White Paper

The Urgent Need for Hybrid Integration

Sponsored by: IBM

Stewart Bond Maureen Fleming

July 2017

EXECUTIVE SUMMARY

Over the past few years, business leaders have been the primary drivers of technology change, including making decisions to adopt new applications in the cloud, mandate a cloud-first strategy, offer new capabilities with an API-first strategy, and provide new applications to end users on mobile first.

There are significant benefits to these cloud decisions because they decrease time to value, lower costs, and make it easier for organizations to experiment and innovate. But there are consequences as well, chiefly in the complexity of learning how to integrate applications and exchange data across a decentralized architecture that is largely driven by autonomous development decisions.

The following key triggers illustrate how organizations are responding to changing integration requirements:

- The need to integrate cloud and on-premises applications
- Mobile enablement of applications and digital services
- Proliferation of REpresentational State Transfer (REST) APIs and the need to manage their life cycles
- Supporting analytics-driven applications
- End-user self-service integration

Rather than treating each need for new types of integration individually, hybrid integration is a strategy and framework for the adoption of:

- Technologies, tools, and services that work holistically to meet enterprise integration requirements, regardless of endpoint, endpoint location, architecture, and integration developer role
- Best practices for integration governance that factor in both the drive to innovate by fostering development autonomy and the mandate to identify services that need to be shared more broadly and conditions that require industrialization

This IDC White Paper answers the following questions about the need for hybrid integration:

- How are changes in business strategy and technology adoption requiring changes in how organizations approach integration?
- What are the major events that trigger integration adaptation?

- How are the roles involved in hybrid integration changing and impacting integration development processes, integration governance, and integration tool selection?

SITUATION OVERVIEW

Business requirements ultimately drive technology adoption, and over the past few years, business leaders have been the primary drivers of technology change and resulting adoption decisions. Key examples include:

- Declaring cloud-first technology adoption
- Adding smartphones and tablets to the mix of endpoints requiring custom applications
- Participating in technology-based ecosystems
- Building new businesses by harnessing data from sensors
- Using data provided by external partners, customers, and employees
- Expecting the same level of ease of use in the workplace as workers experience in the consumer world

There are significant benefits to these decisions, including shortening time to value, lowering costs, and making it easier for organizations to experiment and innovate.

Business-led technology adoption decisions also have consequences, particularly for the IT organization. IT is responsible for accommodating each change while maintaining existing systems required to run the business. Today, most new technology adoption involves a shift from centralized control and deployment of applications to decentralized deployments with more autonomy.

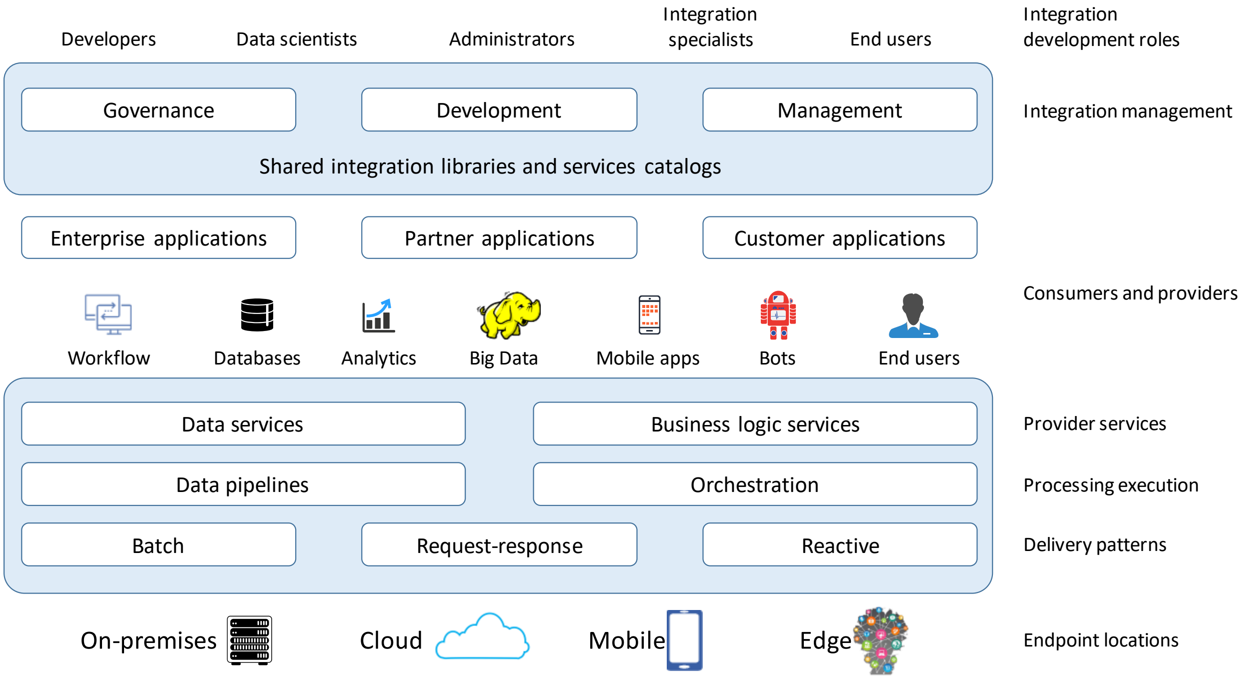
Even while there is large-scale decentralization, IT must support interoperability between existing systems and new applications, regardless of where these applications are located. The net effect is greater overall complexity.

