialist-focused technology, whether it be from a pure-play, enterprise platform or ecosystem-focused iSaaS vendor.

Frame your iSaaS approach within your hybrid integration platform (HIP) strategy to deliver additional value (for example, integration with on-premises applications and data sources), and to implement the appropriate levels of governance and control.

Managed Platform Integration

Projects requiring high-productivity integration tools typically use managed platform integrations. Almost always, there is either a cloud element or a B2B element to the integration requirement, whether it is a LOB integrator connecting salesforce.com into SAP applications, a citizen integrator extracting employee information from Workday into a spreadsheet or an application developer sharing services with a third party. Here the driver is to use cloud-optimized tools to deliver the integration as quickly as possible while avoiding having to build the complex integration infrastructure required for such an endeavor.

The most common technology here is iPaaS and iSaaS, although cloud-based API management platforms and cloud-based messaging platforms also have a part. Integration outsourcing – aka integration brokerage – is another option for companies that simply wish to outsource integration-project implementation and management.

Note: For some advanced organizations, this also can be delivered by allowing self-service capabilities on existing internal integration platforms (essentially a private iPaaS), but this is far from the norm.

This approach is the ***right choice*** for a given integration when:

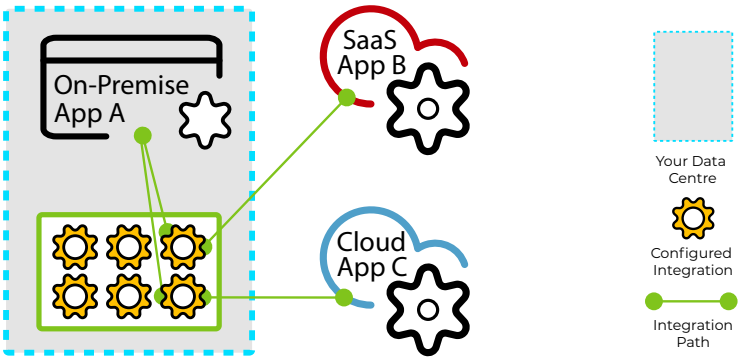
- Citizen integrators want access to data and services to improve business productivity and insight.
- LOB integrators want to integrate SaaS applications to other SaaS applications and/or on-premises applications.
- There are no plans to create enterprise-shared services from the integration.
- There is a strong desire not to depend on central shared services to create the integrations.
- The low upfront costs of iPaaS and iSaaS concerning both budget and project time would accelerate project delivery.

Enterprise Application Integration/ Enterprise Service Bus

Enterprise Application Integration (EAI) uses a hub-and-spoke approach in place of many point-to-point connections. The integration application serves as the hub with spokes connecting to the other applications. EAI allows enterprise IT organizations to automate business processes through near real-time communication between applications.

Early EAI products acted as a central broker, accepting messages from one application, performing integration tasks and sending messages to another application. Today, the Enterprise Service Bus (ESB) is the most common solution for EAI. Centralized teams typically deliver an enterprise-shared-services integration approach that expose shared services to allow access to central IT-managed systems, such as systems-of-record and systems-of-differentiation. Here strong governance is required to protect the systems of record.

Integration Platform Software / ESB



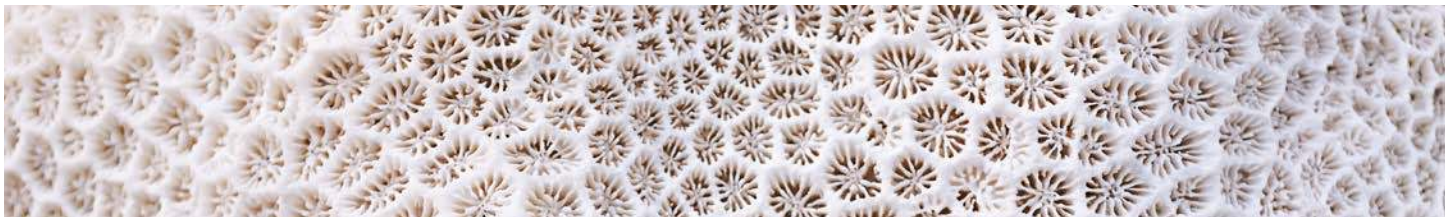
The IT staff within an organization builds connections between its applications and the ESB, and staff members code the transformation and other integration tasks that are required. This approach works for on-premises applications in large companies. They purchase an ESB software license and training, and they employ IT developers to write and test integration code.

