CYCLR

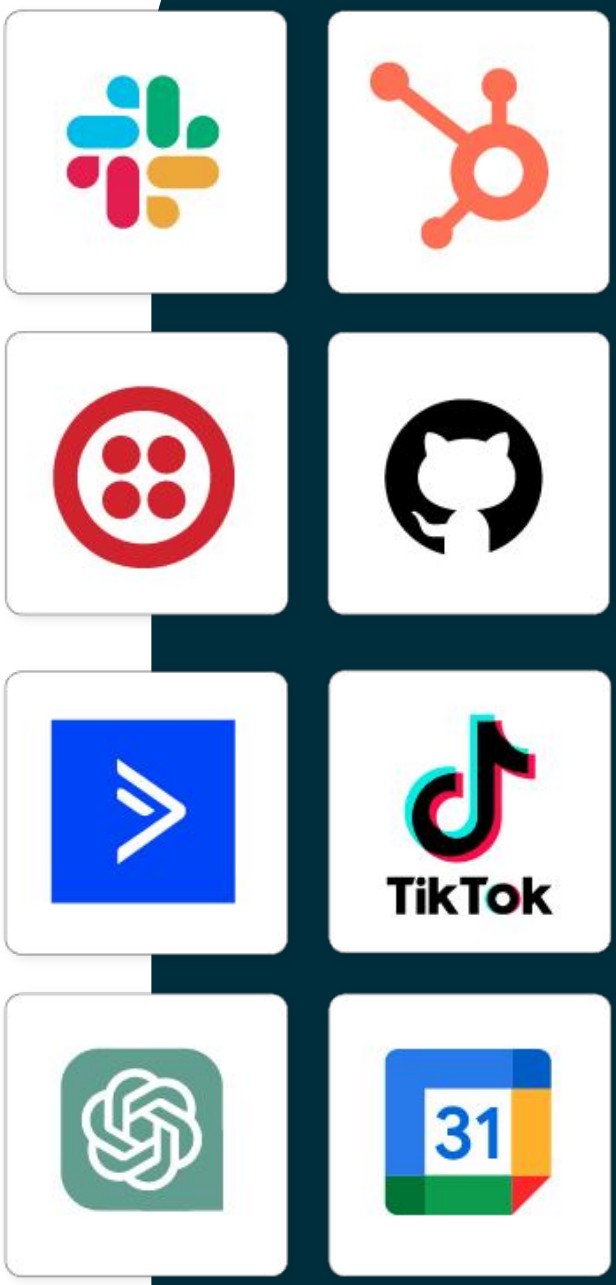
# An Introduction to Integration, iPaaS and Embedded iPaaS

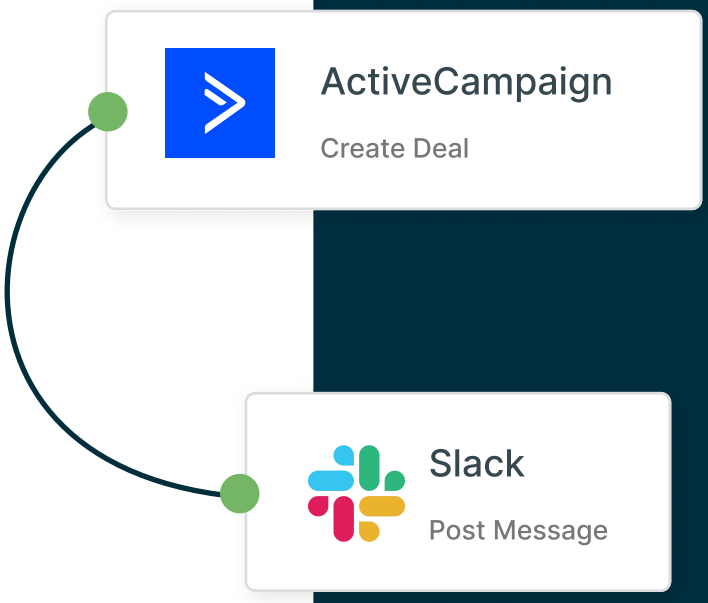
A Guide to Scaling Native Integrations in SaaS Applications

