Hype Cycle for Application Architecture and Integration, 2021

Published 15 July 2021 - ID G00747497 - 82 min read

By Analyst(s): Eric Thoo, Massimo Pezzini

Initiatives: Software Engineering Technologies; Software Engineering Strategies

Composable, digital business is driving application architecture and integration to modernize and transform. Applications and software engineering leaders can leverage the technologies and practices in this Hype Cycle to meet the increasing demand for innovation, agility and scalability.

Analysis

What You Need to Know

The escalating demand for innovation, agility and scalability from application and integration capabilities compels application and software engineering leaders to pursue technological advances and practices better suited to their organizations.

However, increasingly complex, ever-changing application portfolios make it more and more challenging for software engineering leaders to modernize and deliver new application capabilities. The need for greater business agility, remaining competitive and innovation hinges on integration infrastructure composed as part of a broad, cohesive strategy to deal with the disruptive changes associated with digital business transformation.

At the same time, as businesses transition through the COVID-19 pandemic and associated economic changes, leaders responsible for modernizing application and integration infrastructure are under pressure to reduce the cost and improve the performance of their application portfolios. It is becoming critical to focus on improving business process efficiency via integration and automation.

This Hype Cycle, along with Hype Cycle for Software Engineering, 2021, reflects the position, rate of adoption and speed of maturation of innovative technologies and practices that will affect the evolution of application and integration infrastructure. Many of these innovations can have a short- to medium-term impact on application and software engineering leaders' strategies and tactics, but all collectively pave the way for the composable business revolution.

The Hype Cycle

This year's Hype Cycle for Application Architecture and Integration highlights a number of important trends:

- Frictionless sharing of applications and data — Capabilities for integration and application infrastructure are evolving with a growing emphasis on democratized low-code and no-code approaches, API-enablement, data fabric, and support for event-driven design. Implementing these capabilities will lay the foundations for user organizations' application portfolios, to readily access, assemble and provision a broad range of business functions and data for diverse use. Such implementations have led to further developments in packaged integration processes, integration platform as a service (iPaaS), data integration tools and full life cycle API management to support the changing nature of business.

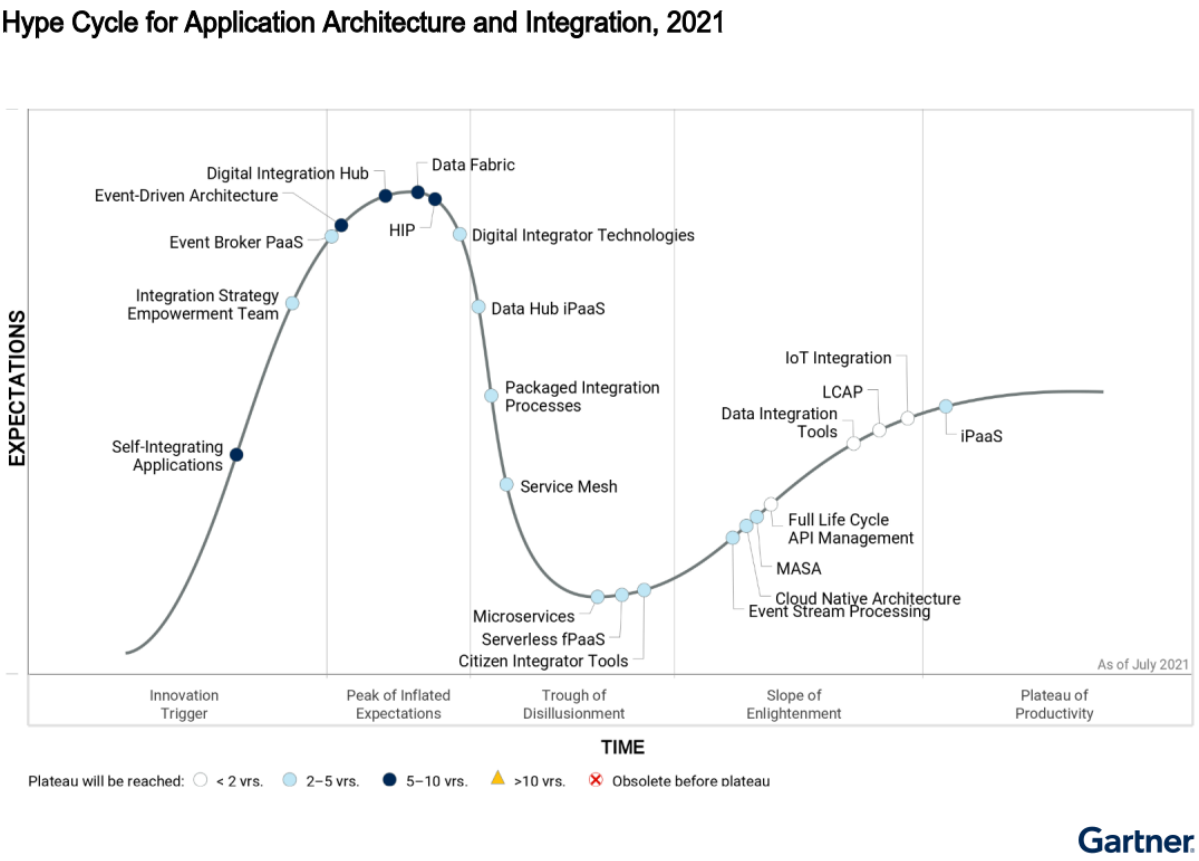
- Modernizing application and integration infrastructure — As organizations navigate pathways to the future of applications, they will require more advanced yet, at times, less mature techniques. These will include implementing application functionality as a service mesh tied to emerging elements of application architecture such as event-driven architecture, event broker platform as a service (PaaS), along with hybrid integration platform (HIP) capabilities.

- Optimizing application services and composition — Motivated to prepare for the composable business, software engineering leaders are increasingly interested in microservices, event stream processing, Internet of Things (IoT) integration and cloud-native application architecture. Additionally, the requirement to provide large-scale, high-throughput and low-latency API platforms is driving the emergence of digital integration hubs, which also provides application leaders an opportunity to reduce costs by offloading and decoupling expensive legacy systems.

- Harnessing artificial intelligence (AI)-augmentation — Software engineering leaders responsible for integration seeking to simplify integration development and improve time to value are investigating innovations through a growing use of AI in integration. Digital integrator technologies and self-integrating applications capitalize on opportunities for using augmented integration to rebalance the work of humans and AI.

- Extensible platform architecture and deployment — Maturing cloud-based and hybrid offerings are making integration and application infrastructure more broadly applicable and easier to build and manage. Leaders responsible for low-code approaches continue to adopt application PaaS, low-code application platforms and various PaaS technologies, for faster time to value and increased developer and integrator productivity. This trend is spawning diverse opportunities to exploit PaaS options across a wide range of cloud-based technologies featured in Hype Cycle for Platform as a Service, 2021.

Figure 1: Hype Cycle for Application Architecture and Integration, 2021



Source: Gartner (July 2021)

Downloadable graphic: Hype Cycle for Application Architecture and Integration, 2021

The Priority Matrix

Application and software engineering leaders should closely monitor the following innovations, which will provide the greatest benefits with the shortest times or at current mainstream adoption:

- iPaaS
- Data integration tools
- Full life cycle API management
- IoT integration
- Low-code application platform (LCAP)

Some innovations will take longer to achieve mainstream adoption relative to the ones above, but have proven to deliver high or even transformational value, including:

- Data hub iPaaS
- Digital integration hub
- Digital integrator technologies
- Event stream processing
- Event broker PaaS
- Integration strategy empowerment team
- Packaged integration processes

Adoption of these innovations requires investment in skills and presents some risks because of their intrinsic complexity or still-limited industry support. However, their market penetration is growing due to many successful deployments and associated lessons learned.

Other innovations in this Hype Cycle are either relatively mature but have a moderate impact, or their low level of industry adoption dilutes their potentially high benefits.

Finally, a small number of innovations are still in the initial stages of their life cycle, so application and software engineering leaders should assess their risks and rewards associated with adoption.

Table 1: Priority Matrix for Application Architecture and Integration, 2021

(Enlarged table in Appendix)

Benefit	*Years to Mainstream Adoption*			
	Less Than 2 Years	*2 - 5 Years*	*5 - 10 Years*	*More Than 10 Years*
Transformational		Digital Integrator Technologies Event Stream Processing	Data Fabric Self-Integrating Applications	
High	Data Integration Tools Full Life Cycle API Management IoT Integration LCAP	Data Hub iPaaS Event Broker PaaS Integration Strategy Empowerment Team iPaaS MASA Microservices Packaged Integration Processes	Digital Integration Hub Event-Driven Architecture HIP	
Moderate		Citizen Integrator Tools Cloud Native Architecture Service Mesh		
Low		Serverless fPaaS		

Source: Gartner (July 2021)

On the Rise

Self-Integrating Applications

Analysis By: Keith Guttridge, Eric Thoo

Benefit Rating: Transformational

Market Penetration: Less than 1% of target audience

Maturity: Embryonic

Definition:

Self-integrating applications will use a combination of automated service discovery, automated metadata extraction and mapping, automated process definition and automated dependency mapping to enable applications and services to integrate themselves into an existing application portfolio with minimal human interaction.

Why This Is Important

Integrating new applications and services into an application portfolio is complex and expensive. Gartner research shows that up to 65% of the cost of implementing a new ERP or CRM system is attributable to integration. The technology to enable applications to self-integrate exists in pockets, but no vendor has yet combined all the elements successfully. As applications develop the ability to discover and connect to each other, the amount of basic integration work will dramatically reduce.

Business Impact

- Improved agility, as the time to onboard applications and services is massively shortened.
- Cost savings of up to 65% when onboarding new applications and services.
- Reduced vendor lock-in, as platform migration becomes simpler.
- Greater ability to focus on differentiation and transformational initiatives, as the “keep the lights on” burden is dramatically reduced.

Drivers

- Cloud hyperscalers providing features such as service discovery, metadata extraction, intelligent document processing and natural language processing

- Automation/integration vendors providing features such as intelligent data mapping, metadata extraction, data fabric, next best action recommendations, process discovery and automated decisioning
- SaaS vendors providing features such as process automation, packaged integration processes, portfolio discovery and platform composability
- A new era in which intelligent application portfolio management is placed on top of augmented integration platforms, in order to be where the challenge is finally addressed

Obstacles

- Embedded integration features within SaaS being good enough to enable organizations to get started quickly, thus stalling investment in improving self-integration capabilities.
- A general lack of awareness of the availability of augmented integration technologies to enable self-integrating applications. Many organizations still view integration as a complex issue requiring specialist tools.
- The lack of a clear market leader that is looking to push this technology forward as the major application vendors look to protect their customer bases.

User Recommendations

Application leaders should:

- Ask their major application vendors about the interoperability of applications within their portfolios. This is the area where self-integrating applications are most likely to emerge first.
- Investigate integration vendors that have augmented artificial intelligence features to automate the process of onboarding applications and services into a portfolio.
- Manage their expectations. Self-integrating applications will provide just enough integration with the rest of the application portfolio to enable a new application to work efficiently.

Sample Vendors

Boomi; Informatica; Microsoft; Oracle; Salesforce; SAP; SnapLogic; Workato

Gartner Recommended Reading

Innovation Insight for Self-Integrating Applications

Data Fabrics Add Augmented Intelligence to Modernize Your Data Integration

Integration Strategy Empowerment Team

Analysis By: Massimo Pezzini

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Emerging

Definition:

An integration strategy empowerment team (ISET) is responsible for designing, implementing and delivering an organization's integration strategy. Hence, it's a "service provider" responsible for deploying the integration technology platform; disseminating best practices; delivering training, support and consulting services; and running an integration community of practice. It serves different personas — integration specialists, as well as ad hoc and citizen integrators — across the organization.

Why This Is Important

Most organizations have set up an integration competency center (ICC) focused on centrally delivering integration projects for the organizational units. Although highly efficient, this model shows limited ability to scale and meet organizations' fast and agile integration needs.