Figure 1 illustrates the complexity of decentralization that many organizations encounter with new applications and new sources of data and adoption of newer types of technology deployed across diverse locations. Each of these deployments becomes a provider or consumer endpoint. This is far different from the integration requirements of 2000, when custom and packaged applications and databases were deployed in an enterprise datacenter and integration was primarily local. Today, there is a greater diversity of technologies consuming services, service providers, and endpoint locations, and integration is situational.

In 2017, partner and customer organizations work together using the same applications or exchange data and services between applications. Different data-centric and business logic–centric integration tools are used in tandem to support interoperability. As the exchange of data and services are decentralized, there is also a shift from batch to event-driven, or reactive, architectures for delivery of data or prompts to trigger a service. Figure 1 also illustrates the growing decentralization — or autonomy — of workers involved with integration and automation development.

FIGURE 1

Enterprise Integration Must Support Decentralized Assets



Source: IDC, 2017

Integration and access standards and the use of interface documentation to instruct developers about how to access data and business logic programmatically have been around for years and are supported by commercial integration software. Continuing advances in technology since 2000 lowered the cost of integration and the flexibility of building applications. For example:

- The push toward service-oriented architectures (SOAs) shifted to a new set of interface architectures providing standardized access to data and business logic offered as services.
- Cloud architecture extended the benefits of SOA, simplifying the interfaces with the use of REST APIs.
- Advances in security made it easier to provide secure access to cloud services without the use of Virtual Private Networks (VPNs) or gateway software.

Each advancement made it easier for developers to support business demands for faster development cycles, new types of applications, and new locations to deploy applications, resulting in a largely decentralized application architecture spanning locations and devices. As functional and nonfunctional requirements became more challenging, innovations in integration and access technologies made it possible to meet those challenges.

Five Key Triggers That Force Changes to Existing Integration Capabilities

With each major change, the cost, performance, and reliability of existing integration methods were no longer as effective. Enterprises could no longer sustain existing integration technology without extending and accommodating new integration solutions supporting new use cases. We've identified five key triggers for the adoption of new integration technology to support new requirements. These triggers are:

- The need to integrate cloud and on-premises applications
- Mobile enablement of applications and digital services
- Proliferation of REST APIs and the need to manage their life cycles
- Supporting analytics-driven applications
- End-user self-service integration

The sections that follow provide use cases about each trigger and how enterprises have adapted to the change.

Trigger 1: Cloud and On-Premises Integration

An organization's shift to public cloud is a significant and disruptive event that signals a need to refactor integration. The following key cloud adoption use cases force changes to an integration portfolio:

- When there is a decision to decommission an older legacy application, replacing it with a major SaaS application, a ripple effect occurs impacting all applications that interact with the legacy application.
- Cloud-first mandates make on-premises application adoption an exception rather than the rule. For developers and integration specialists, the cloud-first policy is an indication that there will be a need to incorporate cloud into core integration capabilities.
- Slow and steady shifts to the cloud for new application adoption create integration manageability problems that become too difficult and expensive without a major change.

The integration architecture teams tasked with determining how to refactor integration to support cloud often do more than simply add cloud integration into their mix of options. They often rework processes, APIs, and types of integration used to manage changes caused by cloud adoption.

For example, business functions that use a core enterprise application, such as financial, HR, sales, or contact center, have an ecosystem of dozens and even hundreds of supporting applications. The main application itself may sit in an enterprise datacenter or may be hosted by a third party. The cluster of applications communicates by web services connected over a LAN inside the datacenter or by VPN when the application is hosted. An integration bus often mediates the interactions. It is also common to synchronize data between applications using Extract Transform Load (ETL) or File Transfer Protocol (FTP) software.