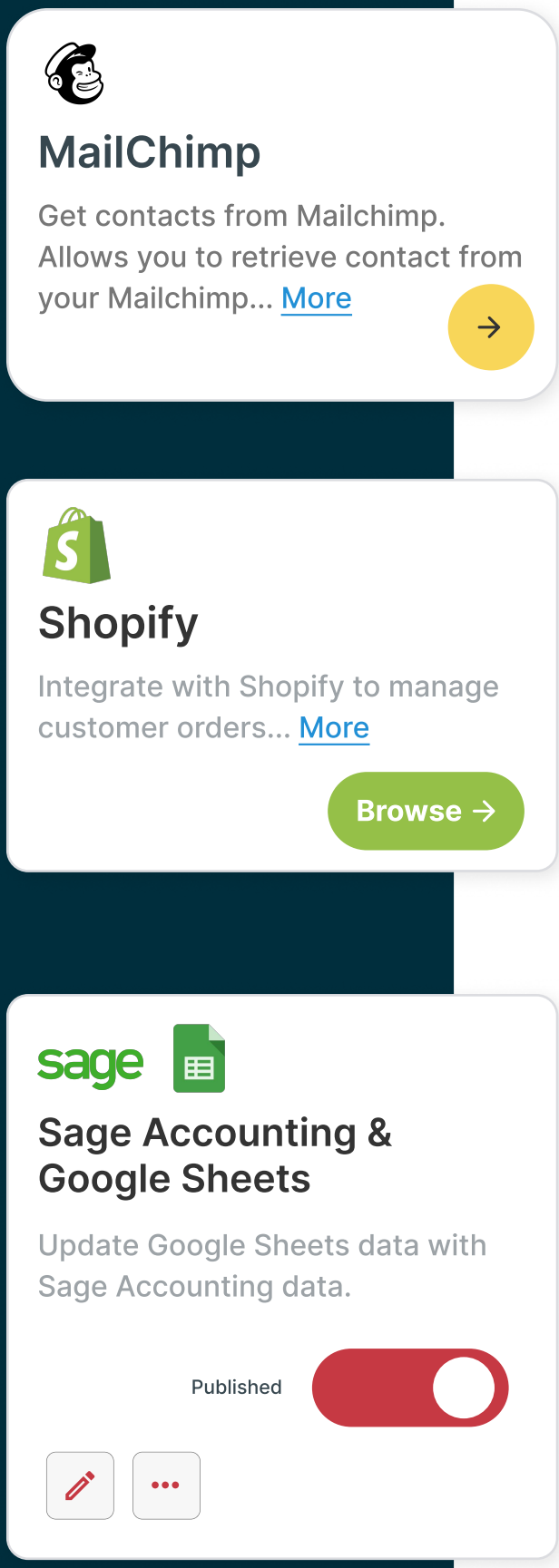
**Report**

# Contents








APIs, iPaaS and Embedded iPaaS are hot topics, yet not everyone is conversant with the differences, benefits and opportunities presented by the various methodologies available to integrate systems.

We have compiled this introductory guide to integration, iPaaS and Embedded iPaaS to provide a foundation for an initial assessment of the options available to you today.