This approach is the ***right choice*** for a given integration when:

- You are embarking on a strategic business transformation or application-modernization initiative that will leverage investment of existing applications using service-oriented architecture (SOA) and APIs.
- You have a large portfolio of integrations to manage.
- You need to transform or modify the data being passed between the applications to make the integration work.
- You need to reuse the data or functionality of the source system for other integrations.
- You want centralized control of access to all their services via application services governance.
- You want to expose service facades that enable access to systems-of-record and systems-of-differentiation via API management or ESB technology.
- You want to create composite applications or expose composite services that aggregate data from different systems of record, API management or ESB technology.
- You want to leverage existing investment in integration technology.

This approach is a ***poor choice*** when:

- You want to enable ad-hoc integrators to assist in implementing integrations.
- Your application portfolio is pivoting to the cloud.
- Your center of gravity for integrations is cloud applications. The low upfront costs of iPaaS and iSaaS concerning both budget and project time would accelerate project delivery.



Integration Brokerage – Outsourced Integration

Integration brokerage (IB) is a form of IT outsourcing that allows you to delegate responsibility for integration-project fulfillment and continuous maintenance to an external service provider. As such, IB is the most radical alternative, because you do not implement your own HIP (although the service provider probably uses one to deliver its IB service). For historical reasons, IB providers specialize in supporting large-scale B2B integration projects, but many of them have extended their IB services to cover a broader spectrum of use cases.

Special considerations for choosing an integration broker (IB)

- Clearly establish whether you need a stand-alone or embedded IB offering, and review IB-offerings suites for additional functionality. Both types of offerings are viable, but are quite different in solution scope.
- Ask for IB project references that represent accelerated onboarding for both the IB customer and its post-project, go-live trading partners. Review them closely to ensure onboarding timelines and processes satisfy your organization's needs.
- Scrutinize the post-project support process and level of participation required/desired by your organization. Do not assume that your IB provider will act as first-level support for issue resolution with your trading partner.
- Seek IB pricing simplicity, transparency and predictability.
- Choose IB providers based on their capabilities in traditional B2B e-commerce, e-invoicing, internal application integration and cloud-service integration, and for their ability to serve your future needs, such as API management and the IoT.

This approach is the ***right choice*** for a given integration when:

- **Aging infrastructure:** Many companies have investments in legacy integration products that are at the end of their product life, but they lack the budget, resources or desire to invest in potentially costly capital expense.
- **IT modernization and drive innovation:** Many companies need new capabilities, often to execute digital strategies, such as analytics, APIs, e-invoicing, collaboration and compliance. These capabilities may not be easily addressed with existing integration solutions. Many companies with limited resources must allocate resources to initiatives for modernizing and maturing integration.
- **Complexity:** Many companies (rightfully) consider integration to be challenging and time-consuming and, therefore, seek advice on the viability and options for outsourcing. The desire to consolidate B2B (and possibly A2A), integrate the infrastructure, manage provisioning of all application and trading-partner connections via one centralized provisioning tool and leverage one mapping development tool for all projects.
- **Time to deployment:** Even if the budget and skills are available to deploy integration technology, many companies find that service providers are more capable of fulfilling time-to-deployment needs for increased scale, auditability, visibility, analytics and e-invoicing.
- **IT infrastructure consolidation** – Macro-business trends to consolidate the IT infrastructure and macro-business trends to do more IT outsourcing.
- **Offsetting limited IT skills with outsourcing** – IB is an IT-managed service offering that delivers people, methodologies and technologies – the latter enabled by cloud-based integration, such as integration platform-as-a-service (iPaaS) – for B2B e-commerce and cloud services integration projects.
- **Need Cloud-based B2B infrastructure and management** – Traditionally, the provider hosted B2B integration software hosted on-premises or via application hosting. Most IB is now technically enabled by iPaaS, a more highly evolved form of integration-as-a-service. Among other things, such cloud-based B2B infrastructure provides IaaS, governance, managed file transfer (MFT) and community management capabilities.

