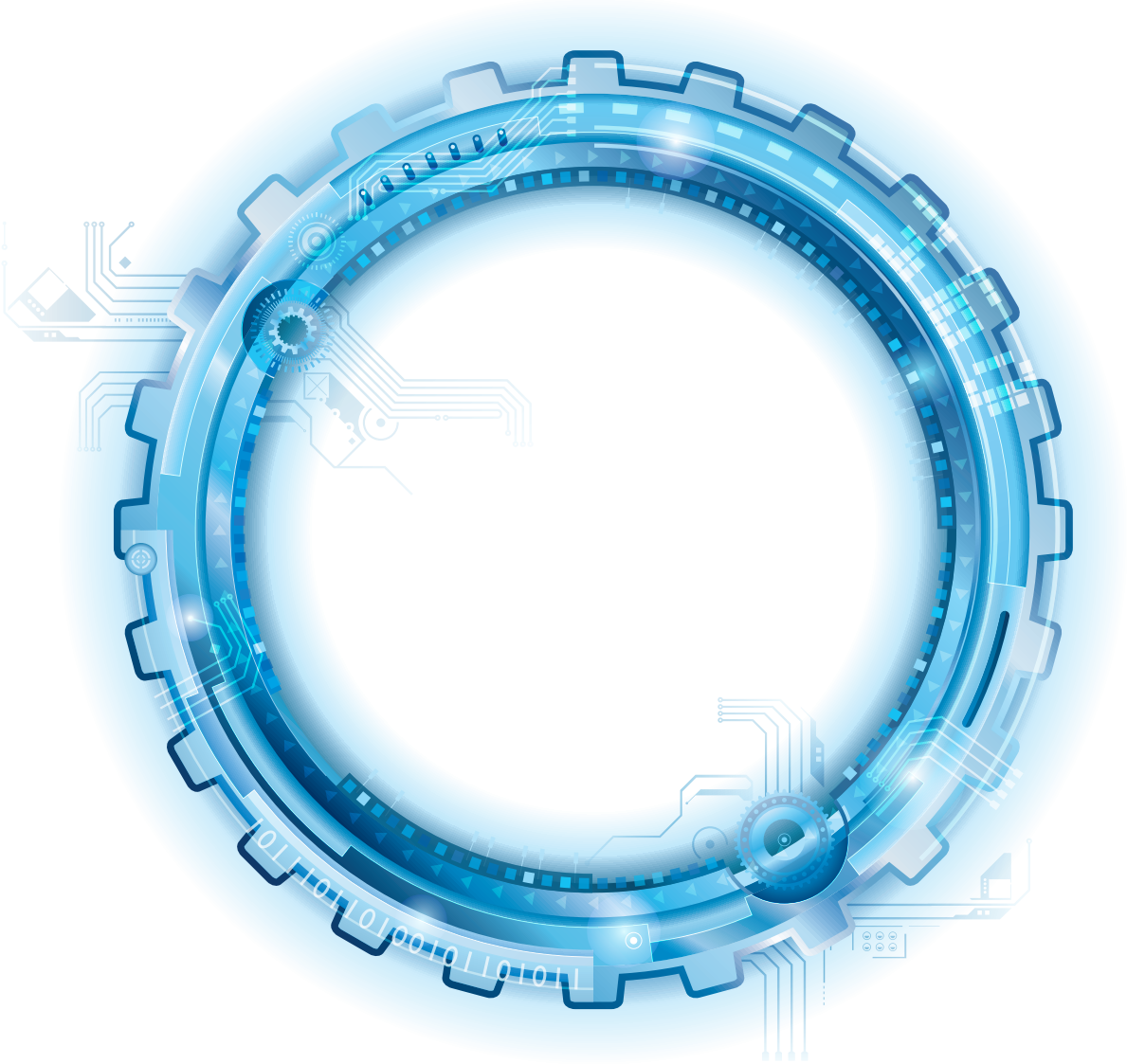
Deloitte.



Hybrid Cloud Integration

Navigate the journey
Unleash the synergy

January 2026

Table of contents

Executive Summary

Hybrid cloud architectures, integrating on-premise footprint with public cloud services, are becoming a pivotal element in any enterprise's cloud adoption journey, especially in the race towards digital sovereignty. Hybrid Cloud Integration requires an intentional approach, and this paper helps you understand why!

Hybrid cloud architectures enable organizations to extend their reach beyond traditional data centers and unlock the potential of cloud through integration.

Hybrid Cloud is more prevalent than ever, especially with the increasing need to be digitally sovereign. The quest for resilience leads to the pragmatic necessity to straddle different environments due to factors such as geopolitical tensions, regulatory compliance, data sensitivity, or the need for a phased transition.

This paper underscores the need for every enterprise taking the step towards hybrid cloud to employ a strategy on how to integrate and orchestrate, optimize and leverage the existing strengths of the cloud and its own on-premise assets to effectively navigate the complexities of contemporary operational landscapes and unleash the synergy between these resources.

The paper discusses practical considerations for different phases, recognizing well that a holistic transformation requires the seamless integration of business, development and operations functions.

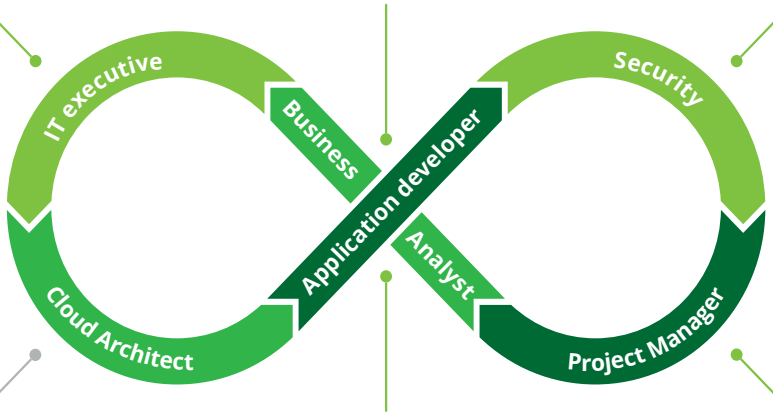
Drawing from lessons learned and challenges faced in a real life cloud modernization project, the paper offers valuable insights and guideposts for organizations embarking on a similar hybrid cloud journey.