An ISET's goal is to overcome these limits by empowering units and individual business users to fulfill integration work in a self-service way by providing them a shared technology platform and a proper set of services.

Business Impact

- An ISET enables decentralized units to reduce time to value and increase business agility by performing integration work by themselves, while keeping their integration costs under control and ensuring overarching integration governance.
- Longer term, the ISET will empower composable business by supporting decentralized fusion teams wishing to collaboratively build new applications by composing packaged business capabilities in an agile way via orchestration, integration and automation tools.

Drivers

An ISET empowers a democratized, self-service and multipersona approach to integration by providing application teams, business units, departments, subsidiaries and business work teams with:

- A set of shared integration technologies, typically based on the hybrid integration platform (HIP) framework.
- The services (training, consulting, support, mentoring and service desk) required to take advantage of these shared technologies.

A growing number of midsize and large organizations will pursue a democratized integration approach, and therefore establish an ISET driven by:

- The ever-increasing amount of work needed to support the differentiated integration use cases stemming from business-efficiency-focused initiatives, from digitalization and from the need to address the transformation required in the postpandemic world.
- Application teams' desire to incorporate self-service integration work in their agile, DevOps-enabled application delivery processes.
- Organizations' growing adoption of composable business, which aims at enabling a wide range of personas to build new, highly focused and customized applications by assembling and integrating predefined "building blocks" (or "packaged business capabilities").

- The widespread vendors' commitment to provide integration platforms designed to appeal to a range of personas, including developers who occasionally need to perform integration work ("*ad hoc* integrators") and business users ("citizen integrators").
- The increasingly facilitated access to integration capabilities within SaaS applications and the growing offering of packaged integration processes, which appeals to ad hoc and citizen integrators.
- The expanding availability of ISET implementation methodologies and services from integration platform providers, consulting companies and system integrators.

Obstacles

- Designing, developing, deploying, managing, maintaining and evolving a suitable HIP, which requires investments in technology and technical, methodological and organizational skills.
- Effectively supporting a potentially large population of "integrators" (up to hundreds or even thousands) with highly differentiated IT skills (from advanced for integration specialists to minimal for citizen integrators).
- Preventing excessive duplication of integration efforts by encouraging knowledge, best practices and artifact (processes, transformation maps, adapters) sharing across different units via a "community of practice."
- Managing governance and compliance in a highly decentralized environment.
- The still-limited industry experience, which makes it relatively difficult to find support and skills to help the ISET set up and quickly climb the relevant organizational, methodological and technical learning curves.

User Recommendations

- Establish your ISET by taking into account its size can vary from a few full-time employees to dozens (or more), depending on your organization's nature (midsize or large) and the scale and complexity of your integration challenges.
- Explicitly and unequivocally position the ISET as a service provider focused on empowering self-service integration and not as a "regulator" setting rules, processes and policies.

- Staff the ISET with personnel with the skills and mindset needed to act as an enabling entity rather than an “integration factory.”
- Define what integration personas are in scope (and what are not) to ensure due diligence for democratized use cases.
- Establish KPIs to measure the ISET’s ability to help its constituents become more innovative, creative and empowered via self-service integration.
- Implement the ISET model in a stepwise approach, which makes it easier to justify investments in terms of business or technical benefits.

Sample Vendors

Boomi; Informatica; Mindtree; MuleSoft; PACE; Quinnox; SAP; TCS; Wipro

Gartner Recommended Reading

Integration Teams for the Digital Era Must Support New Delivery Models

Ensure Your Integration Strategy Supports Modern Integration Trends

At the Peak

Event Broker PaaS

Analysis By: Yefim Natis, Keith Guttridge

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Event broker platform as a service (ebPaaS) plays the role of the intermediary in event-driven architecture (EDA), configuring the event topics, registering event publishers and subscribers, facilitating capture and distribution of event notifications. Event brokers are built on message-oriented middleware (MOM, also known as message brokers) that delivers the essential publish-subscribe capability, then extended with additional EDA-oriented mediation and governance capabilities.

Why This Is Important

Most of the outcomes of digital business transformation depend, in part, on an organization's continuous awareness of relevant business events and its ability to respond in business real time. Event broker services facilitate detection and distribution of event notifications to application services for automated response, dashboards for human action or data stores for further analysis. The alternative to the use of a broker is custom design, which is less effective, more expensive and a higher risk.

Business Impact

Organizations that are aware of their relevant business ecosystem events are better prepared to manage unexpected interruptions and capitalize on opportunities in business moments. They are equipped for broadcasting notable events for simultaneous, multitargeted response. Event broker services enable organizations' versatility in monitoring multiple sources of events and communicating to many responders in parallel, with strong scalability, integrity and resilience.

Drivers

- Increased demand for real-time insights drives organizations to manage event streaming and stream analytics, leading in turn to event brokers for governance and coordination of event traffic.

- Increased adoption of Apache Kafka, by both businesses and leading technology vendors, promotes organizational awareness of the benefits and opportunities that event-driven application design brings.
- The migration of business applications to the cloud demands new platforms and communication infrastructure, driving many organizations to evaluate and adopt event broker services, paired with integration and API management offerings.
- The availability of multiple vendors' ebPaaS, based on open-source standards such as Pulsar, Kafka, NATS, MQTT and AMQP, provides competitive and differentiated options in event broker services for better-tuned fit to customers' use cases.
- Increased maturity of ebPaaS offerings supports more advanced capabilities in performance, data and process management, and optimization of event-driven applications.
- Most leading SaaS offerings support some event processing, increasing awareness of benefits and opportunities of event-driven application design in a large number of mainstream business and government organizations.
- Open-source event brokers are easier to operate and scale, reducing the cost of early experimentation with event-driven architecture and attracting more start-ups and other advanced software engineering teams.
- The increasing popularity of digital integration hubs and other data consolidation approaches gets the near-real-time data accuracy when consuming event streams, instead of database lookups.

Obstacles

- Desire to keep control of all aspects of infrastructure deployment leads some organizations to manual implementation of event-driven communications.
- ebPaaS offerings become too expensive as more proprietary features are added to help differentiate from the competition.
- Event broker functionality, embedded in some platform and application services, fragments control of event streaming across the organization, while delaying a systematic investment in event brokering.
- Some software engineering teams use webhooks and websockets tools to set up event notifications, delaying the full many-to-many experience of EDA that's implemented via an event broker technology.

- Lack of universally supported standards for protocols or APIs for EDA implementation increases costs and complexity of managing a large event-driven application infrastructure.

User Recommendations

- Apply the complementary strengths of service-oriented architecture (SOA) and EDA, and encourage every new project to consider the combined use of both, as appropriate in advanced mesh app and service architecture (MASA).
- Pilot experimental projects using event brokers to gain insight and skills for the upcoming more advanced projects. Even a basic pub/sub middleware service is sufficient as a precursor for a full-featured event broker.
- Give preference to ebPaaS vendors demonstrating the understanding of the full life cycle of event brokers' functionality and responsibility.
- Plan for coordinated use of an event broker and a stream analytics platform. The technologies are different and are used in combination in most advanced event broker use cases.

Sample Vendors

Amazon Web Services (AWS); Confluent; Google; IBM; Microsoft; Solace; TIBCO Software; Vantiq

Gartner Recommended Reading

Innovation Insight for Event ThinkingInnovation Insight for Event BrokersThe 5 Steps Toward Pervasive Event-Driven ArchitectureThe Impact of Event-Driven IT on API ManagementApplying Event-Driven Architecture to Modern Application Delivery Use Cases

Choosing Event Brokers: The Foundation of Your Event-Driven Architecture

Event-Driven Architecture

Analysis By: Yefim Natis, Paul Vincent

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Event-driven architecture (EDA) is a style of application design in which application components communicate indirectly by passing event notification messages via an intermediary (an event broker). EDA is a long-standing architecture model. The demands of digital business, including IoT, continuous intelligence and contextual decisions, are reintroducing EDA as newly relevant to the current generation of mainstream application designers and planners, and placing it back onto the Hype Cycle.

Why This Is Important

EDA provides advanced opportunities for scale, extensibility and resilience in applications through its asynchronous, intermediated, pub/sub design model. Monitoring business and technical events in real time enables continuous analysis of context for advanced intelligence in decision management. Organizations that are interested in digital business innovation will inevitably discover event stream analytics and EDA as powerful components of their application design.

Business Impact

EDA is a key enabling architecture pattern for a number of leading trends in digital business. An event-aware organization is more responsive in its ecosystem, more empathetic in its customer experience and more intelligent in its decision making than a purely transaction-centric business. Competence in EDA accelerates transition to digital business. Lacking event awareness, organizations may struggle to support business at competitive speeds, agility, continuous innovation and cost-efficiency.

Drivers

- Digital business demands real-time context awareness through stream analytics to support intelligent business decisions. Applications that adopt EDA become sources of such context and empower their business decision makers.
- Cloud-native application architectures, often using microservices principles, frequently use EDA to implement more flexible and scalable interservice communication.
- The popularity of Apache Kafka is creating greater awareness of EDA among mainstream organizations and their software engineering leaders.
- Many major application vendors, including Salesforce and SAP, upgraded support of EDA to their applications and application platforms in recent years, enabling more intelligent monitoring of business processes.
- IoT applications use EDA to monitor states of devices. As the use of IoT software continues to increase, so does the adoption of EDA.
- All cloud hyperscalers have added or upgraded their support for EDA by adding and extending their messaging and event brokering services.
- Application integration continues to gain adoption in mainstream organizations, and EDA is a popular model for integration design.

Obstacles

- The lack of productivity and governance tools dedicated to EDA limits the design of EDA-based applications to more advanced engineering teams, and thus delays broader adoption
- The diversity of protocol and API formats and standards for event processing limits adoption and increases implementation costs.
- The design principles of EDA are less well-understood by most development teams, in part because of the familiarity bias in favor of the common and ubiquitous request/reply model, often implemented using REST APIs.
- Event-driven communications can deliver only eventual consistency. Applications that require synchronization of distributed database updates must choose a different architecture.

User Recommendations

- Develop an inventory of EDA-related technologies and practices currently deployed; consolidate and extend EDA capabilities in technology, skills and policies.
- Aim for a pragmatic mixed use of request-driven APIs following the SOA model and EDA, including application design, software life cycle and production management.
- Adopt EDA gradually, as the industry develops required standards, best practices, and improved productivity design and management tools.
- Aim to establish EDA, along with SOA, as the common and complementary architecture patterns, both considered for all application initiatives.
- Work with business stakeholders to coordinate the discovery and analysis of business events; aim for synergy in business and technical modelling of event-driven solutions.
- Manage and mediate event channels aggressively, and understand that their value represents an in-motion view of key business processes and happenings.

Gartner Recommended Reading

Innovation Insight for Event Thinking

Innovation Insight for Event Brokers

Maturity Model For Event Driven Architecture

The Impact of Event-Driven IT on API Management

Choosing Event Brokers: The Foundation of Your Event-Driven Architecture

Digital Integration Hub

Analysis By: Massimo Pezzini, Eric Thoo

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Definition:

A digital integration hub (DIH) provides API-/event-based data access by aggregating and replicating multiple system-of-record sources into a low-latency, high-throughput data management layer that synchronizes with the systems of record via event-driven patterns. A DIH enables scalable, 24/7 data access; reduces workloads on the systems of record; and improves business agility. Organizations can reap additional value by leveraging a DIH in analytics, data integration and composition scenarios.

Why This Is Important

Digital initiatives massively leverage APIs and, increasingly, events, to unlock core business applications data and business logic. However, their success can be undermined when traditional integration architectures face severe performance, scalability and availability issues that often stem from an excessive workload generated in the systems of record. A DIH is an increasingly popular alternative to these approaches as it is able to fix these issues while delivering additional benefits.