These methods are no longer effective once these applications move to a SaaS model. When the business adopts a major SaaS application, the following integration considerations have to be resolved:

- Using ETL or FTP to synchronize application data is not secure enough between a datacenter and the cloud application, and cloud-optimized data integration software is required.
- Communication shifts from a high-speed LAN to slower broadband connections, creating higher integration latency. This may require a rework of services interfaces to narrow their scope while making them more lightweight to make them faster.
- There is the potential of lower reliability between the datacenter and the SaaS application, which means there may be a need for reliable messaging and improved error handling.
- Web services interacting with the legacy application may need to be extended to include REST APIs to support formats required by SaaS applications.
- The integration bus will not be capable of mediating cloud-originating web services requests without use of gateway software or some type of trusted agent.
- There may be a need to integrate assets in the cloud, which means the integration capabilities must be extended to support every workload in every cloud, impacting the new major application and relevant business processes.
- There may be a decision to colocate supporting applications by hosting them in the same cloud as the SaaS application. This may remove latency and reliability concerns, but there is still a need for integration, which means there is a need to adopt new integration software.
- Data associated with the cloud application may need to be replicated to a cloud or on-premises data repository for reporting and analytics, which may require cloud-resident data integration and movement technologies.

With the decision to deploy applications in the cloud, enterprises need to add new integration capabilities while maintaining their existing integration tools. Typically, they purchase integration Platform as a Service (iPaaS) software, which provides a cloud-optimized way to handle integration between clouds and between clouds and datacenters. However, because most enterprises also must maintain capabilities for integrating assets running in their datacenters, all services and data movement of all types regardless of location will need to be managed and maintained, driving hybrid into the heart of operations. This doesn't mean the integration tools must be the same, but they must be able to work together as one hybrid solution. For this reason, integration teams should plan to support existing datacenter-based integration plus iPaaS in combination.

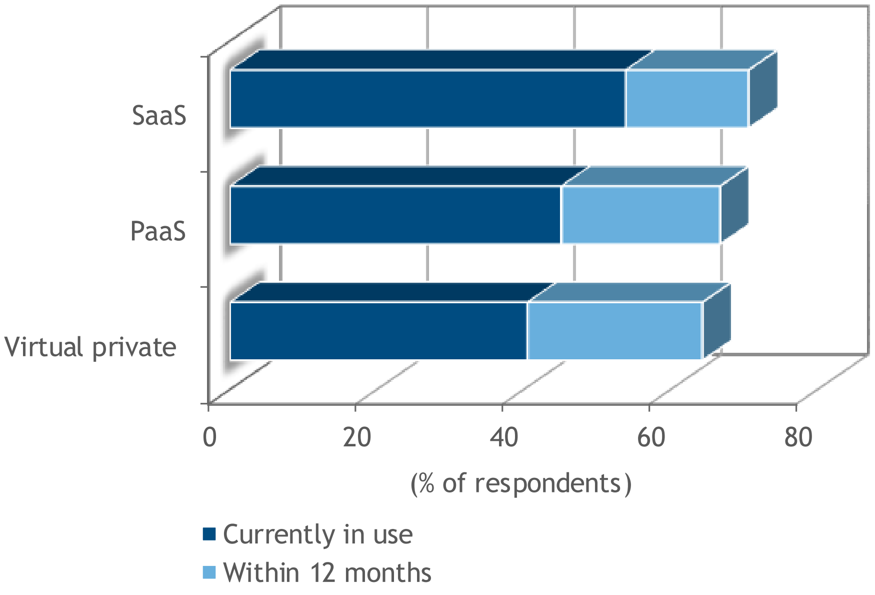
Planning for Mixed Use of On-Premises and Public Clouds

According to IDC's worldwide 2017 *CloudView Survey,* nearly 54% of respondents have adopted SaaS for at least one application, and an additional 17% of respondents are planning to adopt SaaS within 12 months (see Figure 2). Figure 2 also indicates that by 2018, more than 60% of respondents will have adopted not only PaaS but also virtual private cloud – private instance in a public cloud.