Everyone is a stakeholder in the hybrid cloud integration journey. The paper is crafted towards a diverse audience, promising takeaways for the reader depending on the role played in a cloud transformation journey. This paper has in store for YOU, the

IT executive, shaping strategy: insights into the importance of hybrid cloud integration.

Application developer, creating cloud applications: API integration and modernization strategies

Security and Networking Professional: the role played in ensuring effective integration



Cloud Architect, designing and managing cloud: practical guidance on integration strategy and best practices

Business Analyst, involved in business process and requirement gathering for cloud integration: direct impact of process requirements on the quality of integration

Project Manager, overseeing a cloud integration project: tips to navigate challenges and ensure successful project delivery

The relevance of Hybrid Cloud

When Cloud First Isn't always your default strategy?

For the past few decades, a cloud first strategy has been the preferred choice for organizations when it comes to new business initiatives and IT applications. However, this may not always be the right fit for all organizations.

For instance, organizations that have invested significantly in on premises infrastructure or run legacy systems like traditional data center resources often regard these as critical, unique and indispensable to run their business operations; there is no compelling business case to replace such systems which continue to provide time tested functionalities.

Or, industries with stringent compliance requirements such as supply chain, production, or health care, often require applications to be hosted on-premise as this ensures legal compliance and local control on data.

Hence, decision factors to not opt for public cloud could be the cost of change, intellectual property, security, data sensitivity, regulatory compliance, usage patterns etc.

The Best of both worlds

Many organizations find that a hybrid approach, strategically leveraging both cloud and on-premises solutions, provides the flexibility needed to address a diverse set of needs.

A hybrid cloud, sometimes alternatively referred to as multi cloud, is the combination of public cloud(s), private cloud(s) and/or on-premise resources and/or edge computing.

It can be argued that Hybrid cloud offers the best of both worlds marrying the benefits of public cloud (bringing in agility, flexibility, scalability and elasticity benefits) with that of on-premise or private cloud (bringing in control and compliance).

In the race for digital sovereignty, hybrid cloud is more relevant than ever.

Digital sovereignty, a rising concern for organizations, is one of the main forces behind the push to hybrid cloud.

Organizations are increasingly focused on the ability to shift workloads away from global cloud providers to alternatives local regions, private deployments or regional suppliers that offer greater autonomy and insulation to data, technology and operations from geopolitical tensions and regulatory change.

This underscores the business case for the relevance of hybrid cloud.

The legacy factor

As organizations transition to the cloud, they aim to optimize their still valuable on-premise applications by adopting a continuous modernization approach.

On-premise applications are often equated with legacy systems: information systems that were state of the art at the time of implementation, but have since accrued technical debt (i.e., modernization opportunities in software, technology and hardware) that need to be addressed. These classic systems are invaluable to the business as they house critical data and provide unique functions essential to the business.

Eventually, when such systems start to hinder an organization's ability to address disruptions in an agile manner, the organization often turns to continuous modernization approaches.

This involves integrating these applications with newer technologies to leverage emerging cloud capabilities, modular design, and flexible deployment options or converting these applications into digital platforms, rather than resorting to a rip and replace approach which can be risky, costly and time consuming.

This approach allows organizations to strategically decide which IT elements or workloads to migrate to the cloud or retain on-premise, enabling more focus on innovation than on maintenance and operational tasks.

A balanced approach is essential evaluating each project individually, considering the specific requirements of the workloads (such as security, compliance, cost, and performance), and then arriving at a decision that best fits the unique needs of the project or workload.

As a result, snapshots of enterprise landscapes at any given time will typically reflect a hybrid cloud.

The essence of cloud

Cloud computing is characterized by the delivery of scalable and elastic IT enabled capabilities as a service using Internet technologies.

Cloud adoption should be seen as embracing new technologies and methodologies rather than simply migrating all workloads to a public cloud with the expectation of automatic scaling.

It's not about a default preference for the cloud but about leveraging the right combination of solutions and services to enhance efficiency, scalability, and overall business outcomes.

The decision to migrate to the cloud must be well informed, with a clear understanding of the benefits financial returns, innovation potential, business agility, lifecycle management and more. While cloud migration can deliver significant advantages, a cloud first strategy should be tailored to align with an organization's specific goals, priorities, existing infrastructure and regulatory constraints.

Implementing and managing a flexible IT environment in a hybrid cloud presents challenges:
interoperability, data and applications integration, security, change and risk management, to name a few.

Meticulous planning and collaboration between multiple teams and departments is required to achieve a seamless and efficient set up.

"Hybrid cloud at its peak has on-premises and public cloud workloads operating as one cohesive identity."

Analyst, Forrester

Hybrid Cloud in manufacturing

Hybrid Cloud is common in any enterprise landscape
How about we imagine a hybrid cloud landscape in manufacturing?

The advent of Industry 4.0 led to digitization of all manufacturing operations, automated data collection for real time visibility and better decision-making.

The start of manufacturing processes is the interconnection between product engineering and manufacturing execution. This involves exchange of the bill of materials (BOM) between PLM (Product Lifecycle Management) and ERP (Enterprise resource planning) systems.