Business Impact

- Provides digital application audiences with a rich and responsive experience
- Reduces the cost of running systems of record or limits the fees paid to SaaS providers for API consumption
- Helps enable 24/7 operations
- Improves business agility and favors composability by decoupling the API layer from the systems of records
- Maintains an up-to-date picture of fast-changing data used for analytics-based services, notification services and data integration

Drivers

DIH architectures are typically used to deliver an API platform featuring a data management layer between the systems of record and the API service layer itself. In this way, the inquiry workload generated by the API calls doesn't hit the systems of record, which are impacted only when their data must be updated.Therefore, interest in and adoption of DIH-enabled API platforms is fast growing in:

- The organizations that want to:
  - Offload the systems of record to reduce their operational costs, optimize expansive upgrades or reduce the API-limit fees paid to SaaS providers needed to support the high workload generated by digital applications frontending the systems of record themselves.
  - Improve customers' satisfaction by delivering a more responsive and data-rich user experience.
  - Accelerate the transition to composable business, digital and API economy by implementing a comprehensive set of APIs and events.
- The banking, insurance, retail, energy and utilities, higher education, transportation, hospitality and telecom industry sectors. However, other industries (for example, government and healthcare) are also showing interest in this architecture as they are subject to the above-mentioned drivers.
- Large and midsize organizations with limited skills attracted by vendors addressing the opportunity by repackaging their technology portfolios in DIH-oriented value propositions or by coming to the market with packaged DIH-enabled API platforms, at times focused on specific use cases.

Obstacles

Although the emerging packaged DIHs will make implementation easier and faster, the technical complexity in current DIH implementation limits its adoption to leading-edge organizations with sufficiently advanced skills and financial resources.

Such complexity stems from the following issues:

- Dealing with an architecture still not well known in the industry, which implies a scarcity of know-how, experience and skills in turn leading to high costs
- Assembling and managing the varied set of DIH building blocks (API gateways, application platforms, integration platforms, event brokers, data management and metadata management tools)
- Keeping the data management layer in sync with the systems of record by leveraging event-based integration tools (for example, change data capture)

- Addressing the data governance issues deriving from the creation of yet another copy or data structure out of the systems-of-record data

User Recommendations

- Adopt a DIH-enabled API platform when addressing a combination of the following requirements:
  - Providing a responsive and rich omnichannel experience for large audiences (hundreds of thousands of users or greater)
  - Reducing the cost associated with sustaining the API-generated workload hitting the systems-of-record
  - Enabling API "pull" and event "push" services to access data scattered across multiple back-end systems
  - Drastically decoupling the API services from the systems-of-records to enable composable business applications
  - Maintaining an up-to-date "single source of truth" for fast-changing data, which can be used to provide additional services (for example, custom analytics or search) or can be analyzed in real time to detect "business moments"
- Embed DIH initiatives into the overall data hub strategy for governance and integration to avoid ending up with yet another data silo.

Sample Vendors

Cinchy; Fincons Group; IBM; Informatica; Mia-Platform; Microsoft; Oracle; SAP; Sesam; Software AG

Gartner Recommended Reading

Innovation Insight: Turbocharge Your API Platform With a Digital Integration Hub

Data Fabric

Analysis By: Ehtisham Zaidi, Robert Thanaraj, Mark Beyer

Benefit Rating: Transformational

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

A data fabric is an emerging data management design for attaining flexible and reusable data integration pipelines, services and semantics. A data fabric supports various operational and analytics use cases delivered across multiple deployment and orchestration platforms. Data fabrics support a combination of different data integration styles and leverage active metadata, knowledge graphs, semantics and ML to automate and enhance data integration design and delivery.

Why This Is Important

A data fabric leverages both traditional and emerging technologies in enterprise architectural design and evolution. It is composable and supports flexibility, scalability and extensibility in an infrastructure used by humans or machines across multiple data and analytics use cases. It abstracts data management infrastructure to disintermediate any incumbent platforms, and enables data integration and delivery regardless of the number of on-premises or CSP-based data assets in use.

Business Impact

Organizations benefit as data fabric:

- Provides insights to data engineers and ultimately automates repeatable tasks in data integration, quality, data delivery, access enablement and more.
- Adds semantic knowledge for context and meaning, and provides enriched data models.
- Evolves into a self-learning model that recognizes similar data content regardless of form and structure, enabling broader connectivity to new assets.
- Monitors data assets on allocated resources for optimization and cost control.

Drivers

- A data fabric enables tracking, auditing, monitoring, reporting and evaluating data use and utilization, and data analysis for content, values, veracity of data assets in a business unit, department or organization. This results in a trusted asset capability.

- Demand for rapid comprehension and adaptation of new data assets has risen sharply and continues to accelerate — regardless of the deployed structure and format. The data fabric provides an operational model that permits use cases, users and developers to identify when data experience varies from the data expectations depicted in system designs.

- A shortage of data management professionals is increasing the demand for accurate and actively utilized metadata to make system design, data availability and data trust decisions.

- Catalogs alone are insufficient in assisting with data self-service. Data fabrics capitalize on machine learning to resolve what has been a primarily human labor effort using metadata to provide recommendations for integration design and delivery.

- Business delivery and management professionals find it difficult to identify adjacent, parallel and complementary data assets to expand their analytical models. Data fabrics have the capability to assist with graph data modeling capabilities (which is useful to preserve the context of the data along with its complex relationships), and allow the business to enrich the models with agreed upon semantics.

- Significant growth in demand and utilization of knowledge graphs of linked data as well as ML algorithms to provide actionable recommendations and insights to developers and consumers of data can be supported in a data fabric.

- Organizations have found that one or two approaches to data acquisition and integration are insufficient. Data fabrics provide capabilities to deliver integrated data through a broad range of combined data delivery styles including bulk/batch (ETL), data virtualization, message queues, use of APIs, microservices and more.

Obstacles

Data fabrics are just past the Peak of Inflated Expectations. The main challenges surrounding broad adoption are:

- Diversity of skills and platforms to build a data fabric present both technical and cultural barriers. It requires a shift from data management based upon analysis, requirements and design to one of discovery, response and recommendation.

- Intentional market hype by providers and services organizations purporting a data fabric delivery is adding to market cynicism.

- Misunderstanding and lack of knowledge in how to reconcile and manage a data fabric and a legacy data and analytics governance program that assumes all data is equal will lead to failure.
- Proprietary metadata restrictions will hamper the data fabric, which is wholly dependent upon acquiring metadata from a wide variety of data management platforms. Without metadata, the fabric requires analytic and machine learning capabilities to infer missing metadata, and while possible, will be error prone.

User Recommendations

Data and analytics leaders looking to modernize their data management with a data fabric should:

- Invest in an augmented data catalog that assists with creating a flexible data model. Enrich the model through semantics and ontologies for the business to understand and contribute to the catalog.
- Invest in data fabrics that can utilize knowledge graph constructs.
- Ensure subject matter expert support by selecting enabling technologies that allow them to enrich knowledge graphs with business semantics.
- Combine different data integration styles into your strategy (bulk/batch, message, virtualization, event, stream, replication and synchronization).
- Evaluate existing tools to determine the availability of three classes of metadata: design/run, administration/deployment and optimization/algorithmic metadata. Rate existing and candidate platforms and favor those that share the most metadata.
- Focus on a similar transparency and availability of metadata between PaaS and SaaS solutions.

Sample Vendors

Cambridge Semantics; Cinchy; CluedIn; Denodo; IBM; Informatica; Semantic Web Company; Stardog; Talend

Gartner Recommended Reading

Top Trends in Data and Analytics for 2021: Data Fabric Is the Foundation

What Is Data Fabric Design?

Top Trends in Data and Analytics for 2021: Data Fabric Is the Foundation

Emerging Technologies: Data Fabric Is the Future of Data Management

HIP

Analysis By: Massimo Pezzini

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

The hybrid integration platform (HIP) is an architectural framework that defines integration and governance capabilities and enables differently skilled personas to tackle multiple integration use cases across hybrid, multicloud setups. A HIP implementation typically consists of an assembly of diverse integration tools, from one or more providers, which are managed as a cohesive, federated and integrated whole, typically by an integration strategy empowerment team.

Why This Is Important

As organizations pursue digital and composable business initiatives, they find that the integration challenges they must address are growing in complexity and quantity. Cloud services, cloud data warehouses, ecosystems, mobile apps and Internet of Things (IoT) devices are new endpoints that they must integrate with traditional applications and data sources. The HIP helps software engineering leaders implement the integration and governance capabilities needed to integrate all their IT assets.

Business Impact

Each organization's HIP implementation will differ to reflect specific requirements. But in all cases, it will alleviate integration challenges by:

- Supporting centralized control and governance, while leveraging decentralized and collaborative integration delivery

- Improving business groups' self-sufficiency and agility by reducing their reliance on specialist integrators of limited availability
- Accelerating the time to value for integration-intensive business initiatives

Drivers

A HIP implementation typically consists of an assembly of on-premises and cloud-delivered Integration platforms, API management platforms, event brokers, metadata management tools and other use case-specific components, often from different providers. Despite the complexity of such a setup, a growing number of midsize and large organizations are implementing HIP-inspired platforms to:

- Enable a range of diverse integration personas to perform integration work in a self-service fashion. These personas include: integration specialists (professional integration developers), "ad hoc" integrators (application developers, SaaS administrators and business technologists who occasionally have to perform integration work), and citizen integrators (business users who want to automate personal or workgroup processes).
- Integrate a wide variety of endpoints residing in cloud environments, on-premises data centers, ecosystem partners, and mobile and IoT devices by leveraging APIs, events and batch mechanisms.
- Support a differentiated set of use cases, including, but not limited to, application, data, B2B, process, IoT, API and event integration; robotic process automation; and digital integration hub
- Deploy integration platform capabilities in a hybrid, multicloud scenario — that is, one featuring a combination of public and private clouds and on-premises data centers — and embed them in applications and edge systems.

Although not all organizations need to address all these requirements, almost all organizations will have to tackle some of them. Therefore, most midsize, large and global organizations will have to deploy at least a subset of the capabilities defined in the HIP framework.

Obstacles

Organizations will face key challenges when implementing an HIP-inspired platform:

- A growing number of providers have released integrated technology suites mirroring, at least in part, the HIP framework. In many instances, though, a HIP implementation requires the aggregation of multiple products from different providers — a daring technological deployment and skills-building effort for the less technically skilled organizations.
- Such a technology aggregation poses operational challenges. Use of a wide range of product-specific tools leads to suboptimal outcomes and skills duplication. However, implementing a single, cross-product "control plane" may require notable investments in technologies and skills.
- A HIP is often deployed to enable self-service integration by a variety of organizational units. To avoid chaotic duplication of efforts and high costs, software engineering leaders must define and enforce well-balanced governance policies.

User Recommendations

Software engineering leaders responsible for integration should:

- Modernize their strategy by implementing a HIP-inspired infrastructure to enable collaborative and decentralized integration delivery, carried out by a variety of personas and addressing diverse use cases.
- Implement a HIP, If they work for a large organization, by federating different vendors' products, instead of buying an out-of-the-box HIP. This will make it easier to maintain backward-compatibility with in-place integration platforms and mitigate the risk of single-vendor lock-in.
- Implement a HIP, If they work for a midsize organization, by adopting an iPaaS, whenever possible, to reduce the complexity of effort. Many iPaaSs provide a subset of the HIP framework capabilities that is generally sufficient for such organizations.
- Adopt a stepwise, initiative-by-initiative HIP implementation strategy, which is much easier to justify than a "big bang" approach and reduces complexity and risk.

Sample Vendors

Boomi; IBM; Informatica; Jitterbit; MuleSoft; Oracle; SAP; SnapLogic; Software AG; TIBCO Software

Gartner Recommended Reading

How to Deliver a Truly Hybrid Integration Platform in Steps

Digital Integrator Technologies

Analysis By: Eric Thoo, Keith Guttridge

Benefit Rating: Transformational

Market Penetration: 5% to 20% of target audience

Maturity: Emerging

Definition:

Digital integrator technologies apply artificial intelligence (AI) techniques, such as machine learning (ML) and natural language processing (NLP), to assist integration design and delivery. Areas these technologies focus on include engagement via chatbots or voice, assistance of flow automation via next best action and intelligent data mapping, and insight for processing optimization and intelligent platform operations.

