



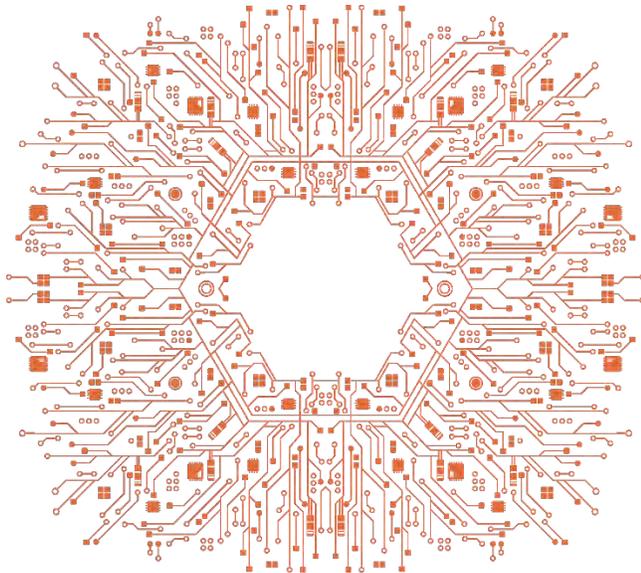
# Low-cost Enterprise Application Integrations

How serverless reduces the cost of  
enterprise integrations



# Contents

Introduction	2
Enterprise Application Integration (EAI)	3
Vendor Connectors	4
Integration Platform as a Service (iPaaS)	5
Server-based Integrations	6
Serverless Integrations	7
Conclusions	8



# Introduction

Systems are at the core of the competitive advantage that organizations possess, but often a lack of connectivity between them leads to silos of information that further exacerbate the departmental silos that form within organizations. As organizations aspire towards 'data driven decision making' they often find that the lack of connectivity between systems and data prohibits the timely identification of meaningful insights that might lead to better decisions. In response, many organizations proceed down the road of Enterprise Application Integration (or EAI).

Enterprise Application Integration is the process of connecting the organisation's systems and software. Integrations enable the flow of data from one program/application to another, providing new interfaces and the ability to manage and derive value from the data within the organization.

As more and more of today's core systems are complex hosted solutions, organizations seeking to undertake EAI often make the vendor or their IT service provider their first point of call. This leads to the selection of a range of solutions which either; benefit the vendor or are limited by the capabilities of the IT solution provider.

In both scenarios, and for a range of technical reasons discussed in this paper, the organization is unable to maximize its return on investment. Typically, this would be achieved by delivering the value of integrations while reducing their total life cycle cost. In the following chapters we look at the range of potential options available to enterprise and discuss strategies for the reduction of the cost based on an appreciation of the limitations of the solutions being offered.

# Enterprise Application Integration (EAI)

## Information Sharing

Without EAI, the flow of information between departments is often manual and labor intensive. When the operations department need to see a sales report to forecast workload, the Sales team need to spend time collating the information required from across multiple systems. For some organizations, the process of sharing information is just 'the cost of doing business', one that is an administrative burden.

EAI enables the flow of information between the core systems of the organization or external software platforms. It enables the consolidation of data and the sharing of this information with greater ease and speed to a wider audience. This translates to cost savings as departments can see the data that they need when they need it.

## Process Automation

EAI streamlines the process of synchronizing and moving data between the software platforms used by the organization. In addition, EAI can offer workflow automations that rely on system information. This minimizes the need for; manual extraction of data from a given platform, manual workflows that are repetitive in nature, and further upload or sharing of that information. With faster transfer and processing of information from around the organization, business units can access and make decisions on that information quicker. This leads to better decision making as it is no longer necessary to wait days for the data to be processed; it is available upon request.

## Improved User Experience

Enterprise Integrations can eliminate the need for multiple data entry in addition to providing a range of



other user experience benefits. As data is entered into one software package, it can be quickly replicated or transformed for entry into another. Since many departments use different software tools, this means having the information you need directly in the platform where it is needed, leading to higher quality data. Data quality issues can be a significant issue for companies which are striving to understand their customer, or other performance metrics. If there is conflicting data in different systems, this can lead to poor decision making. EAI can eliminate this by ensuring that one platform is the single source of truth for a given type or class of data, allowing all other systems to replicate from the trusted source. This reduces the need for checking the validity of data before use and drives operational efficiencies on multiple fronts.

## Timely Insights

New opportunities can be derived from timely insights based on the data stored across an organization's systems. This is enabled through EAI as data readily flows between platforms without human intervention. As customer data is updated in a CRM, this is made available to an ERP platform which can be used to plan for and manage projects. When all systems are connected it enables an organization to easily drill down from a strategic view of the organization to a tactical view which can support fine tuning of business operations. With EAI, the time to generate insights is significantly reduced when combined with modern Business Intelligence solutions.