FIGURE 2

Cloud Adoption Plans

Q. Describe your organization's current or near-term plans for adoption.



n = 6,068

Source: IDC's *CloudView Survey*, 2017

Despite the widespread adoption of public cloud, the *CloudView Survey* indicates that enterprises also plan to customize applications and processes inside their datacenters. The proportion of respondents who don't have any plans to run custom business processes in the cloud is equal to the proportion of respondents who have plans for cloud-based process execution.

Trigger 2: Mobile Enablement of Applications and Digital Services

Just as many organizations have a cloud-first strategy for systems of record, many organizations also have a mobile-first strategy in place for systems of engagement with customers, employees, and partners. Applications offered on mobile phones and tablets have a separation between the mobile app and back-end business logic and data services. Mobile devices access back-end services via APIs, regardless of where the service is located: in on-premises datacenters or in the cloud, or both depending on the application and use case.

The following key business-driven use cases create the need to reexamine integration methods and new capabilities required for mobile enablement:

- A mobile-first strategy for any new system of engagement
- Automating business processes that are mobile by nature, such as quality assurance in manufacturing, site inspection in housing, and retail and insurance appraisals
- A need to provide mobile device access to legacy applications in support of an increasingly mobile workforce, such as selling and sales support and field service
- Providing secure and authorized third-party mobile application access to business logic or data to be used by customers, partners, or contractors

An architecture team faced with the task of determining how to expose business logic and data via secure APIs to mobile applications will need to do more than the equivalent of connecting another web app to an integration bus. Mobile applications are designed to be run on mobile devices, not fixed in a specific location, and are generally uncontrolled and connected via unreliable networks with varying bandwidth. When the business enables mobility for any existing or new applications:

- A security-first design is required for developers creating APIs between back-end services and mobile devices as well as a build and test methodology to ensure corporate assets are being protected from unwanted access and use via mobile channels.
- Cloud is the preferred location for back-end business logic and data services being accessed by mobile devices because it is "closer" to mobile devices, available whenever and wherever internet connectivity exists. Placing this logic in the cloud may require another layer of connectivity and replication to/from on-premises if a legacy application is part of the use case.
- Production APIs must be managed to ensure the back-end services are performing adequately, and as systems are updated or the mobile app is modified, resulting changes are made to the APIs.
- Session and state management is included in the API functionality, mobile applications, and integration technology solutions.

Enabling a functional mobile application requires new integration software and will result in a need for mobile device management software. API management software is recommended to meet these requirements. iPaaS is recommended when APIs are cloud resident, accessing cloud and/or on-premises back-end services.

Trigger 3: Proliferation of REST APIs That Need to Be Managed

REST APIs are the access standard for new application projects and enablement of digital business. Different business and developer trends caused a proliferation of REST APIs, requiring improved management and governance.

Developer trends include:

- Adoption of microservices architecture as a core evolution in DevOps is aimed at speeding up the development of applications and other types of technology enablement, allowing developers to build applications through the assembly of independently deployable, small modular services available as a service through a lightweight interface.
- Robotic process automation (RPA) focuses on the development of micro activities and workflows that automate tasks formerly performed manually by end users. Each of these sets of activities is packaged as a service with an API that, when called or triggered, executes the automation. This may include automating a series of steps that cross a set of applications, performing calculations in a spreadsheet, or capturing unstructured text or paper and structuring it into a data model, for example.
- Custom shared services, which are business logic or data that is customized by an enterprise, are available over a network and accessible through a standard API. Shared services are implemented on-premises or in the PaaS tier of a cloud, depending on the purpose of the shared service.

Externally focused digital business use cases include:

- Executing a platform and developer network strategy involves businesses offering fee-based access to technology-based services via an API. Over the past year, there has been a proliferation of developer network offerings as companies offer platforms for use by developers to cost effectively extend the developer's existing applications, or the developer can add value to the platform by building on top of it. Today, many of the brand names in business — such as Visa, Mastercard, Walmart, GM, Ford, and Barclays — are adopting an API-first platform or developer strategy.
- Externalizing APIs for partner and customer use is also common to make it easier to automate doing business with each other.

As APIs became associated with innovation, under the term *API economy,* many integration architects with mature SOA integration infrastructure in place were confused with the benefits of REST APIs. DevOps teams also tended to try to manage them on their own or tie them into existing integration solutions.