Why This Is Important

Digital integrator technologies aim to simplify integration development efforts. By anticipating user needs and making next-best-step recommendations for designing an integration flow, inference algorithms identify suitable prepackaged integration content, help rectify errors in flows and improve performance. Advanced digital integrator technologies dynamically optimize integration processing and platform operations, with capabilities to auto-adjust runtime, auditing and self-healing.

Business Impact

- AI-enabled integration platforms provide automated guidance for integrating applications and data — thereby enabling less-technical integrator roles, to perform integration tasks as well as simplifying tasks for integration specialists.
- Initiatives to modernize integration platforms using AI can adopt a low-code or no-code paradigm to empower all integrator personas.

Drivers

- Delivery of integration is becoming pervasive rather than a specialist task. Digital integrator technologies empower a broad range of specialist, ad hoc and citizen integrators — thus advancing the notion for democratizing integration and enabling the composable enterprise (see Future of Applications: Delivering the Composable Enterprise and The Applications of the Future Will Be Founded on Democratized, Self-Service Integration).
- Digital integrator technologies in the form of a conversational user experience expedites efficiencies in creating integration processes or in monitoring the operational state of the integration platform.
- Increasing use by line-of-business teams connects software and makes independently designed applications and data structures work as integrated solutions.

Obstacles

- Rather than a breakthrough moment, organizations deploy incrementally when being conservative about embracing this evolution.
- Governance challenges reign when there is limited or no availability of comprehensive lineage/metadata management capabilities that track the activities and outcome. It may be difficult to ensure the traceability of integration flows or potentially substantial, consequential damages created by flawed next best steps guided by flawed data.
- Experiences that AI learns come from a variety of integrators, not only specialists but also citizen integrators who may not offer proven techniques. Poor design practices that become popular through overuse will misdirect the recommendation engine.

User Recommendations

- Provide self-guided integration designs in ways that will make implementation of integration flows easier, faster and less expensive.
- Support self-service integration tasks by business roles.
- Target simpler scenarios where past experience can be used to train ML systems in integration.
- Apply algorithms to learn and analyze integration processes to autogenerate end-to-end integration flows, understand the performance characteristics of the services involved and provide suggestions to optimize the integration process going forward.

Sample Vendors

Boomi; IBM; Informatica; Microsoft; Oracle; SAP; SnapLogic; TIBCO Software; Tray.io; Workato

Gartner Recommended Reading

Innovation Insight for AI in Integration Technologies

Sliding into the Trough

Data Hub iPaaS

Analysis By: Keith Guttridge, Eric Thoo

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Data hub integration platform as a service (iPaaS) supports integration between applications and system endpoints via a centralized intermediary store that persists the (often-normalized) data before delivery to the destination. This is different to the pass-through architecture that most established iPaaS offerings utilize today and provides additional information management and analytics capabilities as well as API access to the data store.

Why This Is Important

The data landscape for most organizations is fragementing across on-premises applications and datastores and an ever increasing number of SaaS applications and cloud data stores. This is causing increasingly complex integration and governance challenges. iPaaS vendors are converging application and data integration technologies, including data stores, to increase the appeal of their offerings as a one-stop shop for all integration needs.

Business Impact

Data Hub iPaaS provides the following benefits:

- Simplification of integrating applications and data sources.
- Improved data management and data governance.
- Improved resilience of the production environment with record/replay capabilities for integration errors and system availability.
- Simplified access to the centralized data model via APIs instead of connecting direct to application APIs.

- Analytics of the data in real time to gain business insight and potentially enable business moments.

Drivers

- Organizations looking to improve data management across on-premises and cloud based applications and data sources.
- Organizations looking to build a customer engagement hub.
- Organizations looking to build a digital integration hub.
- Organizations looking to create a hybrid transactional analytics platform.
- Organizations looking to reduce the number of vendors providing integration and data management technologies.
- iPaaS vendors converging various integration technologies.

Obstacles

- Regional compliance policies for data stores.
- Industry compliance policies for data stores .
- Organizations compliance policies for data stores.
- Preference for best-of-breed integration and data management technology.
- Organizational structure impeding unified approach to integration and data management.
- Vendor landscape is mostly small startups with only a handful of large vendors providing this service.

User Recommendations

- Recognize that this is currently still a relatively new market. The few vendors that do provide this capability often do so for relatively niche use cases. It may take several years before data hub iPaaS becomes general-purpose enough for most clients. Given that the data is stored within the data hub iPaaS, this brings with it extra challenges such as security, resilience and compliance that regular iPaaS vendors do not have to worry about.

- Combine offerings from several technology categories and vendors (such as iPaaS + data store + analytics) if the current offerings in the data hub iPaaS market are not suitable for your needs. Once established though, the combination of iPaaS, data management, real-time analytics and machine learning has the potential to significantly disrupt how organizations integrate their application and data portfolios as well as their B2B partners.

Sample Vendors

Cinchy; Domo; ForePaaS; IBM, Informatica; SAP; Sesam

Gartner Recommended Reading

Magic Quadrant for Enterprise Integration Platform as a Service

Magic Quadrant for Data Integration Tools

Packaged Integration Processes

Analysis By: Abhishek Singh

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Packaged integration processes (PIPs) are predefined integration solutions designed to automate and standardize common business processes that require integration of multiple endpoints. Examples include PIPs for order to cash, hire to retire and procure to pay. Multiple integration platform and SaaS providers, as well as service providers, make available a portfolio of PIPs to support ERP, HCM, CX and other integration requirements for both cloud and on-premises applications.

Why This Is Important

Packaged integration processes are important for delivering integrations faster in common business scenarios as they provide near to complete out-of-the-box solutions. However, they are not plug-and-play solutions and might require some customization work. PIPs help reduce time to value for “commodity” integration scenarios, increase productivity in integration delivery and facilitate democratized, self-service integration.

Business Impact

The notion of PIPs has been in the market for many years, often known as “accelerators” or “recipes” for integration between applications. However, PIPs are now becoming popular because of the prevalence of SaaS offerings. PIPs enable ad hoc and citizen integrators, as well as integration specialists, to deliver integration, thus enabling the self-service model of integration delivery.

Drivers

- Recently, the total number of vendors providing PIPs as a part of their offerings has steadily increased, primarily driven by customers' adoption of SaaS offerings. Examples include S4/HANA integration with Salesforce, NetSuite integration with Shopify and many more.
- In addition to the increased investment in PIPs by vendors, a key factor contributing to hype is the growing customer demand for very fast integration, which is shifting the integration development toward practices that facilitate citizen integrators and therefore promote the use of PIPs.
- PIP offerings are especially attractive to midsize organizations and LOBs of large organizations that have limited IT skills and cannot handle overly complex integration requirements. Many of these organizations see PIPs as a way to rapidly deliver integrations without investing massively in new skills.

Obstacles

The obstacles to the adoption of PIPs are:

- Some of the PIPs provided by the vendors pose a risk of vendor lock, which can negatively affect your integration strategy.
- Lack of flexibility of the PIP, leading to rigid integrations. Perfectly acceptable for nondifferentiating use cases, if you want to integrate the same as everyone else, but you have to change your working practices to match the PIPs.

User Recommendations

Leverage PIPs when trying to automate the integration of common and undifferentiating business processes to reduce implementation costs and accelerate time to value. Software engineering leaders responsible for integration should:

- Identify business application integration processes in the backlog that are nondifferentiating, and can therefore be implemented as quickly and inexpensively as possible via PIPs.
- Empower new integration personas, such as ad hoc and citizen integrators, to take on responsibility for integration in part by providing them with approved PIPs that they can customize and deploy themselves.
- Test your ability to implement a PIP efficiently and effectively by performing a proof of concept (POC) for each PIP identified for implementation.

Sample Vendors

Boomi; Celigo; Informatica; Jitterbit; Mulesoft; Oracle; SAP; Snaplogic; Workato; Zapier

Gartner Recommended Reading

Accelerate you Integration Delivery by Using Packaged Integration Processes

Innovation Insight for AI in Integration Technologies

Choose the Best Integration Tool for Your Needs Based on the Three Basic Patterns of IntegrationChoosing Application Integration Platform TechnologyToolkit: RFP Templates for Application Integration Platforms

Service Mesh

Analysis By: Anne Thomas

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Definition:

A service mesh is a distributed computing middleware that optimizes communications between application services within managed container systems. It provides lightweight mediation for service-to-service communications, and supports functions such as authentication, authorization, encryption, service discovery, request routing, load balancing, self-healing recovery and service instrumentation.

Why This Is Important

A service mesh is lightweight middleware for managing and monitoring service-to-service (east-west) communications, especially among microservices running in ephemeral managed container systems, such as Kubernetes. It provides visibility into service interactions, enabling proactive operations and faster diagnostics. It automates complex communication concerns, thereby improving developer productivity and ensuring that certain standards and policies are enforced consistently across applications.

Business Impact

- A service mesh helps ensure resilient and secure request-response communication between services deployed in Kubernetes and other managed container systems.
- Service mesh middleware is one of many management technologies that provide software infrastructure for distributed applications deployed in managed container systems.
- This type of middleware, along with other management and security middleware, helps provide a stable environment that supports “Day 2” operations of containerized workloads.

Drivers

- Service mesh adoption is closely aligned with microservices architectures and managed container systems like Kubernetes. Service mesh supports needed functionality in ephemeral environments, such as service discovery and mutual Transport Layer Security between services.
- As microservice deployments scale and grow more complex, DevOps teams need better ways to track operations, anticipate problems and trace errors. Service mesh automatically instruments the services and feeds logs to visualization dashboards.
- A service mesh implements the various communication stability patterns (including retries, circuit breakers and bulkheads) that enable applications to be more self-healing.
- Many managed container systems now include a service mesh, inspiring DevOps teams to use it. The hyperscale cloud vendors provide a service mesh that is also integrated with their other cloud-native services.
- Independent vendors, such as Buoyant, HashiCorp and Kong provide service meshes that support multiple environments.

Obstacles

- Service mesh technology is immature and complex, and most development teams don't need it. It can be useful when deploying microservices in Kubernetes, but it's never required.
- Users are confused by the overlap in functionality among service meshes, ingress controllers, API gateways and other API proxies. Management and interoperability among these technologies hasn't yet been addressed by the vendor community.
- Many people associate service mesh exclusively with Istio, even though it isn't the most mature product in the market and has a reputation for complexity.
- Independent service mesh solutions face challenges from the availability of platform-integrated service meshes from the major cloud and platform providers.

User Recommendations

- Delay adoption of service mesh until your teams start building applications that will get value from a mesh, such as applications deployed in managed container systems with a large number of service-to-service (east-west) interactions.
- Favor the service meshes that come integrated with your managed container system unless you have a requirement to support a federated model.
- Reduce cultural issues and turf wars by assigning service mesh ownership to a cross-functional PlatformOps team that solicits input and collaborates with networking, security and development teams.
- Accelerate knowledge transfer and consistent application of security policies by collaborating with I&O and security teams that manage existing API gateways and application delivery controllers.

Sample Vendors

Amazon Web Services; Buoyant; Decipher Technology Studios; Envoy; F5; Google; HashiCorp; Istio; Kong; Microsoft; Red Hat; Solo.io; Tetrate; VMware

Gartner Recommended Reading

How a Service Mesh Fits Into Your API Mediation Strategy

Assessing Service Mesh for Use in Microservices Architectures

Microservices

Analysis By: Anne Thomas

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

A microservice is a tightly scoped, strongly encapsulated, loosely coupled, independently deployable and independently operated application service. Microservices architecture (MSA) applies the principles of service-oriented architecture (SOA), DevOps and domain-driven design (DDD) to the delivery of distributed applications. MSA has three core objectives: Continuous delivery, precise scalability and improved stability.

Why This Is Important

Microservices architecture promises powerful application agility, scalability and resilience. It is a way to build cloud-native applications, and it facilitates continuous delivery practices. But the architecture is complex, with disruptive cultural and technical impacts. Misconceptions about microservices often push software engineering teams to use them indiscriminately, leading to overly complex architectures that fail to deliver anticipated benefits and often make things worse.