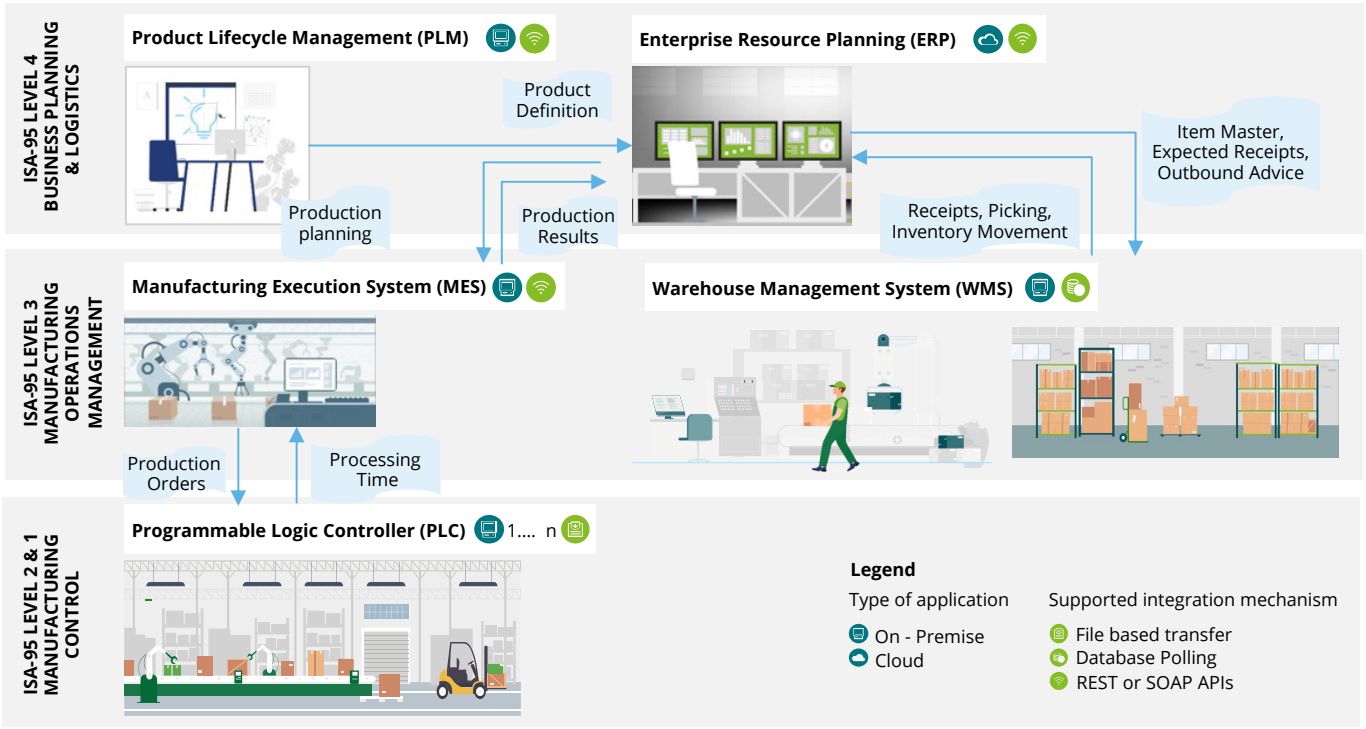
The BOM defines the components required to produce a product, which is an input to a production order in the ERP.

The concept of 'connected manufacturing', a key component of Industry 4.0, refers to top down production planning: connect and interchange information between ERP and MES (Manufacturing Execution System) to release production orders in real time, and further down from the MES to machine cells to schedule, manage and track each production step at machine control level. A warehouse management software (WMS) helps create visibility into inventory levels and manage material replenishment to the shop floor.

A glance at the supply chain and production shop floor systems of a manufacturing partner having global operations will reveal a fair share of on-premise applications across sites.

The **figure** below depicts a typical hybrid cloud landscape supporting manufacturing and the resulting information flow. The legend depicts possible variations for each application supported protocol, footprint (i.e., cloud or on-premise, which may manage multiple physical machines on the production floor.)

Figure 1. Information flow in a hybrid cloud for manufacturing



The illustration uses ANSI/ISA 95, an international standard, which provides a framework for integrating enterprise and control systems, outlines a model for managing manufacturing operations, divided into several levels with specific functions and responsibilities.

- Level 4: Business planning and logistics for enterprise resource management.
- Level 3: Manufacturing operations management to optimize production.
- Levels 2 and 1: Real time control systems and sensor/actuator control for managing physical processes.

Please note the information flows and applications are not an exhaustive list.

Hybrid Cloud Integration

The rapid evolution of business needs implies seamless integration of disparate applications and systems, diverse data and events, for various personas with distinct user journeys.

It is not a simplistic connection of applications anymore!

Integration is a linchpin in hybrid cloud
As evident from the previous section, planning, control and execution of manufacturing processes can be achieved only if systems (on-premise and cloud applications) are interconnected and able to access data-bi-directionally in an automated manner through integration.

Modernization approaches leads to microservices, APIs and integration
Most innovative companies support business transformation by decomposing their applications into specific processes and capabilities, taking the essential transformative step to move from monoliths to modern microservice architecture. They then invest in technologies such as APIs to interconnect these modular microservices (encapsulating the forementioned processes and capabilities) across applications in the landscape.

An API (Application Programming Interface) is a set of protocols that allow applications to communicate with each other. An API enables integration by making assets in the application available, thus, extending the application's features and values.

Application modernization involves a three pronged approach focusing on technology, architecture and functionality to improve application components. Some organizations may choose to adopt a legacy application modernization approach, migrating from a tenured (end-of-life) on-premise version to the latest cloud version.

Integration isn't straightforward in hybrid cloud landscape
The key to successful hybrid cloud integration is a well thought out strategy that considers the unique needs and challenges of both on-premises and cloud environments.

This paper discusses considerations to be accounted for while planning, architecting and executing Integration & APIs in a Hybrid cloud landscape. These insights are derived from challenges faced and lessons learned from implementing integrations in a complex hybrid cloud/IT environment in a real life cloud transformation project.

The Enterprise Integration Platform

Select the integration tool based on current and future use cases and capabilities

It is imperative to prioritize identification of an organization's processes and key use cases and thereby determine the desired capabilities of the enterprise integration platform.

On-premise applications may be databases, mainframes or custom inhouse built applications each with its own mechanisms (and limitations) of integration. Most of these classic systems, built and designed for a different era, typically provide asynchronous, queued/batched data exchange capabilities to meet traditional needs such as file transfers, database CRUD (Create, Read, Update, Delete) operations, stored procedure invocations, advanced queuing etc.