As the adoption and demand for REST APIs proliferated, there was a significant recognition across enterprises that REST APIs and API management have an important role in integration. The use cases vary widely but are linked by the assumption that people other than the developer will want to use the API in an automated fashion. If there is anything wrong with the API, the applications using the API will not function. That means anyone who needs the API will:

- Need to look at documentation to understand the details of the functionality offered by the API
- Have a dependency on the performance and availability of the API
- Be concerned with life-cycle changes and version control
- Want to ensure APIs are secure

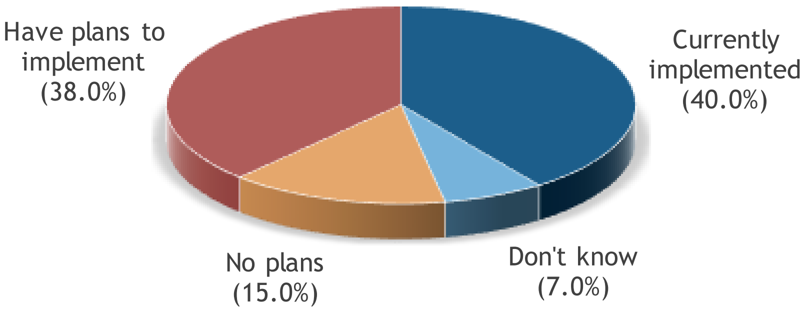
Planning for Best-of-Breed API Governance and Management

Development teams that are providing APIs are rapidly improving their skills and processes associated with API design and change management. Figure 3 shows that 40% of enterprises in IDC's 2017 *CloudView Survey* already have life-cycle capabilities associated with their APIs, while 38% of enterprises have plans to implement such capabilities. Given the relatively short life of development and use of REST APIs, the speed to maturity is very rapid.

FIGURE 3

REST API Life-Cycle Adoption

Q. What is your ability or plans to create and publish REST APIs with life-cycle capabilities?



n = 6,068

Source: IDC's *CloudView Survey,* 2017

Beyond ensuring API security and reliability, the teams providing APIs need software that will:

- Help them document the APIs
- Manage a catalog of discoverable APIs
- Provide versions of APIs and version controls
- Monitor performance
- Measure usage

For API providers selling digital services via APIs, there is a need to associate APIs with monetization models and either pass consumption data to billing systems or directly handle billing. This has driven up the demand for and use of API management software over the past several years.

API management should be able to connect to integration buses and iPaaS to handle complex integration and orchestration use cases. Services managed anywhere – whether API management, iPaaS, or integration buses – should be available as needed across any of the enterprise's integration solutions.

Trigger 4: Supporting Analytics-Driven Applications

Building the information services used to support analytics-driven applications creates the need for new types of integration tools. Whether the initiatives use the term *smart, intelligent,* or *predictive,* the goal is to harness data to identify opportunities or problems as quickly as possible. This is a common pattern in digital initiatives, including:

- Near-real-time promotions or offer management, where some combination of current interest or current location and purchasing pattern behavior indicates the prospect would be receptive to an offer (eCommerce applications as well as marketing programs aimed at messaging and mobile apps are common in businesses that hope to improve conversion rates as they improve their targeting.)
- Predictive maintenance in IoT, where a prediction model evaluates data to identify the probability of equipment failure (The goal is to reduce the cost of scheduled maintenance and prevent downtime.)
- Smarter buildings, where temperature and lighting sensors work together with analytics and automated controls to lower the cost of energy
- Smarter cities, where traffic and transit flow are optimized, work crews and emergency services are dispatched in response to social media impressions, and resources delivered via utilities are reallocated as demand shifts
- Sentiment analysis to convert social and interactive content as well as structured data into an information model that predicts customer sentiment, which is used in a variety of ways to improve customer experiences

Using predictive analytics in these and similar use cases is a velocity-intensive Big Data integration problem. The value of the design is in the system's ability to respond within a certain time window, while the accuracy of the prediction has a dependency on large volumes of data. Legacy integration approaches cannot meet the demands of processing data in a timely manner or handle the continuous integration of large volumes of data. When predictive analytics is used as part of a larger application, the IT team tends to learn that:

- Moving data from a source to the target using ETL or file transfer technology may not be timely enough.
- Transmitting individual data events over a network to the target system may not be reliable enough.
- The protocols and formats of the source data are often incompatible with traditional adapters used to connect to source or target systems.
- Traditional integration to normalize the data cannot handle the high volumes of data.
- Data may be received out of order, which is problematic when the time of data origination is critical.
- Third-party data not under control of internal resources may be leveraged, requiring additional validation and verification checks or arm's-length integration to avoid contamination or compliance issues.