Business Impact

- Microservices increase business agility by enabling teams to incrementally deliver new features and capabilities in their software products in response to changing business requirements.
- Microservices improve the scalability of the software engineering organization by enabling small teams to work independently to deliver different services within an application.
- Microservices allow teams to change one part of an application, without the delay and cost of changing the entire application.

Drivers

- Software engineering teams adopt microservices architecture to facilitate a continuous delivery practice. The architecture must be combined with strong DevOps practices to enable teams to safely deploy small, independent features to production systems at the frequency at which they are delivered.
- When applied well, the architecture increases the independence of different parts of a large application, enabling multiple development teams to work autonomously and on their own schedules.
- Microservices architecture facilitates the building of cloud-native applications that support robust scalability and resiliency requirements.
- Microservices are frequently deployed in managed container systems, which can dynamically scale service instances in response to load requirements and automatically recover services that have failed.
- When combined with chaos engineering and resiliency practices, microservices architecture enables self-healing systems that can continue to operate through partial outages.

Obstacles

- Microservices architecture and its benefits are often misunderstood, and many software engineering teams struggle to deliver outcomes that meet senior management expectations. For example, microservices should not be shared, and they will not save you money.
- If you aren't trying to implement or improve your continuous delivery practice, you will almost certainly be disappointed with the microservices cost-benefit equation.
- Microservices architecture is complex. Developers must acquire new skills and adopt new design patterns and practices to achieve its benefits.
- Microservices disrupt traditional data management models.
- Microservices require new infrastructure.
- Microservices are related to but not the same as APIs or containers.
- Many software engineering leaders underestimate the cultural prerequisites. Success depends on applying mature agile and DevOps practices and changing team structures to align with service domains.

User Recommendations

- Set clear expectations by defining business goals and objectives for microservices architecture adoption based on realistic cost-benefit analysis of the architecture.
- Use microservices architecture as a tool to help you attain those goals. Don't view microservices as a destination.
- Avoid "microservice washing" conventional SOA, three-tier architecture and integration. Recognize the difference.
- Improve outcomes by creating guidelines for where and when software engineering teams should and should not use microservices architecture.
- Keep application architecture as simple as possible to achieve your goals.
- Address cultural concerns by aligning teams along business domain boundaries, investing in distributed computing architecture skills and improving DevOps practices.

Gartner Recommended Reading

Leading Teams to Success with Microservices Architecture

Designing Services and Microservices to Maximize Agility

10 Ways Your Microservices Adoption Will Fail — and How to Avoid Them

Serverless fPaaS

Analysis By: Anne Thomas

Benefit Rating: Low

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

Function platform as a service (fPaaS), also known as function as a service (FaaS), is a serverless execution platform for event-triggered application components known as functions. Like all serverless platforms, fPaaS enables you to run code without provisioning or managing the underlying system or application infrastructure. fPaaS pricing models allow users to pay in microincrements only for actual usage, rather than preprovisioned resources required to support projected peak loads.

Why This Is Important

fPaaS can deliver significant savings for certain types of workloads via its consumption-based micropricing model. The programming model also enables software engineers to rapidly deploy and configure new functions with little or no assistance from operations teams.

Business Impact

- An fPaaS can offer significant cost savings and virtually unlimited scalability for applications with highly variable capacity requirements.
- Serverless technologies like fPaaS abstract and commoditize infrastructure technologies, increasing developer and operator productivity, enabling organizations to respond rapidly to digital business moments and deliver new applications and features faster.

Drivers

fPaaS is gaining momentum because of:

- Potential cost savings — The micropricing model charges for small increments of compute time per invocation, which can be advantageous for small, spiky workloads. The model is less favorable for large, consistent workloads.
- Rapid solution delivery — The serverless model reduces the amount of work developers and operations teams need to do to build, deploy and configure solutions.
- Integration with hyperscale xPaaS — The hyperscale vendors make it easy to use their fPaaS with their other cloud-native xPaaS offerings.
- Broad use-case support — fPaas can support a broad spectrum of application use cases, from basic websites to complex analytical processes.

- Edge computing efficiencies — Deploying functions at the edge enables processing close to the source.
- Embedding within other xPaaS — Some xPaaS vendors embed an fPaaS in their platforms to host code components, such as rules and workflow routines. Examples include the InRule decision management platform and the Zoho low-code application platform. These systems hide the complexity of the fPaaS programming and operating model.

Obstacles

fPaaS is facing challenges because:

- Cost savings don't always materialize — fPaaS pricing isn't favorable for applications with consistently high invocation rates. Also, fPaaS-based applications often require other xPaaS, such as API management, data management and notifications.
- Latency — fPaaS-based applications can suffer from cold-start issues.
- Lock-in — fPaaS-based applications aren't portable across vendor solutions.
- Resource constraints — fPaaS is inappropriate for memory- and compute-intensive workloads.
- Lacking infrastructure — DevOps teams require development frameworks, testing and debugging tools, security services, and management technology. A fledgling ecosystem is emerging to address these requirements, although most ecosystem players focus only on Amazon Web Services (AWS) Lambda, and tooling for other fPaaS offerings is limited.
- Alternative solutions — Many developers prefer to use more general-purpose platforms, such as Kubernetes or low-code application platforms.

User Recommendations

- Minimize vendor lock-in by ensuring that your software engineering teams don't limit their skills and practices to the proprietary features of a single fPaaS.
- Estimate fPaaS costs based on expected invocation rates, memory requirements, execution times and other xPaaS dependencies. Don't presume that fPaaS is always a less expensive option.
- Evaluate whether fPaaS is a good fit for your applications and your teams' development skills. Consider aPaaS or managed container solutions as alternatives.
- Identify use cases where fPaaS offers a strategic benefit from a cost or agility perspective. Consider fPaaS for microservices deployments and for applications with highly variable or unpredictable capacity requirements.
- Avoid using fPaaS for high-memory or compute-intensive workloads. Don't port existing monolithic applications to fPaaS.
- Use fPaaS if it's an integral part of another solution, such as edge computing, decision management or low code.

Sample Vendors

Amazon Web Services; Cloudflare; Google; InRule; Microsoft; Netlify; Red Hat; Vercel; Zoho

Gartner Recommended Reading

A CIO's Guide to Serverless Computing

Security Considerations and Best Practices for Securing Serverless PaaS

Decision Point for Selecting Virtualized Compute: VMs, Containers or Serverless

Citizen Integrator Tools

Analysis By: Massimo Pezzini, Tim Faith

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

Citizen integrator tools are typically cloud-hosted services providing very intuitive, no-code integration process development tools. This way expert business users with minimal IT skills can handle relatively simple application, data and process integration tasks (or "automations") by themselves. Citizen integrator tools also provide a rich set of packaged integration processes (PIPs) that business users can rapidly configure and run with no assistance from integration specialists.

Why This Is Important

Organizations must address a growing amount of integration challenges in shorter and shorter timeframes, which implies having at their disposal several "integrators" equipped with high productivity tools.

Citizen integrator tools enable business users with minimal IT skills to perform self-service integration work, thus increasing the organization's overall delivery capacity. However their ungoverned proliferation can lead to security and compliance risks and duplicated costs.

Business Impact

Citizen integrator tools enable business users to automate tasks currently integrated via slow and error-prone manual methods. Integration specialists or ad hoc integrators (developers, SaaS administrators), also use these tools to quickly sort out simple tasks instead of using more powerful, but expensive and complex tools. Therefore, citizen integrator tools contribute to improving organizations' efficiency, productivity, agility and innovation by reducing the relevant integration costs.

Drivers

- Citizen integrator tools may help deliver business value faster, reduce integration costs and support tactical or strategic digital initiatives. These outcomes are achieved by enabling rapid, pervasive integration by a wide range of employees within (and potentially also outside) the organization. However they are available in many forms, which address different markets and needs: PIPs — At times called "recipes," these are prepackaged and configurable sets of integration flows, available stand-alone (at times for free), as embedded capabilities in SaaS or as add-ons to integration platforms. As such buyers are typically application owners or SaaS administrators. Integration software as a service (iSaaS) — Cloud services that enable users to implement brand new PIPs and to deploy, run and customize existing ones. They are typically sold to individual business users or work teams. Integration platform as a service (iPaaS) — These are targeted to professional integrators, but several iPaaS provide an iSaaS-like development environment and/or make available collections of configurable PIPs atop their platform.

- iSaaS tools have achieved notable traction in the consumer and SMB markets, thanks to their very low cost of entry, intuitive user experience, low skills demand and their rich set of PIPs. However, they have failed to penetrate other segments due to their lack of enterprise capabilities and services (for example, consulting).

- PIPs and iPaaS providing citizen-integrator-oriented capabilities are becoming more and more popular in midsize, large and global organizations. The growing use of AI, ML, NLP and chatbots in iPaaS offerings to facilitate integration development is augmenting their appeal for citizen integrators, thus further favoring adoption.

Obstacles

- Business users are increasingly technology savvy and often driven by time-to-market pressures, especially in the post-pandemic era that requires fast reaction to sudden changes in the business environment. This will increasingly urge them to adopt cloud citizen integrator tools, rather than wait for their IT colleagues to methodically perform integration work for them. However, this will create a few challenges: If not framed in a proper governance model, citizen integrator tools adoption by business users will inevitably lead to security, compliance, management and governance issues.
- Although some central IT departments will adopt a positive attitude and proactively address these challenges, others will try to stop business users from leveraging these tools to prevent these risks. In addition, excessive expectations for ultra-easy, super-fast integration and the simplistic nature of some citizen integrator tools may still lead to disappointment, thus hindering their more widespread adoption.

User Recommendations

Software engineering leaders responsible for integration should:

- Engage with business teams to understand their automation needs and identify to what extent citizen integrator tools can improve their responsiveness and productivity.
- Approve, certify and support a set of citizen integrator tools that meet these needs and make them available to internal users in a self-service way. This will help to prevent the uncontrolled proliferation of similar tools and maintain a degree of centralized governance and monitoring.
- Beware when selecting citizen integrator tools that: some tools are rather simplistic and lowest-common-denominator in nature; and PIPs provided by SaaS vendors may have been designed for a professional IT developer audience.
- Give preference to providers that can support both “professional” and citizen integrator requirements when selecting an iPaaS.
- Frame citizen integrator tools, including those embedded in SaaS applications, in your hybrid integration platform (HIP) strategies.

Sample Vendors

Adeptia; Celonis (Integromat); elastic.io; IFTTT; Microsoft; Quickbase; Tray.io; Workato; Zapier

Gartner Recommended Reading

Accelerate Your Integration Delivery by Using Packaged Integration Processes

The Applications of the Future Will Be Founded on Democratized, Self-Service Integration

Quick Answer: When to Use (or Not Use) Embedded Integration Features Provided by Your SaaS Vendor

Climbing the Slope

Event Stream Processing

Analysis By: W. Roy Schulte, Pieter den Hamer

Benefit Rating: Transformational

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Event stream processing (ESP) is computing that is performed on streaming data (sequences of event objects) for the purpose of stream analytics or stream data integration. ESP is typically applied to data as it arrives (data "in motion"). It enables situation awareness and near-real-time responses to threats and opportunities as they emerge, or it stores data streams for use in subsequent applications.

Why This Is Important

ESP is a key enabler of continuous intelligence and related real-time aspects of digital business. ESP's data-in-motion architecture is a radical departure from conventional data-at-rest approaches that historically dominated computing. ESP products have progressed from niche innovation to proven technology and now reach into the early majority of users. ESP will reach the Plateau of Productivity within several years and eventually be adopted by multiple departments within every large company.

Business Impact

ESP transformed financial markets and became essential to telecommunication networks, smart electrical grids and some IoT, supply chain, fleet management, and other transportation operations. Most of the growth in ESP during the next 10 years will come from areas where it is already established, especially IoT and customer experience management. Stream analytics from ESP platforms provides situation awareness through dashboards and alerts, and detects anomalies and other significant patterns.