# Vendor Connectors

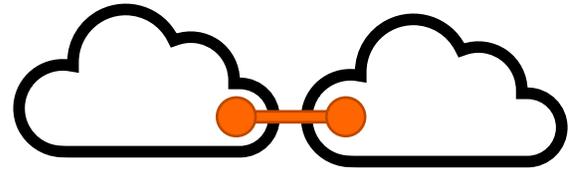
Some Enterprise Applications come with connectors and integration tools that are meant to solve the problem of getting data in and out of the platform. In some cases, additional payment is required to enable these solutions, and although connectors are often marked as a one-stop-solution for all things In and Out, unfortunately they rarely deliver on that promise.

When purchasing a new ERP (or upgrading), customers should consider the total cost of ownership of the platform inclusive of integration costs. Modern organizations can only thrive when they have fast access to data for effective decision making, and this means integration. Unfortunately for many Enterprise Applications, these integration tools have been somewhat of an after-thought.

## Difficult to Use

These tools are purported to simplify the process of integrating data from the vendor's platform to another and are marketed with simple workflow type interfaces that, in reality, are complex and difficult to understand. Unsurprisingly as customers become aware of the platforms, they are introduced to specialist developers who have been trained and 'certified' as partners of the vendor, meaning they are incentivized to ensure you are going to stay inside the ecosystem provided.

Some of the pitfalls of the solutions provided today are a product of the lack of a recognized customer or user. Vendor specific solutions appear built with neither developers nor customers in mind. In an attempt at providing something for everyone they end up becoming a solution for neither party. Developers are frustrated by awkward UX and functionality, and customers are overwhelmed by the complexity.



## Additional Costs

The marketing hype around these products is that it is necessary for them to be as complex as they have become, and that the best approach is to hire developers experienced in their use. Unfortunately for the customer, this means hiring a very niche skillset and paying a large amount for those skills.

In addition to the cost of developing solutions, customers are often surprised to find that these integration tools are separately licensed. There is the misconception that this is the only (or best) way to achieve what they are trying to do, and as the solution eventually moves into production and enters the support and maintenance phase, license costs combined with additional consumption based charged build to values well beyond their initial estimates.

## No DevOps

Without adequate DevOps governance and automation, another problem typically emerges shortly after "go live", where the integrations may not perform as intended and fixes need to be implemented while the integration is in place. Vendor solutions may not have equivalents of all environments used by their targets (DEV, QAS, PRD) and may lack appropriate automations to enable managed deployment of changes between environments. This all leads to a high level of uncertainty as to whether an integration will work and whether human error has been introduced along the way.

# Integration Platform as a Service (iPaaS)

iPaaS is a cloud-based integration solution which has pre-built tools for building and deploying integrations between cloud (or on-prem) based applications. An iPaaS generally enables basic connectivity to enterprise applications and the transfer and transformation of data between them.

The proposed benefits of an iPaaS are that they include all the tools necessary for the development and execution of integrations, support lifecycle management and monitoring of integrations, enable the application of business logic and are mostly elastic in design and scale well with workloads.

Unfortunately, while much progress has been made into the capabilities of these platforms, they often come with a range of downsides which leads to poor outcomes for customers.

## Lacking Features

Whilst generally the platforms are improving, today's iPaaS platforms often lack the full set of features required for data operations and business logic that a company requires to meet its specific needs.

Firstly, there is a problem of missing connectivity. Some enterprise applications require the use of custom / proprietary connectors which an iPaaS does not have, which then must be built or added at additional cost.

Secondly, extracting data from one system and inserting it into another is only part of the solution. When data transformations and business logic are required, the solutions provided to achieve these are often clumsy and require additional programming skills in unfamiliar or proprietary coding languages to deliver the goods.

## Niche Skills

iPaaS vendors advocate for their solution by claiming that citizen developers can make use of their low-code or no-code solution. On the surface, these platforms are 'easy to use' but many customers quickly realize this not completely the case. Without appropriately trained staff / contractors to deliver the outcomes required, the platform will fail to deliver a real ROI with development times and costs blowing out.

New users of iPaaS platforms face a learning curve like that of learning a new programming language, and considerable investment is required to do anything other than very basic tasks.

When considering the challenge of lacking features this problem is further exacerbated. Customers buy into the platform expecting a full feature set, however, inevitably require the support of qualified platform-certified developers to achieve anything other than a rudimentary outcome.

## No DevOps

Integrations built on an iPaaS by citizen developers may lack the appropriate governance required for enterprise grade software solutions. While there may be advertised benefits for rapid development, the lack of versioning and staged deployment as core functionalities means that developers may be required to rebuild solutions when moving from development into production. Without the right tools like CI/CD pipelines and Git / repositories, keeping track of changes to the solution becomes difficult over time, meaning customers are locked into working with those who originally built the solution.