Integration is becoming complex

With the proliferation of digital channels and entry points into the landscape, and emergence of new and evolving integration personas, integrations in hybrid multi cloud environments are becoming increasingly complex.

Thereby, the range of capabilities expected of an integration platform in a modern hybrid cloud enterprise landscape can be very broad, as depicted in the **figure** below, reflecting the need to support future use cases and emerging trends.

Balancing versatility over Best-of-breed

An integration platform needs to be versatile to address traditional use cases while also providing support for modern interaction modes (e.g., API Management and Gateways).

Though most enterprise integration platforms provide support for this broad range, some may still be evolving in certain capabilities while being strong in others (e.g., Managed File Transfer and Business Business integration for EDI transactions.)

While it is ideal to have a single enterprise integration platform from an operational and management perspective, additional integration products may be required to complement an existing platform or for best-of-breed capabilities addressing niche use cases.

Be prepared to invest in multiple cloud integration products from different vendors to build up a future ready enterprise integration platform.

Figure 2. Broad range of integration capabilities

Message oriented middleware
Messaging framework supporting asynchronous communication (e.g., Rabbit MQ, Apache Kafka, IBM MQ etc.)

API management and gateway
Processes and tools enabling creation, deployment, monitoring and maintenance of APIs, ensuring secure and efficient access to backend services by consumers.

Business—Business
Electronic exchange of data and documents using standard formats between business entities thereby reducing operational inefficiencies

Event management
Support Event Driven Architecture through detection, correlation and monitoring of events in integrated systems and applications.

Managed file transfer
Seamless and controlled file transfer processes, addressing challenges of security, compliance & data integrity associated with traditional methods.

Service bus
Central communication hub, enabling protocol transformation, message routing in interactions between disparate decoupled systems

Application connectivity
Methods and technologies to support communication between different applications (e.g., connectors, protocols, message formats)

Process matters

APIs are “table stakes” technology .....foundational, because they link together the individual systems that contain data about people, businesses and things to derive value from that data.

Analyst, Gartner

An Integration and API strategy is not a technical exercise alone

APIs are key drivers of innovation and transformation, often serving as the front end of entire business models. APIs enable businesses to connect with customers, streamline business operations, and bring new offerings to the market.

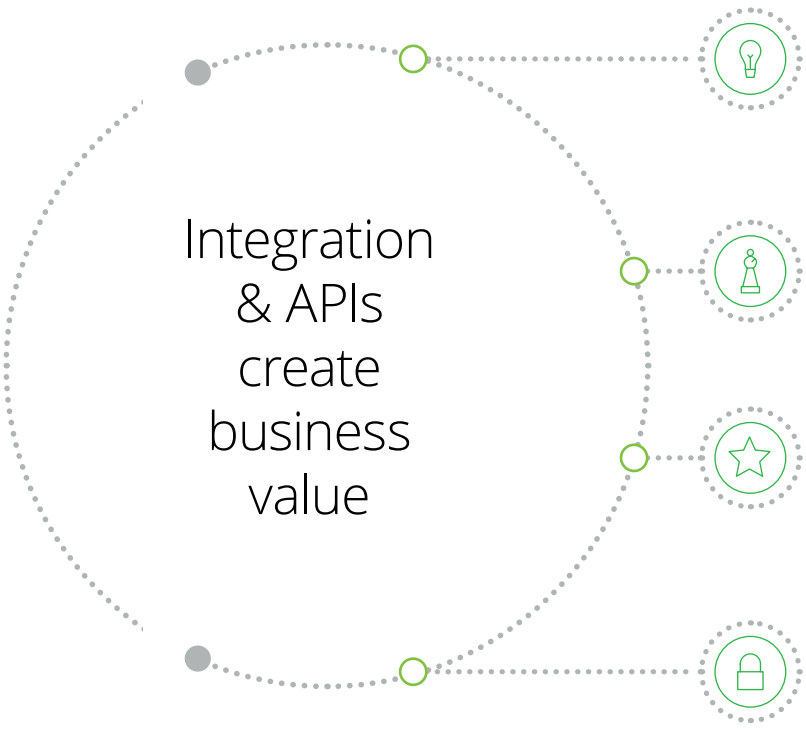
The ease in creating APIs and the direct relation with business goals made APIs very popular.

The evolution of the API first or API led connectivity approach is rooted in the vision that internally or externally shared APIs shouldn’t be treated merely as technical components, but as a product relevant in the context of a business.

A KPI for APIs is their adoption metric, determined by the number of consumers, which is made possible through integration.

A well defined integration strategy focused on business enablement and continuous improvement is critical for an organization to realize business value, as depicted in the **figure**.

Figure 3. Business value created by integration & APIs



Accelerate innovation

Unlock assets from diverse sources to create a cohesive narrative of the bigger picture from the integrated data. Real time access to market trends, customer preferences, and industry shifts augments decision making and is a goldmine for innovation

Enable business agility

APIs enable modernization of legacy systems; Integrations enable these systems to co exist and connect to modern platforms in the enterprise landscape. This is a strategic step to enabling organizations to be more agile to business disruptions

Automate business processes

Streamlining processes through application integration reduces manual intervention and cost, thereby leading to efficiencies. Through B2B integration, internal systems and partner ecosystems can effectively interchange data, hence extending reach of operations

Enhance experience

Integrating various applications, channels and endpoints in a seamless manner is a key differentiator in providing superior experience to consumers

Integration requirements need to be defined in the context of the larger process design

Today, products provide Web based graphical environments for designing APIs, which enable consumers to be more involved in defining the way of working, through every step of the API lifecycle (definition, development, operations) fostering involvement and ownership from business.

Process or workstream owners need to define a clear business process and articulate why, which, and when data exchange is required. Their insights into existing workflows and business processes are invaluable for defining integration needs.

This process flow, in turn, becomes an input to subsequent integration deliverables. The figure below illustrates how an integration solution design is informed and shaped by business process flow and requirements.

When integration requirements are grounded in the relevance and depth of a related business process, this results in a more robust design adhering to both functional requirements and non functional SLAs. (A cautionary note: Introducing integration will not fix the issues caused by inefficient processes.)