Drivers

Five factors are driving ESP growth:

- Companies have ever-increasing amounts of streaming data from sensors, meters, digital control systems, corporate websites, transactional applications, social computing platforms, news and weather feeds, data brokers, government agencies and business partners.
- Business is demanding more real-time, continuous intelligence for better situation awareness and faster, more-precise and nuanced decisions.
- ESP products have become widely available, in part because open-source ESP technology has made it less expensive for more vendors to offer ESP. More than 40 ESP platforms or cloud ESP services are available. All software megavendors offer at least one ESP product and numerous small-to-midsize specialists also compete in this market.
- ESP products have matured into stable, well-rounded products with many thousands of applications (overall) in reliable production.
- Vendors are adding expressive, easy-to-use development interfaces that enable faster application development. Power users can build some kinds of ESP applications through the use of low-code techniques and off-the-shelf templates.

Obstacles

- ESP platforms are overkill for most applications that process low or moderate volumes of streaming data (e.g., under 1000 events per second), or do not require fast response times (e.g., less than a minute).
- Many ESP products required low-level programming in Java, Scala or proprietary event processing languages until fairly recently. The spread of SQL as a popular ESP development language has ameliorated this concern for some applications, although SQL has limitations. A new generation of low-code development paradigms has emerged to further enhance developer productivity but is still limited to a minority of ESP products.
- Many architects and software engineers are still unfamiliar with the design techniques and products that enable ESP on data in motion. They are more familiar with processing data at rest in databases and other data stores, so they use those techniques by default unless business requirements force them to use ESP.

User Recommendations

- Use ESP platforms when conventional data-at-rest architectures cannot process high-volume event streams fast enough to meet business requirements.
- Acquire ESP functionality by using a SaaS offering, IoT platform or an off-the-shelf application that has embedded CEP logic if a product that targets their specific business requirements is available.
- Use vendor-supported closed-source platforms or open-core products that mix open-source with value-added closed-source extensions for mainstream applications that require enterprise-level support and a full set of features. Use free, community-supported, open-source ESP platforms if their developers are familiar with open-source software and license fees are more important than staff costs.
- Use ESP products that are optimized for stream data integration to ingest, filter, enrich, transform and store event streams in a file or database for later use.

Sample Vendors

Amazon; Confluent; Google; IBM; Informatica; Microsoft; Oracle; SAS; Software AG; TIBCO Software

Gartner Recommended Reading

Market Guide for Event Stream Processing

Adopt Stream Data Integration to Meet Your Real-Time Data Integration and Analytics Requirements

Market Share Analysis: Event Stream Processing (ESP) Platforms, Worldwide, 2020

Cloud Native Architecture

Analysis By: Anne Thomas

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Cloud native architecture is the set of application architecture principles and design patterns that enables applications to fully utilize the agility, scalability, resiliency, elasticity, on-demand and economies of scale benefits provided by cloud computing. Cloud native applications are architected to be latency-aware, instrumented, failure-aware, event-driven, secure, parallelizable, automated and resource-consumption-aware (LIFESPAR).

Why This Is Important

Many organizations are moving to cloud native architecture as they shift their application workloads to cloud native application platforms. Cloud native principles and patterns enable applications to operate efficiently in a dynamic environment and make the most of cloud benefits. Organizations that simply "lift and shift" legacy applications to cloud native platforms often find that the applications perform poorly, consume excessive resources and aren't able to fail and recover gracefully.

Business Impact

- Cloud native architecture ensures that applications can take full advantage of a cloud platform's capabilities to deliver agility, scalability and resilience.
- It enables DevOps teams to more effectively use cloud self-service and automation capabilities to support continuous delivery of new features and capabilities.
- It can also improve system performance and business continuity, and it can lower costs by optimizing resource utilization.

Drivers

- Organizations want to make the most of cloud computing to support their digital business initiatives, but they can't fully exploit cloud platform benefits without cloud native architecture.
- Software engineering teams are adopting cloud native architecture to support cloud native DevOps practices, including self-service and automated provisioning, blue/green deployments, and canary deployments. A basic set of rules known as the " twelve-factor app" ensures that applications can support these practices.
- Cloud native architecture includes practices such as application decomposition (following the mesh app and service architecture [MASA] structure), containerization, configuration as code, and stateless services.

Obstacles

- Cloud native architecture adds a level of complexity to applications, and development teams require new skills, new frameworks, and new technology to be successful.
- Without proper education, architects and developers can apply the principles poorly and deliver applications that fail to deliver the expected benefits. This leads to developer frustration in adopting the new patterns and practices.
- Not every cloud-hosted application needs to be fully cloud-native and developers may be confused about when they need to use particular patterns to address their specific application requirements.

User Recommendations

- Use the twelve-factor app rules and the LIFESPAR architecture principles to build cloud native applications.
- Incorporate cloud native design principles in all new applications irrespective of whether you currently plan to deploy them in the cloud. All new applications should be able to safely run on a cloud platform, even if it doesn't fully utilize cloud characteristics.
- Apply cloud native design principles as you modernize legacy applications that you plan to port to a cloud platform to ensure that they can tolerate ephemeral or unreliable infrastructure. Otherwise, they are likely to experience stability and reliability issues.
- Select an application platform that matches your cloud native architecture maturity and priorities. Recognize that low-code platforms enable rapid development of cloud-ready applications, but they won't provide you with the full flexibility to apply LIFESPAR and twelve-factor principles.

Sample Vendors

Amazon Web Services; Google; Microsoft; Red Hat; VMware

Gartner Recommended Reading

How to Help Software Engineering Teams Modernize Their Application Architecture Skills

How to Modernize Your Application to Adopt Cloud-Native Architecture

A Guidance Framework for Modernizing Java EE Applications

Guidance Framework for Modernizing Microsoft .NET Applications

MASA

Analysis By: Anne Thomas

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

Mesh app and service architecture (MASA) is a composition-based application architecture that enables application delivery teams to respond rapidly to changing business demands and support multiple experiences. A MASA application is implemented as a mesh of distributed, loosely coupled, autonomous and shareable components, including multiple fit-for-purpose apps supporting unique experiences and composable multigrained back-end services. Apps and services communicate via mediated APIs.

Why This Is Important

MASA describes the foundation for modern business application architecture. It enables multiexperience applications. It supports agility and rapid delivery of new capabilities. It is a technical architecture that enables composability. It facilitates incremental modernization of legacy applications while providing mechanisms that ensure security and robust operations.

Business Impact

MASA enables organizations to respond rapidly to opportunities and disruptions through extension and recomposition. It enables multiexperience. It enables cloud-native architectures. MASA is an architecture for individual applications, as well as a strategy for modernizing the application portfolio. It provides an evolutionary approach that enables development teams to iteratively modernize their applications in direct response to business priorities.

Drivers

The initial impetus to shift to MASA was to enable existing applications to add support for mobile experiences. But MASA enables many other critical application capabilities, such as:

- Multiple experiences for different types of devices and modalities, such as voice, touch, wearables and immersive technologies.
- Distinct, optimized experiences for the different personas that use an application.
- Rapid response to disruptive events and changing business priorities via composition of existing services and creation of new experiences.
- Greater flexibility through loose coupling of components.
- Improved application performance, scalability, security and resilience through intelligent mediation.

Obstacles

- The biggest obstacle to MASA is the extensive technical debt embedded in existing application portfolios.
- MASA requires application functionality to be encapsulated and exposed via APIs. Those legacy applications must be modernized and refactored to convert the embedded business logic into composable services.
- The architecture enables iterative modernization, but it will take years (perhaps decades) to modernize the entire application portfolio.
- MASA also requires an investment in API mediation and multiexperience technologies.

User Recommendations

Software engineering leaders responsible for architecture and infrastructure:

- Ensure that development teams have competence in user experience design, service-oriented architecture, API design and domain-driven design.
- Task your architects with updating existing technical architectures, governance mechanisms and success metrics to align them with using a MASA approach to modernize application delivery.

- Analyze your business's digital transformation roadmap and identify and prioritize applications to modernize to support those needs.
- Encapsulate data and functionality in existing applications and expose them via APIs to enable composition.
- Mediate API traffic to apply governance, performance and security policies.
- Take a pragmatic approach to creating services: encapsulate, extend or refactor existing applications or build new services.
- Determine appropriate service granularity based on your objectives. Don't presume that all services within MASA must be microservices.

Gartner Recommended Reading

Adopt a Mesh App and Service Architecture to Power Your Digital Business

3 Key Practices to Enable Your Multiexperience Development Strategy

Leading Teams to Success with Microservices Architecture

Mediated APIs: An Essential Application Architecture for Digital Business

How to Apply Design and Architecture to Multiexperience Application Development

Accelerate Digital Transformation With an API-Centric (Headless) Architecture for Enterprise Applications

Full Life Cycle API Management

Analysis By: Shameen Pillai

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Definition:

Full life cycle API management involves the planning, design, implementation, testing, publication, operation, consumption, versioning and retirement of APIs. API management tools enable API ecosystems and publishing APIs that securely operate and collect analytics for monitoring and business value reporting. These capabilities are typically packaged as a combination of developer portal, API gateway, API design, development and testing tools as well as policy management and analytics.

Why This Is Important

APIs are widely used and accepted as the primary choice to connect systems, applications and things to build modern composable software architectures. The use of APIs as digital products monetized directly or indirectly is also on the rise. Advancing digital transformation initiatives across the world have emphasized the need for creation, management, operations and security of APIs and made full life cycle API management an essential foundational capability every organization must have.

Business Impact

Full life cycle API management provides the framework and tools necessary to manage and govern APIs that are foundational elements of multiexperience applications, composable architectures and key enablers of digital transformations. It enables the creation of API products, which may be directly or indirectly monetized, while its security features serve to protect organizations from the business impact of API breaches.

Drivers

- Organizations are facing an explosion of APIs, stemming from the need to connect systems, devices and other businesses. Use of APIs in internal, external, B2B, private and public sharing of data is driving up the need to manage and govern APIs using full life cycle API management.
- APIs that package data, services and insights are increasingly being treated as products that are monetized (directly or indirectly) and enable platform business models. Full life cycle API management provides the tooling to treat APIs as products.
- Digital transformation drives increased use of APIs, which in turn increases the demand for API management.
- APIs provide the foundational elements required for growth acceleration and business resilience.
- Developer mind share for APIs is growing. Newer approaches to event-based APIs, design innovations and modeling approaches such as GraphQL, are driving interest, experimentation and growth in full life cycle API management.
- Cloud adoption and cloud-native architectural approaches to computing (including serverless computing) are increasing the use of APIs in software engineering architectures, especially in the context of microservices, service mesh and serverless.
- Regulated, industry-specific initiatives such as open banking and connected healthcare, along with nonregulated, opportunistic approaches in other industries are increasing the demand for full life cycle API management.

Obstacles

- Lack of commitment to adequate organizational governance processes hinders adoption of full life cycle API management. This can be due to lack of skills or know-how, or due to too much focus on bureaucratic approaches rather than federated and automated governance approaches.
- Lack of strategic focus on business value (quantifiable business growth or operational efficiencies) and too much focus on technical use cases can disengage business users and sponsors. This is particularly apparent in cases where API programs fail to deliver promised return on investment.
- Traditional, single-gateway approaches to API management do not fit well to a modern, distributed application environment.
- Partial or full set of API management capabilities provided by vendors in other markets such as application development, integration platforms, security solutions, B2B offerings, etc., can create confusion and potentially shrink the market opportunities.

User Recommendations

- Use full life cycle API management to power your API strategy that addresses both technical and business requirements for APIs. Select offerings that have the ability to address needs well beyond the first year.
- Treat APIs as products managed by API product managers in a federated API platform team.
- Choose a functionally broad API management solution that supports modern API trends, including microservices, multigateway and multicloud architectures. Ensure that the chosen solution covers the entire API life cycle, not just the runtime or operational aspects.
- Use full life cycle API management to enable governance of all APIs (not just APIs you produce), including third-party (private or public) APIs you consume.
- Question full life cycle API management vendors on their support for automation of API validation and other capabilities, as well as their support for a modern, low-footprint API gateway.