Data required for prediction models typically comes from multiple sources, establishing a need to create data streams that continuously and reliably deliver data to the analytics target. Organizations begin by increasing the frequency of batch delivery using existing ETL and file transfer software. As they gain experience, however, an increasing number of enterprises switch to an event-driven architecture to handle the speed and volumes required to support near-real-time predictive analytics use cases.

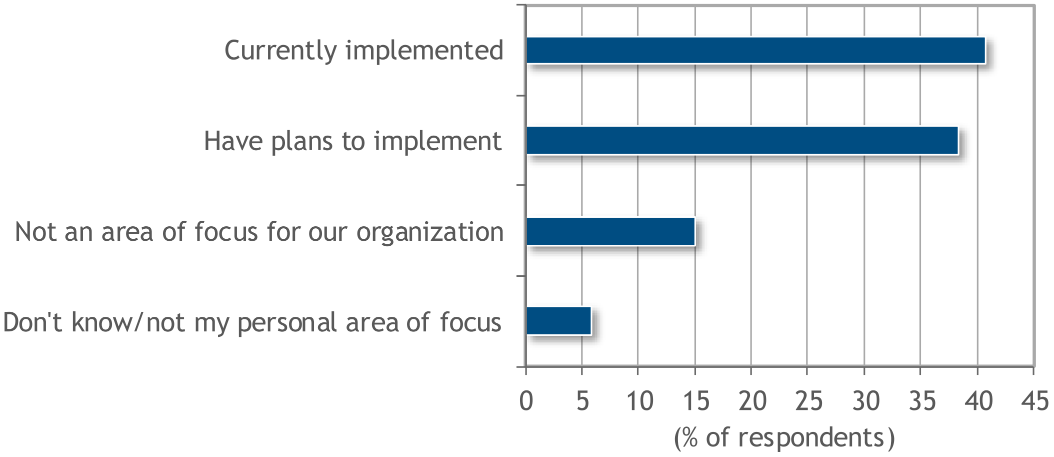
Planning to Support Event-Driven Architecture for Predictive Analytics

According to IDC research, organizations that begin the journey to predictive analytics-driven applications are often reluctant to make the switch from traditional batch-oriented data integration supporting at-rest analytics infrastructure. The primary reason for their reluctance at the beginning of these types of projects is a belief that their time windows align with their current information architecture, so there isn't a need to change.

However, our research shows that as teams learn more about response windows, the change to event-driven design is happening rapidly. Figure 4 shows that in 2017, 41% of the 6,068 respondents to IDC's *CloudView Survey* have already implemented an event-driven architecture for their real-time analytically based initiatives. In our 2016 survey, only 29% of respondents had implemented such an architecture. This year's survey indicates that 38% of respondents plan to adopt an event-driven architecture.
Last year, 30% of respondents were in early stages of adoption, and 23% of respondents were planning to adopt. This indicates progress in transitioning from early stages to successful adoption.

FIGURE 4

Event-Driven Architecture Adoption



n = 6,068

Source: IDC's *CloudView Survey,* 2017

As the survey data indicates, organizations that haven't begun the shift toward continuous collection and processing of data will be at a disadvantage, particularly where this capability is used to support competitive differentiation. Key supporting technology includes reliable messaging software to transport the data. Gateway software may be used at the source to filter irrelevant data to reduce data volumes. Once the data is delivered, streaming integration software normalizes the data in conjunction with publish-and-subscribe messaging to queue and time order the events and to forward data though processing before delivering it to the analytics target. If the analytics process determines further action is required, an event is triggered, which kicks off processes to respond.

Trigger 5: Self-Service Integration

Applications, data, and technology are becoming more diverse, but there is also a drive for simplicity and faster time to value for integration solutions. The new generation of business workers cannot wait months or years for projects to be completed and is looking for opportunities to solve its own problems. New self-service integration products provide a means for end users to integrate and automate tasks to make themselves more productive and self-sufficient.

Self-service integration encompasses a variety of scenarios, including:

- Tactical automation by end users of a process that involves connecting to two or more applications
- Situational integration associated with a project or short-term need (With this type of integration, the ease of integration provides an opportunity to automate the workflow, distribution of tasks, and delivery of data across two or more applications.)
- Use by developers for rapid integration or workflow-oriented use cases
- End-user data prep to automate the activities requiring creation of data sets that can be used for analytics, monitoring, or reporting

Products used for data or application integration have offered low-code options for several years to provide options for organizations that want trained nondeveloper specialists to manage integration activities. There are also integration tools optimized for use by developers, particularly to extend application development and include integration. These approaches are IT focused and prohibitive for end users that have tactical needs for integration. However, the following trends now make it possible to create useful products for end users:

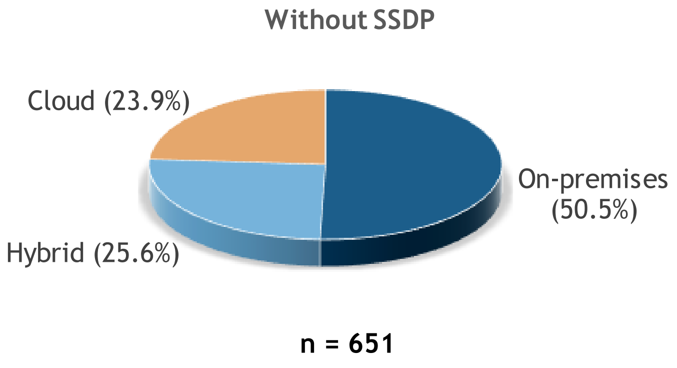
- Widespread use of APIs that make it easier to connect two or more services
- Open data access and metadata-rich formats making data more understandable and easier to use
- Growing technical literacy among nondevelopers across the workplace
- Scarcity of developer resources for tactical use cases
- Availability of simplified end-user-oriented cloud integration and workflow services

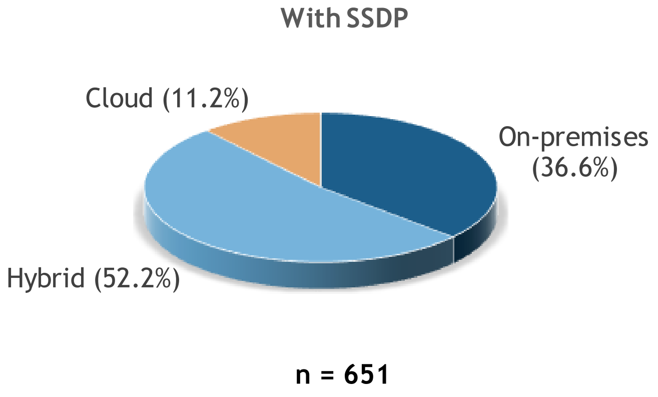
Planning to Use Commercial Software for Self-Service Integration

Self-service data preparation (SSDP), also known as data wrangling, may be considered an early mover in providing self-service functionality to the enterprise. SSDP software has been in use for many years in "name your favorite spreadsheet" software, but over the past several years, new products and software vendors have emerged to make data access, blending, wrangling, and preparation an easier task for business analysts and data professionals alike. Results of a fall 2015 IDC survey of data integration professionals illustrated that more complex hybrid data environments exist in organizations that use SSDP software compared with those that don't. Figure 5 illustrates that organizations with SSDP software have more data in hybrid and cloud-only environments compared with those that don't have SSDP software.

FIGURE 5

Hybrid Distribution of Data in Organizations with and Without SSDP





Source: IDC's *Data Integration End User Survey,* 2015

The demand for SSDP software in the enterprise foreshadowed the same demand for self-service application integration in the enterprise, and in some ways, this demand echoes the demand for business adoption of cloud applications and services outside of IT control. Organizations that are faced with end users demanding self-service integration and automation capabilities need to:

- Consider standardizing on an approved solution for self-service integration that provides a level of control and governance in keeping with user abilities and corporate policies.
- Develop technology acquisition policies that prefer services that offer open APIs accessible by self-service integration solutions.
- Monitor integration solutions implemented by end users to ensure that compliance, security, and technology policies are not being violated.
- Factor the near-term eventuality of end-user self-service into integration adoption decisions.

The creation of self-service integration became possible only with the use of APIs and better user interfaces from advances in HTML. Unifying the different tools that manage services into a common catalog makes it easier to enable a self-service tier for secure and authorized end-user access. This self-service approach aligns with business objectives to speed up the technology enablement of innovation and to provide self-determination to both the business units and the knowledge workers in those units.

THE NEED FOR A HYBRID APPROACH TO INTEGRATION

The need for hybrid computing is driven by demand as organizations support diverse cloud, edge, and on-premises locations to deploy their digital assets and application while benefiting from the flexibility to decrease costs and to increase choice. Hybrid integration is a critical element of hybrid computing because it provides a means to connect to and integrate two or more services regardless of the consumer or provider locations.