# Where do I start?

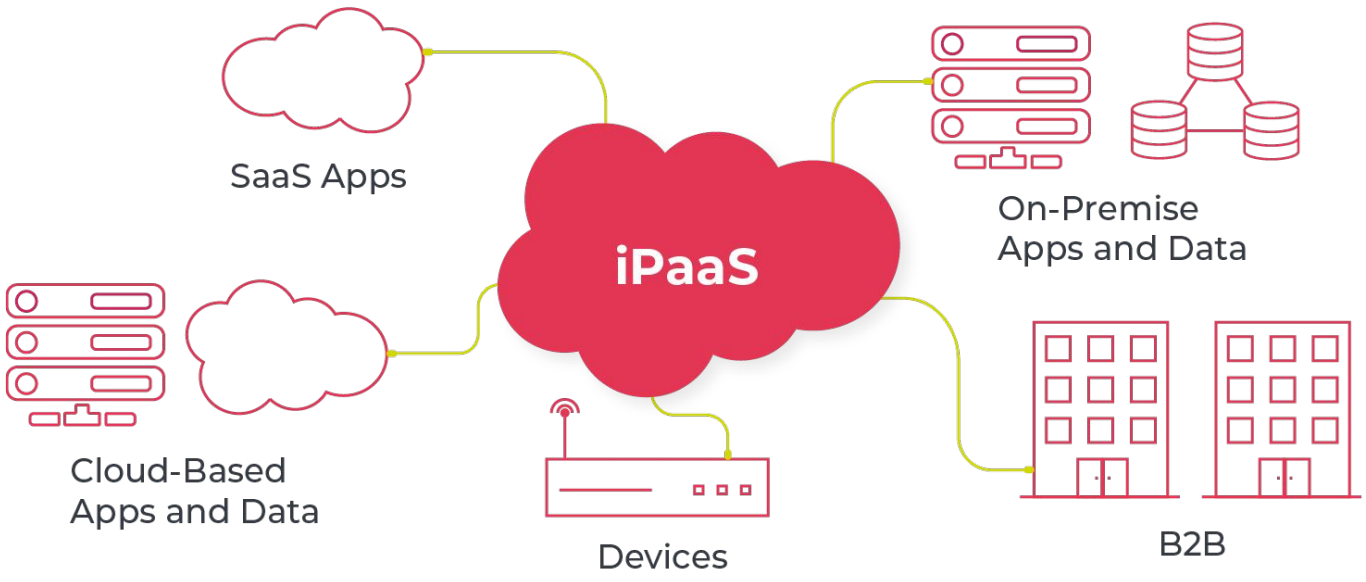
## What is an API?

An API is a ‘connection point’ provided by a software vendor to enable you to connect their application with other applications.

The API itself does not function on a stand-alone basis without some degree of programming to enable it to converse with another API. Connecting two APIs together so that they can communicate is the essential building block of an integration.

## What does iPaaS stand for?

Integration Platform as a Service.



**What do iPaaS solutions do?**
iPaaS solutions provide tools to aid in the creation and maintenance of integrations. They give users ready-made tools and elements to construct connections between systems. They provide a toolkit that enables APIs to be programmatically linked together.

As the iPaaS solution creates the link between systems, it is also the perfect place to run automations and logic, providing users with more than just simple updates from one platform to another.

**What used to happen?**
Before the existence of iPaaS solutions, the only way to create a link between systems was through having a bridge coded directly between them. These would largely be written from the ground up by a developer and be ‘custom’ for each individual company. A developer intensive process.

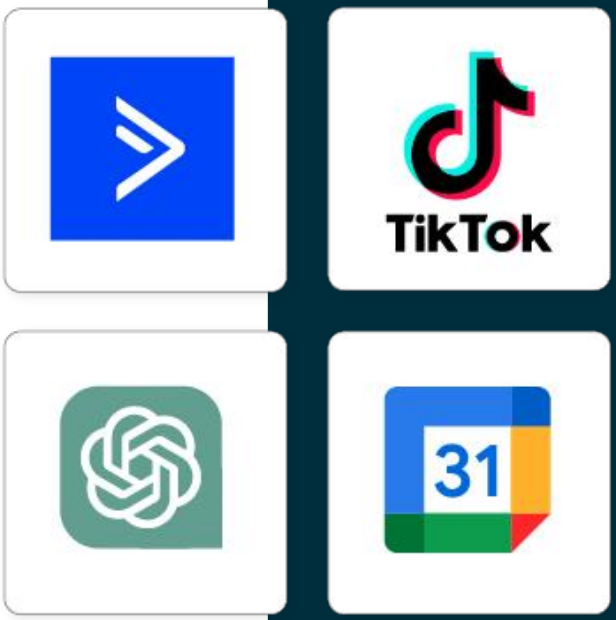
In addition, each bridge would have to be individually maintained. When a change was made to one of the APIs being used, developer support would be required to make any appropriate adjustments to the integration itself.

**Why does iPaaS exist?**
The rise of SaaS platforms, the proliferation of APIs and the inevitable desire for consolidation/ systemisation meant that the customer's need for integrations rapidly developed from a niche requirement to a general expectation.

However, with all of the different flavours of APIs in use, there is still no common standard for one API to instantly be able to talk to another.

iPaaS solutions create standardised, and hopefully easy to use, wrappers around APIs, translating them into that common language. This allows integrations to be created and data to be requested, manipulated and transferred without having to write custom code every time.