Figure 4. How process influences integration design

Process flow
Process description <Describe current state process, future state requirements and need for change. Include a process flow diagram. Address the following: • Scope of the process: where does the process begin and conclude? What is the boundary of the process? • Individual tasks or activities that are performed, in sequence. Specify the inputs and outputs for each. Methods and channels of communication
Business requirements <Reference from traceability matrix, User Story, Feature>
ENTITY/BUSINESS UNIT <Business department responsible for the process>
Personas <Users/personas involved, and their Roles and responsibilities. Also include estimated number of users expected to utilize this process>
Data entity <Data entity—inputs and outputs to the process, entity data relationship mapping>
Business process trigger <Any triggers for the process could be any action that impacts relevant entities triggering the process>
Dependencies <Any processes that need to be complete before this process can execute, any system dependencies, operational aspects>
Assumptions <Process assumptions>
Criticality <Business criticality>

Integration solution design
Interface requirement <In the process, specify the interfacing requirement(s). When do the applications need to communicate. E.g., A create/update on same entity needs to be called out>
OPERATIONS <Data entity operations>
PROCESS TRIGGER <Data triggers—any specific data entity/system criteria to kick off interface>
DATA SELECTION/SORTING/FILTERING <Selection Criteria>
VOLUME <Volume in current state. Also, include any known estimations for future state>
FREQUENCY <Frequency requirements in future state: near real time, scheduled etc.
AVAILABILITY <Business availability for interface/acceptable downtime, action to take if interface doesn't run for any duration, eventual consistency between systems>
EXCEPTION PROCESSING <Any specific business errors>
NOTIFICATION <Any user/groups that need to be notified for specific conditions for business purposes>

Involvement through the lifecycle

The involvement of process/workstream owners in the API or integration lifecycle extends beyond the initial stages. They need to take responsibility to include integration in the agenda of process workshops, provide inputs for i nventory of required integrations, create process definitions, contribute to mappings, identify business entities, review and provide feedback on API contracts, identify test scenarios with test leads, ensure availability of configured test environments with data, and so on.

This requires a mandate from program management so process/ workstream owners can allocate the necessary time and assume ownership for these activities. Consequently, a shift in mindset is required all through the organization.

Integration and API strategy needs to be embedded in the organization's digital culture and aligned with business objectives, helping the organization pivot towards an innovation path.

The **table** below outlines activities where workstream/process owners are expected to be involved or take ownership through the project lifecycle, ensuring their engagement in shaping an operational API.

Table 1. Activities & deliverables by phase where process owners need to be involved

Phase	Process/workstream owners	Integration team	Deliverables
Discovery	• Run workshops • Finalize process design • Map as is to to be functionality	• Assess as is and to be landscape • Define integration strategy and touchpoints • Determine integration design patterns	• Defined epics, features and user stories • Integration Strategy • Integration touchpoints and data flow diagrams
Design & Build	• Provide layouts or definitions (file, database, etc.) of source and target applications, mapping rules from as is mapping for integration design • Review API specifications and sign-off integration solution design	• Define common data model • Define API specification • Build Integration Solution design with Mappings • Detailed design and documentation of prioritized functional and technical user stories • Iteratively prioritize user stories and define product backlog • Build sprints and testing (Unit)	• Built and tested, demoed and signed off functional and technical user stories (including integration interfaces and APIs) • Refined and prioritized sprint/ product backlog • Test results
Testing	• Workstream/process owners are involved in identification of scenarios and related test data. • Application leads ensure application readiness i.e., prepare test systems with data and configurations • Provide sign off on scenarios	• Provide testing support and defect fix	• Signed off test reports
Deployment	• Prepare production systems with configurations and data	• Perform cutover/deployment steps	• Deployment plan • Production environment ready for go live

Formulating an Integration & API Strategy

APIs and integration are, at times, perceived as alternative approaches. Nothing could be farther from the truth.

Analyst, Gartner

Proprietary standards and protocols

Modern cloud platforms adhere to the best, proven and current practices for business integration requirements; however, not all on-premise applications can claim the same.

While APIs are a popular way to enable integration with legacy applications, the APIs available in classic systems use proprietary formats and protocols, requiring API mediation layers, wrapper approached. For e.g., APIs which support only SOAP protocol (a time tested protocol for APIs), and not more modern API architectural styles (REST, GraphQL etc.). Therefore, an integration tool is required for complex mediation scenarios.

Traditionally, an ESB (Enterprise Service Bus) was used; functioning as a transit system to mediate complexities associated with disparate protocols, data formats, and communication patterns.

Nowadays, an iPaaS (Integration Platform as a Service), the next generation integration platform with pre built robust connectors and templates, provides secure methods of connectivity between on-premise and cloud applications.