Sample Vendors

Axway; Google; IBM; Kong; Microsoft; MuleSoft; Software AG

Gartner Recommended Reading

Magic Quadrant for Full Life Cycle API Management

The Evolving Role of the API Product Manager in Digital Product Management

How to Use KPIs to Measure the Business Value of APIs

API Security: What You Need to Do to Protect Your APIs

Top 10 Things Software Engineering Leaders Need to Know About APIs

Data Integration Tools

Analysis By: Ehtisham Zaidi, Mark Beyer, Robert Thanaraj

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Definition:

Data integration tools enable the design and implementation of data access and delivery capabilities that allow independently designed data structures to be leveraged together. Data integration tools have matured from supporting traditional bulk/batch integration scenarios to now supporting a combination of modern delivery styles (such as data virtualization and stream data integration). Data integration tools are now expected to support hybrid and multicloud integration scenarios.

Why This Is Important

Data integration tools are needed by organizations to support distributed data management and deliver data at all latencies across a range of use cases. These include MDM, analytics, data science, data warehousing, and multicloud and hybrid cloud integration.

Data integration tool suites are expected to deliver simpler interfaces to support less-skilled roles like citizen integrators. Growing requirements for automated data integration also require support for data fabric architectures.

Business Impact

Organizations adopting mature data integration tools increasingly exploit comprehensive data access and delivery capabilities. They get immediate benefits in the form of:

- Reduced time to integrated data delivery
- Cost savings (by reduced integration technical debt)
- Quality enhancements (for analytics/data science products)
- Flexibility (to access new data sources)

Integration tools that support data fabric designs will increase the productivity of data engineering and data science teams.

Drivers

- Ability to execute data integration in a hyperconnected infrastructure (irrespective of structure and origins) and the ability to automate transformations through embedded ML capabilities are the most important drivers for organizations investing in modern data integration tools.
- Traditional data integration architectures and tools (which focused solely on replicating data) are slow in delivering semantically enriched and integrated datasets that are ready for analytics. This exacerbates data integration tools to provision a mix of variable latency, granularity, physical and virtualized data delivery. This is another major reason to invest in these tools.
- Activities for self-service data access and data preparation by skilled data engineers, citizen integrators and other non-IT roles spur requirements for new data integration tools.
- While traditional data integration tools have now become mature on technical metadata ingestion and analysis to support data integration activities, there is still room for maturity for data integration vendors to introduce capabilities to harness and leverage “active” metadata. Organizations must therefore investigate and adopt data integration tools that can not only work with all forms of metadata, but also share it bidirectionally with other data management tools (e.g., data quality tools) to support data fabric architectures for automation.
- Dynamic data fabric designs bring together physical infrastructure design, semantic tiers, prebuilt services, APIs, microservices and integration processes to connect to reusable integrated data. Vendors will continue to add data integration functionality or acquire technology in these areas.

Obstacles

- Tightly integrated data integration tool suites in which all components share metadata (both active and passive), design environment, administration and data quality support remain an area for improvement in the data integration tools market.
- The popularity of data preparation (and other self-service ingestion tools), with the sole focus on analytics use cases demonstrated, will create some confusion in the market, slowing the advance of data integration tool suites.
- The demand for a seamless integration platform that spans and combines multiple data delivery styles (batch with data virtualization, for example), multiple deployment options (hybrid and multicloud) and multiple personas currently exceeds the capabilities of most offerings.
- Most existing data integration tools are limited in their ability to collect and analyze all forms of metadata to provide actionable insights to data engineering teams to support automation.

User Recommendations

- Assess your data integration capability needs to identify gaps in critical skill sets, tools, techniques and architecture needed to position data integration as a strategic discipline at the core of your data management strategy.
- Review current data integration tools to determine if you are leveraging the capabilities they offer. These may include the ability to deploy core elements (including connectivity, transformation and movement) in a range of different data delivery styles driven by common metadata, modeling, design and administration environments.
- Identify and implement a portfolio-based approach to your integration strategy that extends beyond consolidating data via ETL to include stream data integration, event recognition and data virtualization.
- Make automation of data integration, ingestion and orchestration activities your primary goal for the year, and focus on those data integration tools that can support data fabric designs.

Sample Vendors

Denodo; Fivetran; HVR; IBM; Informatica; Matillion; Precisely (Syncsort); Qlik; Talend; TIBCO

Gartner Recommended Reading

Magic Quadrant for Data Integration Tools

Critical Capabilities for Data Integration Tools

Market Share Analysis: Data Integration Tools, Worldwide, 2019

Position Your Product to Benefit From the Rise of Intercloud Data Integration

Data Fabrics Add Augmented Intelligence to Modernize Your Data Integration

LCAP

Analysis By: Paul Vincent, Jason Wong, Yefim Natis

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Mature mainstream

Definition:

A low-code application platform (LCAP) is an application platform that supports low-code declarative and often visual, programming abstractions, such as model-driven and metadata-based programming. An LCAP supports end-user interface creation, includes a database, and is used for rapid application development with simplified software development life cycle tooling.

Why This Is Important

LCAPs are one of the most popular types of development tools supporting the low-code paradigm. They support general web and mobile application development with high productivity while reducing the need for deep developer skills, and are mostly cloud-based. They are widely adopted for developer personas ranging from enterprise software developers to citizen developers. Over 200 vendors support a wide variety of business use cases and industry specializations for digital business automation.

Business Impact

Speeding up application development while reducing developer effort can be transformative for business IT. Businesses adopt LCAP to deliver more automation and reduce their application backlogs as well as enable more self-service application development. Mostly cloud-based, LCAP vendors are also accelerating development of new capabilities to increase their use-case coverage and justify their subscription costs.

Drivers

- LCAPs have evolved from rapid application development, business process management technologies and SaaS extension platforms through their evolution of common capabilities around user interface, database, business logic definition, process orchestration and integration of now-ubiquitous API services. The demand to deliver new business automations through applications continues to outstrip conventional application development capacity. This is despite the rise of SaaS usage for standard business services — indeed the latter has resulted in more demands for custom-made extensions that has resulted in a large part of the LCAP marketing being SaaS vendors' LCAP: the market is dominated by Salesforce.

- Through the requirement for LCAP to enable competitive SaaS and complete applications, they have evolved toward multifunction capabilities. LCAPs overlap with the business process automation/iBPMS market for workflow use cases, and the MXDP market for user-interface-driven use cases.

- Some vendors have recently focused on cloud-native scalability to support larger B2C deployments and deeper governance tooling to support remote and distributed developers to enable postpandemic business and IT development fusion team structures. Through support for composing applications from multiple API and service types, LCAPs can cover an increasingly large set of enterprise application requirements, with some enterprises starting to choose them as a strategic application platform.

Obstacles

- Current LCAP market share is heavily biased toward some very large hyperscalers and a few successful independent vendors. However, Gartner commonly speaks with clients that have multiple LCAP offerings deployed across the enterprise.
- LCAPs have been implemented by the main SaaS platform vendors whose market dominance and deep pockets could diminish the opportunities for a large number of small LCAP vendors. However, this really means that for most enterprises the question is not whether to adopt LCAP, but which LCAP(s) will they focus on and invest in.
- LCAP like most low-code trades productivity for vendor lock-in (of both applications and developer skills). Vendor cancellations (like Google App Maker) do occur.
- Licensing models vary across vendors and can be regularly updated by vendors, and may not scale for new use cases. This can lead to vendor disillusionment!

User Recommendations

Software engineering leaders, CIOs and CTOs should:

- Evaluate application lock-ins due to the lack of portability or standards for low-code models. This technical debt will accumulate fast, and means that vendor relationships (and contracts) need to be considered strategic. Architecture needs should be considered — for example whether to use the built-in database for all use cases.
- Weigh annual subscriptions against the productivity benefits during application development (and maintenance). Subscription costs for LCAP are typically per end user, encouraging maximum LCAP adoption per paid-up user.
- Ensure developers have access to the tools that make them productive, whether LCAP or others, and are governed accordingly. Different developers with different skill sets will vary in their successful adoption of LCAP.
- Assess LCAP vendors. The large number of vendors implies possible future market instability, although to date there have been few cases of LCAP retirements.

Sample Vendors

Appian; Betty Blocks; Kintone; Mendix; Microsoft; Oracle; OutSystems; Quickbase; Salesforce; ServiceNow

Gartner Recommended Reading

Magic Quadrant for Enterprise Low-Code Application Platforms

Critical Capabilities for Enterprise Low-Code Application Platforms

Identify and Evaluate Your Next Low-Code Development Technologies

IoT Integration

Analysis By: Benoit Lheureux

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

IoT integration refers to the integration strategies and technologies needed to assemble end-to-end IoT-enabled business solutions. IoT-specific integration challenges include integrating IoT devices, operational technology (OT), digital twins and multiple IoT platforms. More traditional IoT project integration challenges include integrating IoT applications and digital twins with enterprise applications, data, business processes, SaaS applications, B2B ecosystem partners and mobile apps.

Why This Is Important

Every IoT project requires significant integration work — some unique to IoT projects — to enable IoT devices, IoT applications and various existing business applications to work well together. In a recent survey, a majority (71%) of companies reported that they made moderate to major investments in their integration strategy to support IoT projects (see Survey Analysis: Companies Recognize Integration as a Key Competency for Internet of Things Projects).

Business Impact

- IoT integration is an essential functional requirement for all IoT projects.
- All software engineering leaders (SWELs) and application leaders responsible for IoT projects must address IoT integration, and to successfully deliver IoT products, they will either have to train or hire software engineers with unique-to-IoT integration skills.
- Special integration skills and tools are often needed for IoT projects (e.g., for OT integration).

Drivers

- Extraordinary IoT project technology heterogeneity — e.g., multiple types and OEMs offering IoT devices, brand-new and decades-old products and equipment, diverse IoT device data heterogeneity, and diverse applications systems to be integrated.
- A proliferating desire to ingest and analyze IoT data to support data-driven business decisions.
- Proliferation of IoT projects (for which IoT integration is always required).
- IoT integration is a key challenge for IoT projects. A Gartner survey found that companies can't rely on a "one-size-fits-all" approach to IoT device integration, and had to integrate their IoT projects with many different types of IT endpoints (see Survey Analysis: Companies Recognize Integration as a Key Competency for Internet of Things Projects).
- To fully realize the benefits of IoT, companies will eventually need to integrate new IoT technologies with legacy (i.e., pre-IoT) business applications and software using new, enhanced workflow (see How Can Organizations Integrate IoT Digital Twins and Enterprise Applications?).
- Complex, distributed IoT projects often involve a mix of IoT devices, IoT platforms, business applications, mobile apps, cloud services and (often) external business partners. Such complex IT projects are needed to enable new IoT-enabled outcomes — e.g., self-diagnosing and self-repairing assets and equipment, "lights-out-manufacturing," or product-as-a-service.
- The need for owner-operators in heavy-asset industries (e.g., manufacturing, utilities, oil and gas production) to integrate IoT-connected devices and digital twins hosted on multiple IoT platforms.
- A growing need to align time-series data generated by various IoT-connected assets and equipment with traditional EAM master data (e.g., BOM) for the same assets and equipment.
- Performance and scalability — that is, potentially large numbers of IoT devices, products and equipment with high API throughputs and large volumes of time series data must be integrated.

Obstacles

- SWELs tend to focus on building software engineering teams for IoT projects with skills in IoT data, applications and analytics — rather than skills in IoT integration.
- Few engineers have IoT software development skills, and even fewer have IoT integration skills.
- TSPs investing in IoT products (e.g., IoT platforms) tend to focus more on IoT data, applications and analytics, rather than on integration, which creates integration functionality gaps.
- A function gap among many general-purpose integration tools (e.g., ESB, iPaaS) for many of the IoT-specific integration needs of IoT projects (e.g., IoT devices, OT equipment or LOB applications such as MES). While many integration tools support modern IoT device protocols (e.g., APIs, MQTT and OPC-UA), most cannot connect to older, "brownfield" OT equipment.
- IoT integration products focused on OT integration (e.g., OSIsoft, Skkynet) may be needed and must be licensed separately.
- Perceived high cost of IoT-specific integration tools or services.