The iPaaS platform is a 'middleman' that facilitates easy integration between multiple disparate platforms. A well designed platform should be simple to implement, manage and maintain.



**What is embedded iPaaS?**
An API is a ‘connection point’ provided by a software vendor to enable you to connect their application with other applications.

The API itself does not function on a stand-alone basis without some degree of programming to enable it to converse with another API. Connecting two APIs together so that they can communicate is the essential building block of an integration.

**What are the alternatives?**
Integrations can still be manually coded using APIs and/or direct database access. They must handle user authentication (so users only have access to data within their permission scope}. These integrations have to ensure that the API versions being used are kept up to date so methods are not deprecated.

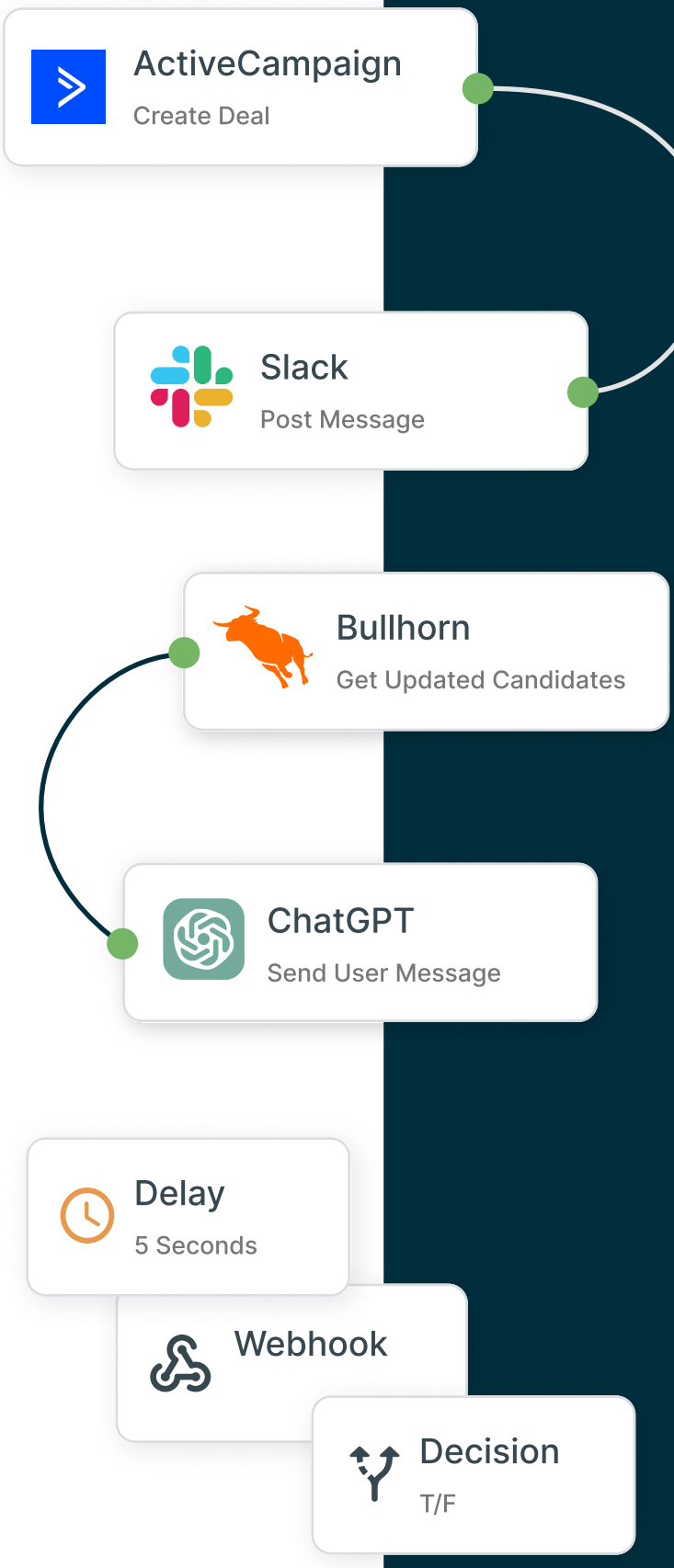
**The Term iSaaS**
iSaaS solutions are typically simper, lighter touch, iPaaS applications hat are open to the general man on the street to create and set up their own integrations. These are similar in scope, have a user friendly interface, but lack the depth of logic and automation capabilities a full iPaaS solution offers.

