



Hybrid Integration Platform

Buyer's Guide 2021
Checklist



Set-up	
	Be a SaaS Platform or SaaS managed
	Enables deployment to my Azure contract
	Enables deployment to my GCP contract
	Enables deployment to my AWS contract
	The platform should not need external resources such as a Database or Virtualizers beyond what is a part of it
Operation	
	The platform must be fully operated by the supplier
	The integration platform must be delivered in 3 zones
	Has its own monitoring
	Have a dashboard with a view of execution by integration
	Supports external monitoring
	Offer a real-time status page that allows monitoring the availability of platform services and allows automatic notification of customers in the event of incidents
Solution Architecture	
	The platform must have processing isolation, so that an arbitrarily large or malformed request does not affect any other integration and does not affect the stability of the platform.
	The platform should be built with the premises of Cloud Native applications, based on the container orchestrator Kubernetes.
	The platform must have an automatic restart mechanism for integration in case of failure without loss of the message to be processed.
	The platform should follow low-code concepts, not requiring a build to publish integrations
	The platform should natively support all its features for creating and publishing integrations, without the need for customization or hiring additional solutions.
	The platform should define a standard message model (for example JSON), with automatic conversion regardless of the source of the information.
	The platform should control the flow of transactions automatically, queuing those that cannot be processed immediately.
	The platform must have an integrated API gateway without compositing needs with external solutions even from the same manufacturer

Event-driven & streaming	
	The platform should support the publication and subscription of events natively, without the need for additional components.
	The platform should support the streaming architecture, allowing data consumption in parallel and with low memory usage.
	The platform must support the consumption of rows of queries to the database in streaming mode.
	The platform should support the consumption of JSON structures in streaming mode.
	The platform should support the consumption of file content in streaming mode.
	The platform should allow the construction of decoupled flows, with flow control and error handling.
Integration Designer	
	It should allow the construction of integration flows in a simple, intuitive way and based on drag and drop of components
	It should be possible to visually build flows with structures based on decisions, flow deviations, iteration, loop and error handling.
	The interface must be 100% web, eliminating the need to install clients, IDEs or plugins.
	The platform should allow the creation of integration models or templates, which can be saved and reused to promote consistency and uniformity in the integrations.
	It must support expression language for easy replacement of values in the connectors used in the flow
	The expression language must support mathematical functions, manipulation of strings and dates, logic and conversion between types.
	It should be possible to visually define parameters for API settings such as Method, Path and Security Plugins
Integration Runtime	
	The platform must support a non-production and production environment
	The integrations created on the platform, when deployed, should already have embedded and enabled monitoring, without the need for any configuration.
	The platform should natively create, manage and maintain the API that the integration needs for external access in a secure way by activating plugins with JWT Token, apikey and security access scope.
	The platform should support the integration of the On-premises environment through site-to-site VPN, creating an Isolated cloud environment.

Governance	
	The platform should offer native and embedded version control of the integrations.
	The platform should handle the exposure of sensitive fields in the monitoring logs
	The platform should provide a mechanism for segregating integrations into projects (or other types of organizational unit) for better governance of company teams
	The platform should provide a safe for storing credentials that will later be used by integrations.
	The platform should provide usage policies for building integrations, such as which security mechanisms are mandatory, which sensitive fields should always be present, name resolution policies, among others.
	Maintain architecture related to documented integrations
	It should allow the management of API Consumers through API keys to control access to an integration or to a group of integrations
Interface	
	The platform must provide an interface of administration administration
	The platform must support multiple languages and locales
	User interface should be based on minimalist design principles, focusing on easy and intuitive use. Tools that refer to development IDEs and that require a very sharp learning curve will not be accepted.
Security	
	Data can be encrypted
	Support mutual authentication with business partners
	Support integrated authentication with client's Identity Provider (SAML v2)
	Have internal safe for storing passwords, certificates and tokens
	Data can be masked in logs
	Access to the platform must be based on profiles
	Should support authentication with the active directory
	Must support dual authentication factor in the administrative interface (MFA)
	It should allow auditing of all user activities
	Must have the concept of ACL for functionalities
	Must support VPN for information traffic
	Storage of data at rest must be encrypted
	In the managed SaaS model, all security is the responsibility of the platform vendor. Example: Management of firewall, network policies and security patches
	Perform periodic pentests
	The Platform must have anti DDOS protection
	Must have continuous component hardening process

Connectors	
	Support enterprise applications such as SAP, Totvs, Salesforce, Oracle EBS
	Support symmetric and asymmetric encryption, including RSA, AES, PGP, CMS
	Support the generation and validation of cryptographic hashes
	Enable intuitive and assisted data transformation
	Support integration with SOAP-based Web Services endpoints, allowing configuration of the SOAP envelope as well as header and authentication details
	Support integrations with REST solutions (API) in order to adopt all the necessary technical standards such as timeout control, compressed data, types of content, error handling and security standards such as authentications using oauth2, tokens, service accounts, certificates, basic authentication, custom headers
	Flexible trigger conditions: <ul style="list-style-type: none"> - REST - HTTP - Event - JMS, RabbitMQ, Kafka and SQS - HTTP with support for sending large files - Scheduled execution - Email
	Native consumption and publication of messages in queues
	Support file storage in a variety of formats including cloud drives
	Support file storage in FTP, SFTP and WebDav
	Support file compression and decompression
	Support traffic control in the flow
	Support the creation of logic in flows with decisions, loopings, interactions
	Support queues and messaging
	Support generation of JWT tokens
	Support LDAP database manipulation
	Support interacting with proprietary format data
	Support miscellaneous conversions from and to JSON
	Support connecting to RPA solutions
	Support integrations with email solutions
	Support data sharing in distributed session
	Have a mechanism to assist in the transformation of IDs between systems, storing to-from tables and supporting automatic translation of IDs.
	Have a non-relational base for storing temporary objects that are necessary within the integration flow. It should be possible to store, retrieve, change and search for these objects in a variety of ways.

	All connectors documented with their connection and data transformation characteristics
	Support and Maintenance
	24X7 support for the SaaS or SaaS environment Managed by the platform vendor
	The solution must have a digital support channel embedded in the platform for interaction with integration specialists.
	All corrections and maintenance of the environment by the platform supplier
	All documentation of use of the platform in English
	All documentation of use of the platform in Portuguese
	The solution should provide Availability SLA above 99.5%
	The training should be given by the manufacturer in English and Portuguese