User Recommendations

SWELs for IoT projects should:

- Clearly identify what IoT integration functionality is needed for IoT projects (see Survey Analysis: Companies Recognize Integration as a Key Competency for Internet of Things Projects).
- Avoid simplistic approaches to IoT integration (e.g., "APIs = integration") that cannot, alone, address all your needs (e.g., does not also address functionality such as IoT data translation, OT integration).
- Hire and/or train software engineers with IoT integration skills.
- Confirm the availability of required IoT integration capabilities for any IoT product or service (see Critical Capabilities for Industrial IoT Platforms).
- Modernize your B2B integration strategy (either via EDI or APIs — see Use APIs to Modernize EDI for B2B Ecosystem Integration) to enable IoT project integration with business partners.

- Align your IoT integration skills with your company's overall integration strategy (see How to Deliver a Truly Hybrid Integration Platform in Steps).

Sample Vendors

Alleantia; Dell Boomi; Informatica; Microsoft; Reekoh; Salesforce (MuleSoft); Sky Republic; SnapLogic; Software AG; Solace

Gartner Recommended Reading

Survey Analysis: Companies Recognize Integration as a Key Competency for Internet of Things Projects,

What Should I Do To Ensure Digital Twin Success?

Critical Capabilities for Industrial IoT Platforms

Use APIs to Modernize EDI for B2B Ecosystem Integration

How to Deliver a Truly Hybrid Integration Platform in Steps

Use the IoT Platform Solution Reference Model to Help Design Your End-to-End IoT Business Solutions

Entering the Plateau

iPaaS

Analysis By: Massimo Pezzini

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

An integration platform as a service (iPaaS) is a vendor-managed suite of cloud services that delivers a mix of application, data, B2B and other integration functionality, as well as API management and event brokering. An iPaaS targets multiple personas: integration specialists, application developers and business users. Organizations adopt iPaaS as their first move into integration technology, and to complement or replace classic integration software so as to enable democratized integration.

Why This Is Important

Organizations' accelerating shift to the cloud is boosting the iPaaS market (up 38% in 2020) and has made it the biggest segment ($3.7 billion) of the integration platform technology market. Its functional breadth makes it the natural alternative to classic integration software (ESB, ETL and B2B gateway software) for large organizations. But, unlike the classic software, iPaaS also attracts midsize organizations and lines of business, due to its ease of access, versatility and low initial cost.

Business Impact

By rapidly and cost-effectively addressing integration needs, iPaaS enables organizations to improve efficiency, provide real-time business insights, increase business agility and introduce innovation faster. iPaaS adoption helps software engineering leaders achieve these goals cost-effectively, efficiently and with less costly skills than are needed for classic integration software. Also, iPaaS makes these benefits accessible to midsize organizations that cannot afford classic platform costs.

Drivers

- The vast iPaaS installed base and the COVID-19 pandemic notwithstanding, the iPaaS market grew quickly in 2020, driven by several business factors. These included organizations' pressing need to automate processes, accelerate digital transformation, react to the dramatic business changes forced by the pandemic, and speed up plans to move to the cloud in order to contain costs and increase agility.

- These factors were strongly at play among midsize organizations — traditionally, heavy users of iPaaS — at least in the less-COVID-19-impacted verticals. Domain-specific iPaaS targeting particular industries, SaaS ecosystems, business processes or geographies has been reasonably successful in this sector, because of its appeal to time-, skill- and resource-constrained organizations.

- The main goal of iPaaS providers now is to maximize opportunities to upsell and cross-sell to their vast installed base. Therefore, they are evolving their offerings into enterprise-class suites that address a wide range of hybrid, multicloud scenarios. Hence, large and global organizations now position iPaaS as a strategic option to complement, but increasingly also to replace, classic integration platform software, which drives more widespread adoption.

- A growing number of SaaS providers "embed" in their applications their own iPaaS, or one from a third party, which they typically extend with a rich portfolio of packaged integration processes (PIPs). This makes embedded iPaaS offerings attractive to organizations that need to quickly address SaaS application integration.

- Providers will keep investing to improve developers' productivity, reduce time to value and shorten the learning curve. The goal is to further expand their potential audience, to include business users. Hence, providers' R&D efforts will focus on using AI, machine learning and natural language processing to assist development and operation, enrich PIP portfolios, and enable CI/CD and DevOps to entice professional developers.

Obstacles

Obstacles to even faster iPaaS adoption include:

- The market's extreme fragmentation (over 150 providers and counting). This makes it hard for user organizations to select the best-fit iPaaS for their needs, could generate a proliferation of diverse, stand-alone and embedded iPaaS offerings, and risks fragmenting service providers' investments in skills building.

- The top five PaaS providers' command of about 60% of the market, and the fact that only seven providers have over 2% share. Hence, the vast majority of providers are of dubious viability, which may discourage the most risk-averse organizations from making strategic investments in iPaaS.
- The API rhetoric of seamless "plug and play" integration, the confusion among less technically savvy users about the differences between iPaaS, RPA and API management platforms, and the growing trend for code-based integration encouraged by OSS integration frameworks. These factors could reduce iPaaS's appeal, at least to large organizations.

User Recommendations

Despite the risks relating to market fragmentation, software engineering leaders responsible for integration should adopt iPaaS when looking for:

- An integration platform for midsize organizations moving to the cloud and for "greenfield" integration initiatives.
- A strategic complement to traditional integration platforms — increasingly in the context of hybrid integration platform (HIP) strategies — in order to empower a collaborative, democratized approach to integration.
- An enabler of self-service integration for ad hoc integrators (such as application developers and SaaS administrators) or citizen integrators.
- A platform to support well-defined, tactical integration projects with low budgets, severe time constraints, and informally defined and incrementally formulated requirements.
- A potential replacement for classic integration platforms that are obsolete or cannot support their changing requirements.

Sample Vendors

Boomi; IBM; Informatica; Jitterbit; Microsoft; MuleSoft; Oracle; SAP; TIBCO Software; Workato

Gartner Recommended Reading

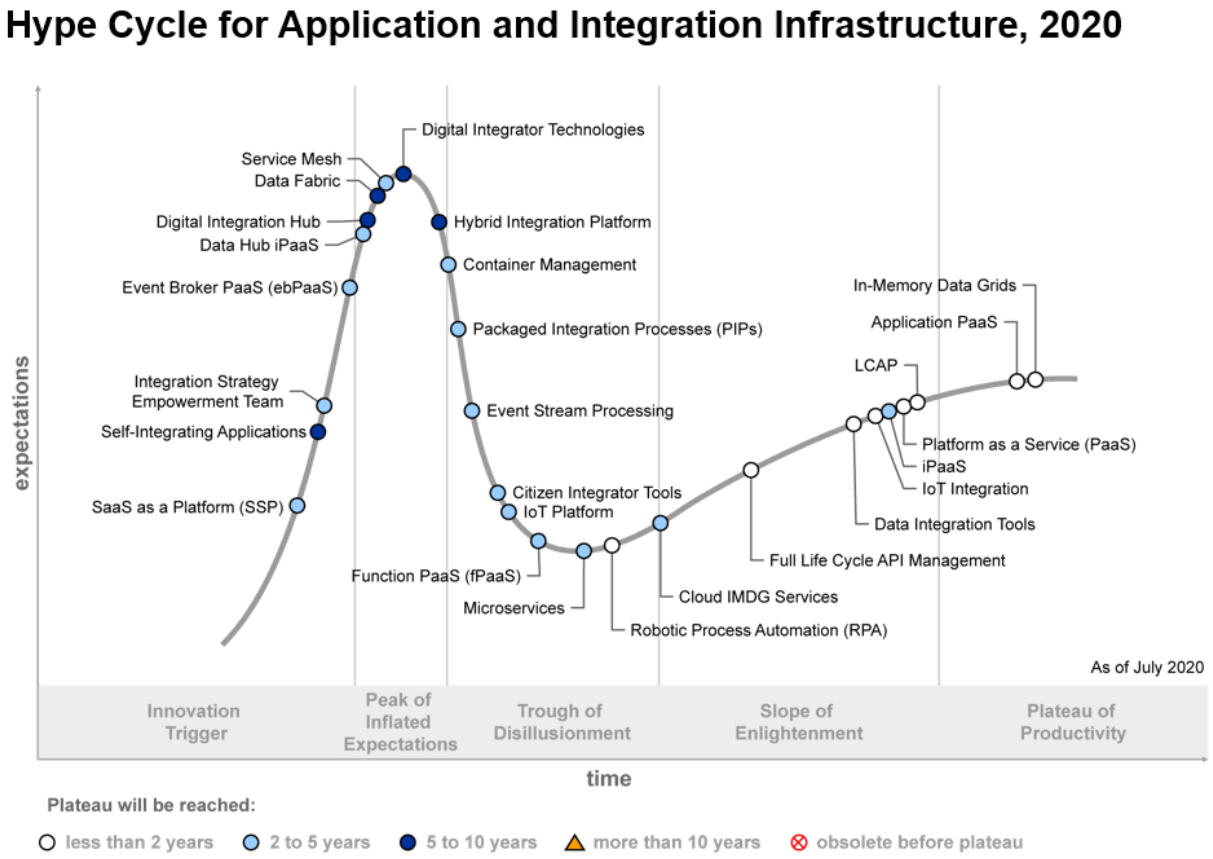
Magic Quadrant for Enterprise Integration Platform as a Service

Critical Capabilities for Enterprise Integration Platform as a Service

Choose the Best Integration Tool for Your Needs Based on the Three Basic Patterns of Integration

Appendixes

Figure 2. Hype Cycle for Application and Integration Infrastructure, 2020



Source: Gartner
ID: 450249

Gartner

Source: Gartner (July 2020)

Hype Cycle Phases, Benefit Ratings and Maturity Levels

Table 2: Hype Cycle Phases

(Enlarged table in Appendix)

Phase	Definition
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Source: Gartner (July 2021) (required)

Table 3: Benefit Ratings

Benefit Rating ↓	*Definition* ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2021) (required)

Table 4: Maturity Levels

(Enlarged table in Appendix)

Maturity Levels ↓	*Status* ↓	*Products/Vendors* ↓
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out-of-box methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constrains replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (July 2021) (required)

Document Revision History

Hype Cycle for Application and Integration Infrastructure, 2020 - 30 July 2020

Hype Cycle for Application and Integration Infrastructure, 2019 - 1 August 2019

Hype Cycle for Application and Integration Infrastructure, 2018 - 26 July 2018

Hype Cycle for Application Infrastructure and Integration, 2017 - 7 August 2017

Hype Cycle for Application Infrastructure, 2016 - 7 July 2016

Hype Cycle for Application Infrastructure, 2015 - 30 July 2015

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

Understanding Gartner's Hype Cycles

Create Your Own Hype Cycle With Gartner's Hype Cycle Builder

Hype Cycle for Software Engineering, 2021

Table 1: Priority Matrix for Application Architecture and Integration, 2021

Benefit ↓	*Years to Mainstream Adoption*			
	Less Than 2 Years ↓	***2 - 5 Years*** ↓	***5 - 10 Years*** ↓	***More Than 10 Years*** ↓
Transformational		Digital Integrator Technologies Event Stream Processing	Data Fabric Self-Integrating Applications	
High	Data Integration Tools Full Life Cycle API Management IoT Integration LCAP	Data Hub iPaaS Event Broker PaaS Integration Strategy Empowerment Team iPaaS MASA Microservices Packaged Integration Processes	Digital Integration Hub Event-Driven Architecture HIP	
Moderate		Citizen Integrator Tools Cloud Native Architecture Service Mesh		
Low		Serverless fPaaS		

Source: Gartner (July 2021)

Table 2: Hype Cycle Phases

Phase	*Definition*
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Phase ↓	Definition ↓

Source: Gartner (July 2021) (required)

Table 3: Benefit Ratings

Benefit Rating ↓	Definition ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2021) (required)

Table 4: Maturity Levels

Maturity Levels ↓	*Status* ↓	*Products/Vendors* ↓
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out-of-box methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constrains replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (July 2021) (required)