Integrations harness the full potential of APIs

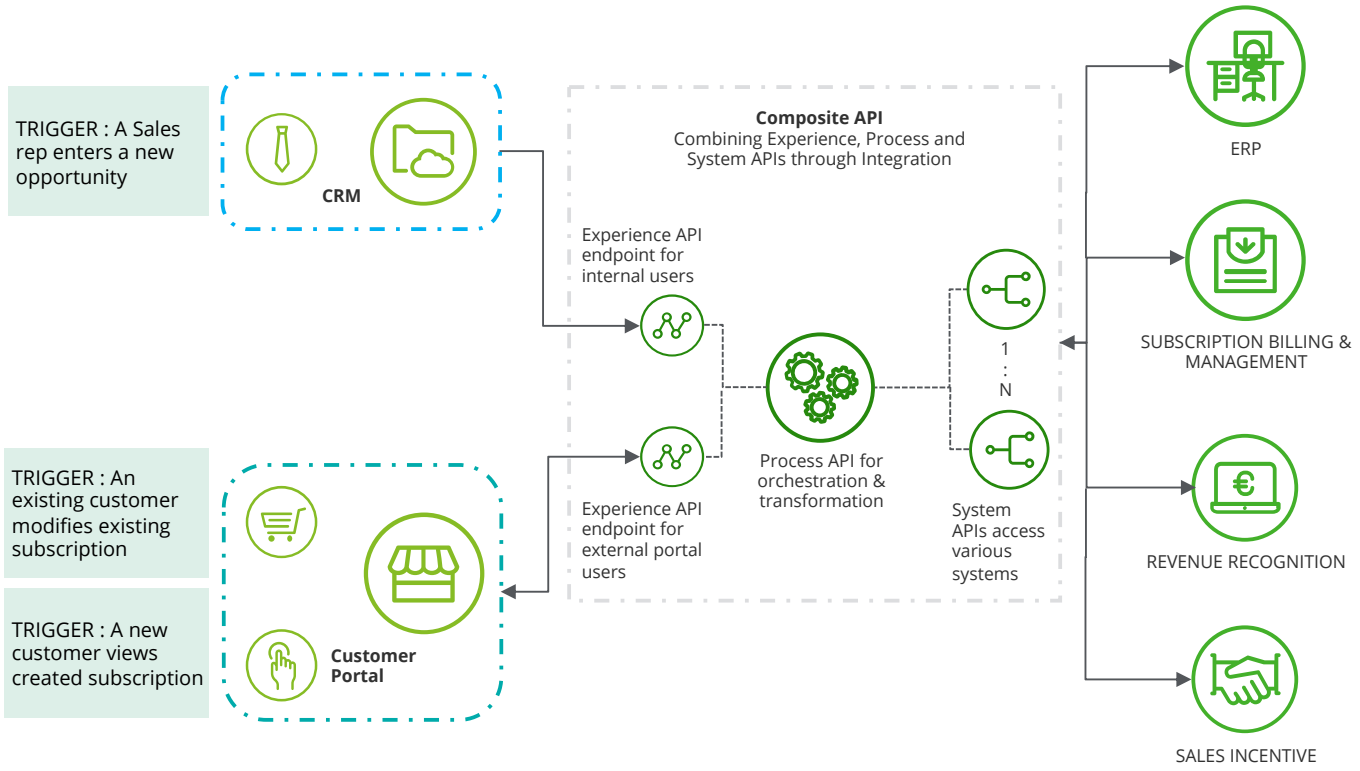
While APIs provide the means for different systems to communicate, integration is the process of making those communications effective, reliable and seamless through orchestration, transformation and mediation.

As seen in previous sections, an API based approach is just one of the many ways to deliver integrations. Integration technology is required to connect disparate applications to combine capabilities and exchange data through different means, which may not always be through APIs.

Therefore, an Integration and API strategy needs to account for how on-premise applications can seamlessly integrate and interoperate with Cloud services and APIs.

This is one of the challenges faced in integration between on-premise applications and the cloud.

Figure 5. How individual APIs are connected using integration techniques to create a composite API enabling flexible consumption business model



Composite APIs

While APIs enhance reusability and connectivity at the application level due to their modular design, it is the integration tools and platforms that unlock the full potential of APIs by combining them to create Composite APIs which generate meaningful business value at the enterprise level.

Integration tools enable data synchronization, automation, and streamlining of business processes, handle complex data transformations, aggregate data to derive insights, and coordinate capabilities across applications. Integration transforms individual APIs into a cohesive and interoperable enterprise architecture, unlocking the full potential of digital transformation.

Not a one-size fits all

The popular integration architecture with three API layers is based on the API led connectivity approach: The cornerstone of the API led connectivity approach:

- System API: Unlocks data from Systems of Record.
- Process API: Orchestrates multiple system APIs for a specific business purpose.
- Experience API: Exposes unlocked assets to end users or consumer applications.

While a three layer API architecture may seem appealing for all integration needs, it may not always be suitable, especially in a hybrid cloud integration involving legacy systems.

Application strategy consideration

It is critical to consider the application strategy of the specific on premise classic system when selecting an integration pattern. If the application is expected to sunset or reach end-of-life soon, investing in a three layer API approach might not be financially prudent. A two layer API approach comprising (only System API and Experience API) could suffice in such cases due to anticipated low reusability.

Quality-of-service factors

Ensuring quality-of-service (e.g., availability, reliability, scalability, latency, throughput) is challenging in a hybrid cloud environment. Reliability aspects, such as queuing and reconnection strategies, must be carefully integrated into the design, considering potential availability issues with on-premise applications due to hardware dependencies, limited scalability, and maintenance windows, which may affect non functional SLAs.

All hands onboard with Security & Networking!

Addressing security risks in a hybrid landscape is an imperative

Digital transformation in the modern world integrates diverse technologies such as analytics, cloud computing, IoT, and DevOps, leading to a hybrid cloud architecture that combines public cloud, private cloud, and on-premise applications.

This setup exposes digital assets (resources and data) to multiple entry points, facilitating access and exchange between internal data centers or legacy systems and external clouds or the edge, thereby creating security risks.

Ensuring security is a continuous process

Organizational security departments must evaluate and approve suitable technologies and protocols for APIs, focusing on areas such as Identity and Access Management, Data Encryption, Application, API and Network Security, Runtime/Threat Protection, and regulatory compliance.

A "secure by design" approach is essential, integrating security considerations into the planning and development phases, rather than as an afterthought.

Networking is the underpin of cloud computing efficiency

In a hybrid cloud/IT strategy, it's crucial for an organization's networking team to participate in decisions about connecting internal networks with the cloud environment (e.g., VPN tunnels, load balancing), implementing related security measures (e.g., DMZ, firewalls, IP allow listing), and determining bandwidth and sizing.

Networking choices affect key non functional SLAs (such as latency, reliability, and high availability), directly impacting the end user experience. From the end user perspective, a consistent experience should be ensured, whether accessing resources on-premises, in the cloud, or at the edge.

Active involvement of Security and Networking teams in digital transformation initiatives

Digital IT departments often set priorities, secure budgets, and assemble teams for digital transformation initiatives. However, successful delivery of an operational integration requires collaboration with the organization's Security and Networking teams. Aligning IT initiatives' priorities is essential to ensure these teams are involved, enabling a focus on risk management and mitigation within Digital IT projects.