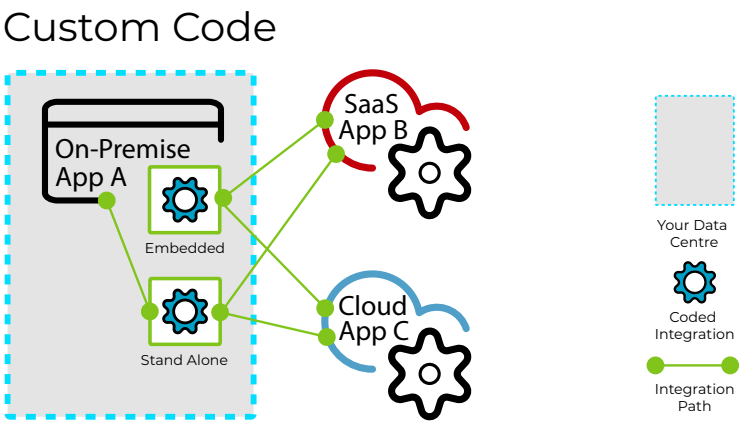
This approach is a ***poor choice*** when:

- Turning the responsibility of your B2B partner relationships over to an external party, which either removes or limits direct interactions with your partner's IT department, thus leading to a potential reduction of inside knowledge of your partner's IT priorities and challenges.
- Concerns about the evolving IB service-provider ecosystem and whether IT providers remain committed to long-term investment in integration brokerage.
- Increased viability of innovative integration alternatives, such as B2B-capable integration platform-as-a-service (iPaaS), where the platform, but not necessarily the integration project, is outsourced.

Custom Code Integration

Development teams always have the option to create a direct integration between systems. This may take the form of a stand-alone application, or it may be an extension of one of the applications being integrated. It may be written entirely from scratch or leverage open-source frameworks, such as Apache Camel or Spring, but in all cases, this type of integration couples two or more systems directly to each other. The flexibility of custom code provides the capability to integrate between otherwise incompatible applications, but these integrations are hard to reuse. Custom-code integrations are the most difficult to deliver.

This approach is most commonly used for data warehouse ingestion and data preparation for analytics engines. Because of the complex nature of the data manipulations, integration specialists usually deliver this approach using a comprehensive range of technologies and techniques.



This approach is the ***right choice*** for a given integration when:

- You have the skill set in-house to design, develop, deploy, operate and maintain the integration.
- You are implementing a small number of integrations.
- You need tight control of the integration's behavior.
- The integration is between applications that are stable and well-supported, with low likelihood of change.

This approach is a ***poor choice*** when:

- You need to reuse the data or functionality of the source system for other integrations.
- Either or both of the applications are expected to change frequently or be replaced in the near future.

This guide illustrates that an "one-size-fits-all" approach does not address all the different integration requirements today's organizations face, so you must weigh the various alternatives carefully before choosing an integration approach for your project. iPaaS platforms are gaining momentum in the industry due to cloud adoption, flexibility and affordability. The other approaches are well-established and may be suitable.

About Actian

Actian DataConnect is a hybrid data-integration platform that can make the process of data migration much easier and lower the risk of business disrupting connectivity issues. With DataConnect, instead of managing a bunch of point-to-point interactions between applications, connections to source systems are managed through the data connect platform. Implementing a hybrid integration platform, such as Actian DataConnect, can help you accelerate migration timelines, improve effectiveness and improve the likelihood of a clean and successful data migration project. To learn more, visit: www.actian.com/dataconnect.

www.actian.com | **Toll Free** +1 888 446 4737 | **Tel** +1 650 587 5500
2300 Geng Rd., Suite 150, Palo Alto, CA 94303

© 2019 Actian Corporation. Actian is a trademark of Actian Corporation and its subsidiaries. All other trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. (DS23-1016)