## Security

One of the selling points of iPaaS is the enablement of citizen developers, however with this comes a greater risk of exposure of data. Care must be taken with all deployments that security policies have been implemented, connections to data sources are secured and API keys are appropriately managed. Without appropriate training an organization's data is at risk.

# Server-based Integrations

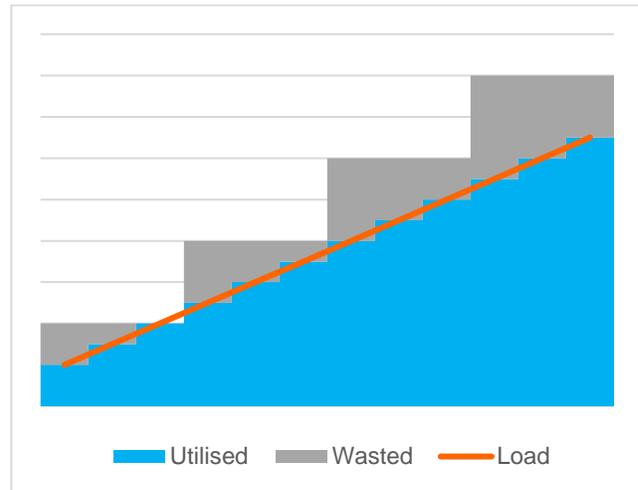
One alternative that customers find attractive is the prospect of a custom integration built on-prem or within their own cloud environment. The selling point for this type of service is that it is built the same way that all applications have been built before them (i.e. they run on a server).

If Cloud-based, these server-based solutions come with a range of advantages.

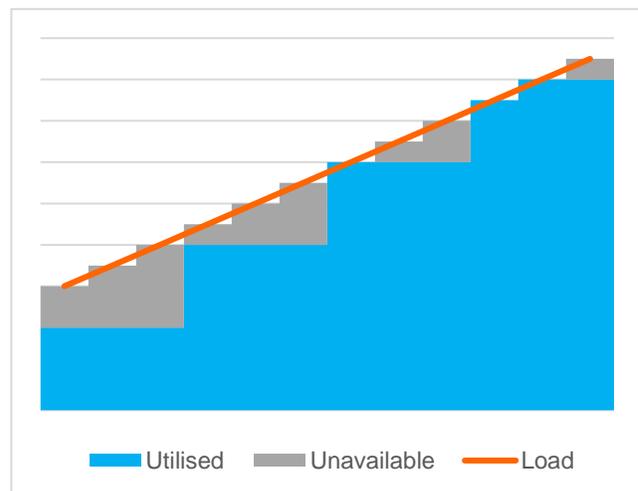
- No / low capex (unless a co-contribution is required)
- Opex model (rental and consumption based).
- Familiar coding environment.
- Performance can be increased through vertical and/or horizontal scaling.
- Easy and efficient fail-over and backup.
- Enables DevOps - CI/CD pipelines.
- Built in security offerings with SSO and OAuth2.

Despite these advantages, server-based solutions suffer from several disadvantages which either discourages investment or leads to risks which the business is ill-equipped to manage. These include, but are not limited to:

- Performance is capped to current instances unless investment is made in providing auto-scaling.
- Total Cost of Ownership exceeds the infrastructure cost.
- Outages hurt productivity.
- Susceptible to DDoS attacks.
- Require patching and updates / planned obsolescence.
- Require down-time for updates / code deployment.
- Struggle to scale with high volume of data without strong investment in routing and load balancing strategies.
- Performance bottlenecks may persist and be troublesome to eliminate.



Another challenge associated with server-based integrations exists in determining the appropriate hardware requirements. As the performance needs of integrations can vary over their full lifecycle, if servers are over specified this can lead to underutilized or 'wasted' resources. The chart above shows an example of this problem.



Conversely if servers are underspecified, this can lead to poor performance of the solution, with hardware induced bottlenecks slowing down integrations. This problem is best illustrated with a graphic comparing the provision vs demand of resources for a server-based integration solution.

Since under provisioning generally leads to a worse outcome than overprovisioning, the latter is usually the default solution leading to higher costs.

# Serverless Integrations

Despite exploring the alternatives, many customers today remain unaware of a solution that when implemented effectively provides low total cost of ownership and is low risk, robust and simple to understand.

Serverless integrations offer the advantages of other solutions without as many of the negatives. Benefits typically include.