Quality-The gatekeeper

Advance Planning for Testing Cycles in a Hybrid Cloud/IT Environment

Testing cycles (such as integration, system integration, user acceptance, performance, end to end) must be planned well in advance.

Incremental integration testing needs to be adopted for rapid feedback and fail fast approach.

To ensure thorough testing of various scenarios and variations, process teams should guide test teams on identification and prioritization of test scenarios. Collaboration among development, testing, business, process and operations teams is essential for successful testing outcomes.

Extrapolated testing effort

Testing on-premise applications poses challenges, as installations may span multiple machines or servers across factory sites.

For example, production control systems that automate and organize processes might have separate physical machines managing specific production orders. These are controlled by machining cells, which need integration with other systems through an iPaaS. The testing effort increases with the number of machining cells in the landscape, and this needs to be factored into the planning of testing cycles.

Early instance mapping exercise

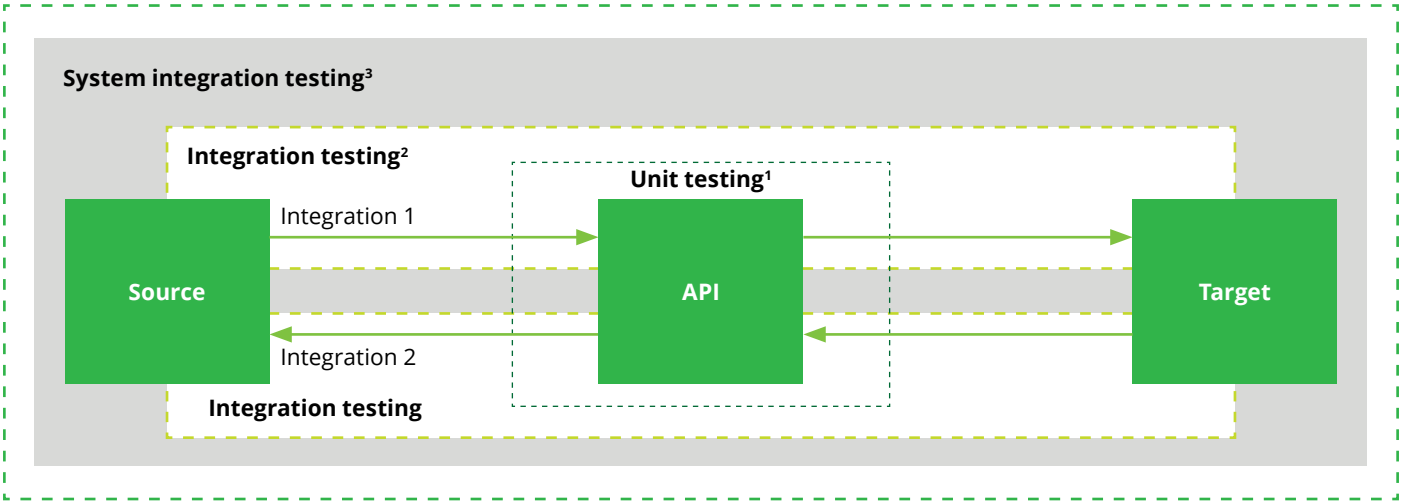
In modern multi environment architectures, it is a common practice to provision four DTAP instances: development (D), testing (T), acceptance (A), and production (P). Non production instances need to replicate the production environment closely, including configurations across machines, servers, and sites.

It is possible all four environment instances are not maintained for on-premise classic systems due to their stable, business as usual state. Typically, only two instances (test and production) will be available.

Detecting this gap during environment instance mapping exercise is crucial, as it serves as an essential entry criterion for any test cycle.

It's important to provision test servers in advance if possible as these involve long lead times, or plan to load data sets on available servers to support testing cycles for specific sites or simulate instance specific landing zones.

Figure 6. Test cycles



1. Integration Developers perform unit testing of individual integration through simulation
2. Developers of involved teams test together to validate technical aspects of individual integrations
3. A planned test cycle with defined entry and exit criteria. Lead by testing team; includes test scripts, scenarios, test data, test reporting, defect triaging, test environment mapping at a minimum.

Accounting for multiple distant sites

The impact of hybrid cloud/IT projects can span multiple factory sites, even across oceans
With numerous on-premise system installations in enterprise landscapes, integration efforts are often underestimated, especially in terms of establishing connectivity and conducting tests.

Regional or local insights may not be ignored
Involving representatives from different sites, such as process owners, application and line managers, at the project's outset ensures alignment on the development and testing impacts. This collaboration minimizes unexpected issues by incorporating site or locale specific insights and addressing unique requirements, leading to a more realistic project timeline. Engaging process owners ensures they fully understand how integrations will affect daily operations.

Enterprise landscapes may vary regionally. For example, a car manufacturer with global operations may use a different system for production schedules in North America versus EMEA region. Enterprise architecture and data quality teams must anticipate and evaluate these differences early on. Data quality is paramount, underscoring the importance of accurate and reliable information.

> For all flavors and modes of hybrid cloud, management isn't easy. You've got different technologies, different responsibility mapping, and potentially hundreds of miles between locations
>
> **Analyst, Forrester**

Observability is key

Robust monitoring practice needs to be implemented
It can be difficult to establish visibility into all systems, applications, platforms, and processes contained in a hybrid Cloud/IT landscape.

Monitoring is an opportunity to gain deeper insights into the landscape by provide end to end visibility into the entire landscape.

Hence, all applications need to integrate logs into a common monitoring tool. User training, creation of user friendly dashboards, presence of a feedback loop are also important.

Monitoring requirements can be manifold-performance monitoring by tracking KPIs, real time error tracking and alerting, uptime and availability monitoring, concurrency, log analysis, usage analytics, proactive error incident management etc.

The true value of observability is realized when users of operational applications embrace and leverage tools and practices effectively
By encouraging a culture of engagement and empowerment around observability tools, organizations can capitalize on insights gained to optimize operations, enhance performance, and prevent incidents proactively.

Agile Integration project delivery: To be or not to be

Collaborative planning

As discussed in previous sections, Integration projects require the collaboration of cross domain teams in an organization.