Hybrid integration is a strategy and framework for the adoption of:

- Technologies, tools, and services that work holistically to meet enterprise integration requirements, regardless of endpoint, endpoint location, architecture, and integration developer role
- Best practices for integration governance that factor in both the need to innovate by fostering development autonomy and the need to industrialize when services are regulated, can cause harm if not properly secured, can diminish an enterprise's reputation by lacking reliability, or have broader dependencies on individual services

While the five triggers discussed previously show the individual adoption of new types of integration solutions, enterprises are witnessing a sprawl of integration tools and services in production as individual teams identify their own specific, autonomous requirements per deployment location per initiative. This sprawl makes it difficult to share assets and best practices, govern when needed, and operationalize to improve efficiency and interoperability with existing processes as needed. Hybrid is not an effort to standardize adoption of new types of integration technology into a single approach but an effort to look at integration as individual services with their own APIs that should interoperate with other integration services as needed.

As the five triggers also illustrate, hybrid integration is not the replatforming of integration but an incremental path to the expansion of an enterprise's integration assets by:

- Leveraging and extending existing integration assets while adding new capabilities as the need arises
- Lowering application latency by using only the capabilities required to handle any integration activity using the most lightweight approach
- Ensuring relevant services in one integration tool are available in others
- Handing off activities from one tool to another as needed
- Speeding up integration by aligning tools with the expanding cast of workers involved in integration

RECOMMENDATIONS

- **Plan and execute against a hybrid integration road map, and use proof of concepts and technology to experiment and learn.** Many organizations struggle to understand why they need to add to their existing SOA infrastructure. They've invested in services that have worked well for years. The IT team resents or doesn't understand why it is being pressured to support alternatives, such as API management. Meanwhile, other stakeholders want faster and more lightweight approaches to integration. In these organizations, all stakeholders should work together to experiment with alternative approaches to determine when to use the web services and orchestration of the existing SOA investments and the more lightweight and more agile approaches that are built around use of REST APIs.

- **Gather requirements to understand hybrid capabilities and constraints to ensure the right level of sophistication and functionality is sought after in hybrid integration technology.** Organizations that aren't fully satisfied with their existing integration portfolio may be planning to replace existing integration investments with modern, more lightweight solutions. While this may make sense, new is not better until it is well understood. It is important to be confident in requirements, skills, and support available in the technology selection process and scope of integration projects.
- **Identify technology alternatives that are capable of interoperating across a hybrid integration stack.** It's not as important to purchase a single hybrid integration solution as it is to assess vital connection points such as libraries, service catalogs, and APIs per type of integration tool required. For example, a centralized catalog for service discovery and access may be adopted to support services that are more broadly shared across an organization. Once the common or shared elements are identified, that may point to a single offering capable of supporting hybrid integration that also accommodates most if not all of the new requirements.
- **Determine whether a hybrid platform or hybrid components are required.** The temptation to pick a platform is irresistible for adoption simplicity. Alternatively, organizations are using multiple integration technologies to support different hybrid use cases to solve the most pressing problem of the day. This latter approach can result in higher support and maintenance costs of multiple solutions, but the former approach often morphs into the latter over the long term as new use cases, technologies, and capabilities are required in the ever-changing hybrid landscape. Ensuring that a platform and components support interoperability across integration tools is critical.
- **Consider hybrid organizational constructs as part of the hybrid integration solution.** A much larger cast of resources will be involved in integration solution development as the business-IT hybrid role and teams become more prevalent in projects and day-to-day business operations. Communication and collaboration will be key to controlling integration solution sprawl across the enterprise.

CONCLUSION

Hybrid is no longer a future plan but is happening now, and organizations are supporting multiple integration technologies and roles to fulfill new integration requirements. To best meet the needs of integration overall, a hybrid integration approach will benefit because it leverages existing assets, supports incremental adoption of new technologies, provides integration tool interoperability, and provides a means for enterprises to determine how to govern where needed while providing autonomy where it is beneficial.

Aligning and organizing resources with integration development capabilities, and opening the lines of communication and collaboration among developers in hybrid business-IT roles, will increase the opportunities for success and decrease integration solution sprawl.

This new generation of hybrid integration is the younger sibling of enterprise application integration, service-oriented architecture, and event-driven architecture, growing up from a first-generation hybrid environment where Band-Aids covered wounds, uncoordinated movements resulted in trips and falls, and awkward interactions made for poor communications. Hybrid integration is maturing by learning from the mistakes of the first generation and its own experiences to become more rounded but focused, driven but cautiously optimistic, and successful through failure to deliver cohesive solutions in a new hybrid world.

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-community.com
www.idc.com