- No / extremely low capex – no need to purchase hardware or licenses.
- Optimised low-cost opex - only pay for what is consumed/used.
- Performance is innately elastic and increases without investment in scaling code.
- Event based – each part of the solution is independent and does not need to run 24/7 when not in demand.
- Easy to understand coding environment.
- Automatic fail-over built in and easy to backup.
- Enables DevOps - CI/CD pipelines with strong deployment governance.
- Built in security offerings with SSO and OAuth2.
- No performance caps with highly parallel processing.
- Outages (if there are ever any) are generally limited to single executions, not the entire solution.
- Persistent services – can handle DDoS attacks.
- Do not require maintenance / patching of core services.
- Thrive with high data volume loads.

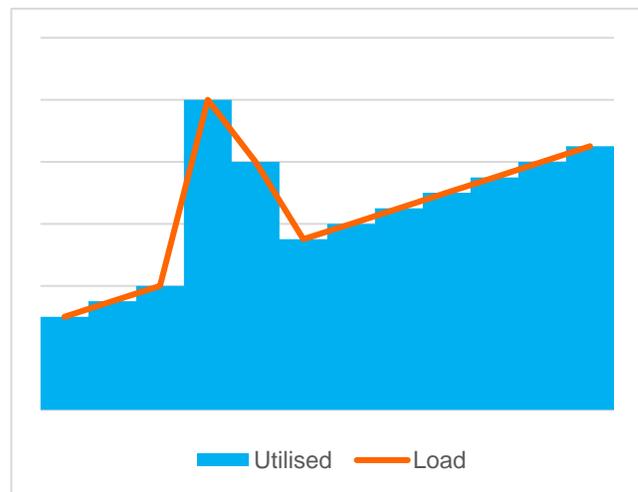
Despite all the advantages, serverless solutions have a few minor disadvantages.

- Performance may be throttled by demand on the shared infrastructure.
- Cold starts for rarely used functions.

- Customers lack experience with serverless technologies, a barrier to uptake.

Fortunately, most of these disadvantages can be mostly eliminated with an appropriate solution architecture or training.

Since serverless applications are highly elastic and scale directly with the load applied, users only pay for the resources consumed. The following chart demonstrates the impact of increased load on the provision of resources; unlike server-based solutions there is almost no underutilized resources.



Since serverless integrations can handle data streams with parallel operations, ultra-high volume and throughput and crucially the ability to manage peak loads becomes automatic.

The impact of this design is an improved experience for users who can expect integrations not to slow down just because the amount of data increases.

Powered by simple coding languages such as Python or Node.js, serverless applications are easy to deploy, configure and understand by developers. Development becomes faster and developers are unconstrained by the limitations of vendor solutions and iPaaS platforms, leading to better outcomes.

For the customer this means breaking free of the 'vendor lock in' that they have been experiencing and start to focus on deriving better and faster insights from their data without costing them the earth.

# Conclusions

When considering integrations for an Enterprise Application, most customers start their journey by contacting their vendor, or their managed service provider (IT partner), or an iPaaS solution provider for support. In many cases, this leads to costly and painful experiences.

- Software vendors are incentivized to maximize the Total Lifetime Value of you as a customer and will offer high-cost services by 'niche' developers to support your business needs.
- A managed (IT) service provider is typically underqualified to provide the right level of software development capability which leads to server-based solutions which are delayed, overrun and fail to meet the business requirements.
- An iPaaS solution will look great on paper but might not cover specific business requirements without substantial investment in the platform and staff capabilities to get it to do what is needed.

Unfortunately for customers, there is a lot of advertising and marketing out there that only leads them down a path of monetization. Searching Google for the disadvantages of an iPaaS, it is hard to come across any independent and unbiased sources of information. Despite this, it is only shortly after taking a particular route that customers start to experience the pitfalls of any one of those solutions.

## Serverless Integrations

The advent of serverless technologies has provided an opportunity to move away from costly integration paradigms, shifting instead to solutions which are highly elastic, scalable, and secure. Combining this technology stack with event driven applications offers new APIs and connectivity to new and legacy platforms and reduces the complexity of solutions significantly.



## Common Language

Leveraging standard coding languages such as python or node.js speeds up the development of solutions and helps future proof them by ensuring that any new software developer can more easily pick up where one has left off. Unconstrained by the limitations of vendor tools and iPaaS solutions, developers are more likely to deliver better performing solutions for their customers.

## Professional Development

Combined with built in security of the cloud computing environment and supported by common CI/CD pipelines, customers can expect development teams to deliver the experiences necessary for their stakeholders all while ensuring that production applications remain operational.

## Learn more

If you would like to learn more about serverless integrations, please feel free to follow us on LinkedIn at <https://www.linkedin.com/company/nanoputian-io/> or contact us at [contact@nanoputian.io](mailto:contact@nanoputian.io)