The goal for an integration manager/lead should be to ensure alignment on project outcomes and objectives, priorities, timelines, resource availability, budget, and environments.

Identifying dependencies with other teams is crucial, and proactive strategies need to be employed to address them at an early stage. For e.g., ensure involvement of key stakeholders from diverse teams (application owners, security and networking, third party vendors, and source and target application teams etc.) in planning, provision documentation such as process descriptions, address availability of environments for testing with application owners, validate data, security and network configurations.

Ensuring alignment on operational, business, and security requirements among stakeholders is crucial. These are documented in the Integration Solution Design deliverable.

Continuous communication

Open and transparent communication is essential. The integration lead/manager should establish a clear working methodology, assisting other teams in understanding the intricacies of integration delivery and achieving early alignment on key milestones. It is crucial to ensure clarity regarding timelines and constraints faced by other teams.

The debate continues: Can integration projects be delivered in agile methodology?

Navigating the complexities of delivering integration projects using agile methods can be challenging due to multiple interdependencies among various teams throughout all phases, and the extended timeframe required to deliver an operational integration that meets the 'definition of done'.

Achieving a fully functional and operational API, whether System, Process, or Experience, often requires numerous iterations before successfully integrating multiple API layers end to end.

Despite these challenges, establishing a structured approach is crucial.

Sprint sized tasks for incremental value

Breakdown the effort involved in creating an API or integration to sprint sized tasks:

i. Establish secure network connectivity between the iPaaS (such as a Virtual Private Cloud, VPC, where integration processes and APIs are hosted) and other systems, which is required for System APIs

ii. Develop System APIs to integrate with systems of records

iii. Create and share an API specification/contract for review, which is required for Experience APIs

iv. Develop the Experience API

v. Develop Process API to build orchestration logic to link all applicable APIs

vi. Perform Testing cycles (integration, SIT, UAT, performance etc.)

vii. Deploy a fully operational API or integration

Each step builds on the previous one, following an iterative approach. This method delivers value in small, manageable increments, enabling continuous testing and validation.

Sequence identified tasks in appropriate sprints

Once the tasks are identified, organize and schedule them within suitable sprints. Activities with extended lead times or dependencies on other teams should be planned several sprints ahead of the actual development of the interface or API definition.

For example, consider Step (i). In a hybrid IT architecture, to approve connectivity patterns and establish secure connections between on-premise legacy applications and a cloud integration platform requires agreement, approvals, and involvement from the organization's IT and networking departments. This process, involves activitie s with long lead times (e.g., configurations related to firewall rules, ports and VPNs). Some integration tools need this connectivity to download schemas, essential for Step (ii). Therefore, Step (i) is a critical prerequisite for starting development and must be timed accordingly.

Agreement on milestones for project risk mitigation

A quick win is how data will be exposed to the consumer, a prerequisite for building the Experience API. Step (iii) involves consumers (such as process owners and source application members) reviewing the specifications and approving Step (iv), reducing change costs if issues are found later.

Parallel Development

During Step (iv), the Experience API is developed without complete orchestration or processing logic. The actual orchestration logic to achieve the desired functionality can be created in later sprints during Step (v), in parallel with source/target application teams. This requires effective coordination, significantly accelerating progress.

Operational integration

Once the API or integration process has successfully completed multiple quality assurance cycles in Step (vi), it is considered stable and reliable for deployment to a production environment in Step (vii), where it handles real data and live user interactions.

It takes two to tango!

Success in integration projects requires collaboration. Embracing an agile mindset enhances teamwork, promotes transparency, and delivers value incrementally. It's essential to apply expertise in achieving "operational" integration to develop an agile delivery approach. This combination ensures a solid beginning and achieves early wins throughout the project's extended timeline.

Conclusion

In conclusion, the journey into the realm of hybrid cloud integration is a strategic maneuver that defines an organization's ability to thrive and be resilient amidst constant disruptions, which is the crux of an organization's need to be digitally sovereign.

This paper attempts a comprehensive exploration of the complex world of hybrid cloud integration; navigating through the why, what, and how, providing actionable and informative insights for specific stakeholders in businesses venturing into this world.

In essence, Hybrid Cloud is an ongoing journey, constantly creating synergy in an ever evolving digital landscape.

References

Deloitte_ERP_Industrie 4 0_Whitepaper

A fresh approach to hybrid cloud adoption | Deloitte US

Factors leading to the rise in hybrid cloud adoption

Hybrid Cloud Is Hard To Manage

Modernize IT Infrastructure in a Hybrid World

Take A Pragmatic Approach To Embrace A Cloud Native First Strategy

The Top 10 Cloud Myths

Executive Essentials: Deliver Applications

Use Continuous Modernization to Optimize Legacy Applications

Use Continuous Modernization to Build Digital Platforms From Legacy Applications

How to Justify Strategic Investments in Integration Technology

How to Achieve Digital Business Excellence by Mastering the Pervasiveness of Integration

How to Justify Strategic Investments in Integration Technology

Ensure Your Integration Strategy Supports Modern Integration Trends

Hype Cycle for APIs, 2023

https://en.wikipedia.org/wiki/ANSI/ISA 95

Top 5 Trends Shaping the Future of Hybrid Cloud

The author

Jita Susan Jacob is a Senior Specialist Lead in Technology and Transformation. She enables hybrid cloud transformation for diverse clients by combining her specialization in integration and architecture, and experience in delivery with global teams.

jitjacob@deloitte.nl
Deloitte Consultative Services B.V.

Acknowledgements

I sincerely appreciate the time and effort spent in detailed review, constructive suggestions and thoughtful insights which helped in refining the content and enhancing clarity.

Bart Pustjens
Partner
Technology and Transformation

Helena Lisachuk
Partner
Technology and Transformation

Rob Dickson
Senior Manager
Technology and Transformation

I would like to thank **Yassin Ben Allal** (Analyst, Technology and Transformation) for the Dutch version of this web-page and **Ting Chen** (Analyst, Technology and Transformation) for initial translation help.

Deloitte.

Designed by CoRe Creative Services. RITM2350113