## What are the benefits of iPaaS?

Embedded iPaaS solutions give you a range of tools to speed up the creation of integration and ease the burden of management and maintenance.

A good Embedded iPaaS solution should ensure that the task of creating an integration is no longer solely a developer's responsibility.

Tools can be used by people who have sufficient understanding of the data and integration use case without programming knowledge. This frees up developer resources to work on core product offerings.



Embedded iPaaS allows SaaS vendors to rapidly expand their integration offerings without adding to their developer backlog by resolving key integration sticking points such as:

- Creating and delivering new integrations expediently.
- API versioning and updating.
- Individual integration maintenance.
- Handling and maintaining multiple API and authentication types.
- Integration being a developer-only task.

**“Embedded iPaaS solutions give you a range of tools to speed up the creation of integrations and ease the burden of management and maintencance.”**

**Who uses embedded iPaaS solutions?**
An API is a ‘connection point’ provided by a software vendor to enable you to connect their application with other applications.

The API itself does not function on a stand-alone basis without some degree of programming to enable it to converse with another API. Connecting two APIs together so that they can communicate is the essential building block of an integration.

| Platform Type | Typical User | Key Headlines | Example Vendors |
|---|---|---|---|
| iSaaS | Individual | Simple, easy to use, light functionality | Zapier<br>IFTTT |
| iPaaS | Enterprise for internal use | Comprehensive, feature rich, internal only | Mulesoft<br>Dell Boomi |
| Embedded iPaaS | SaaS company for integration delivery | Comprehensive, feature rich, internal use and external integration delivery | Cyclr<br>Cloud Elements |

**SaaS Vendors**

Embedded iPaaS solutions provide an interface to create integrations and automation workflows that can be deployed within their application.

The interface allows for a wide range of team members to create integrations, allowing developers to focus on the product.

**System Integrators**

Teams of people who specialise in creating links and automations between systems for their clients' companies.

An Embedded iPaaS solution allows system integrators to standardise their integration delivery and manage multiple clients through a white labelled system.

**Enterprise Companies**

These companies will run their business on a vast array of systems. They can choose between an iPaaS application or an Embedded iPaaS application.

Choice will be dependent upon how technically literate they are and if they want internal users to be able to interface with and deploy integrations.

# Build or Buy - Choosing an Integration Solution

## What should you look at when deciding if iPaaS is right for you?

So you are a SaaS company and the decision over whether you should build or buy an integration solution is an important one. In many markets the concept of 'Build or Buy' abstracts well and a choice between one or the other is easy.

In integration we would argue it is not so simple. We believe there are three viable ways to solve your customers' integration needs. Jumping to the conclusion, we are going to recommend that you adopt at least two.

### OUTSOURCE

Where you deploy a ring-fenced solution that achieves the customer's integration off-platform, an iSaaS platform e.g. Zapier.

### BUILD

Where you start from scratch and write code to achieve each integration system.

### DESIGN

Where you design integrations using an iPaaS or Embedded iPaaS platform, such as Cyclr, that is embedded within your platform.

# Benefits of Each Integration Solution

## OUTSOURCE

The benefit of deploying to a third party, off- platform, outsourced / iSaaS solution is that it is massively scalable, and agile, without disrupting your UI.

## BUILD

The benefit of building your own bespoke solution is that you own 100% of the code and you directly address a precise solution for your end customers. You are meeting their needs in a handcrafted manner.

## DESIGN

Designing a solution using a low- code platform that already has the components for connectivity built in puts your focus into designing the customer solution whilst the code grunt work has been done on your behalf.

# Disadvantages of Each Integration Solution

## OUTSOURCE

The downside is that you are not necessarily addressing your customer's integration requirements at 'point of need'. The burden is on the customer to leave your application and resolve their requirements (and any complexity} with the outsourced/iSaaS provider.

## BUILD

The downside is that these integrations can be fragile and the cost of human resource to maintain the integrations is high. It is a challenge to scale and is not particularly agile.

## DESIGN

iPaaS solutions tend to be both agile and scalable whilst also resolving the customer's need at point of requirement.

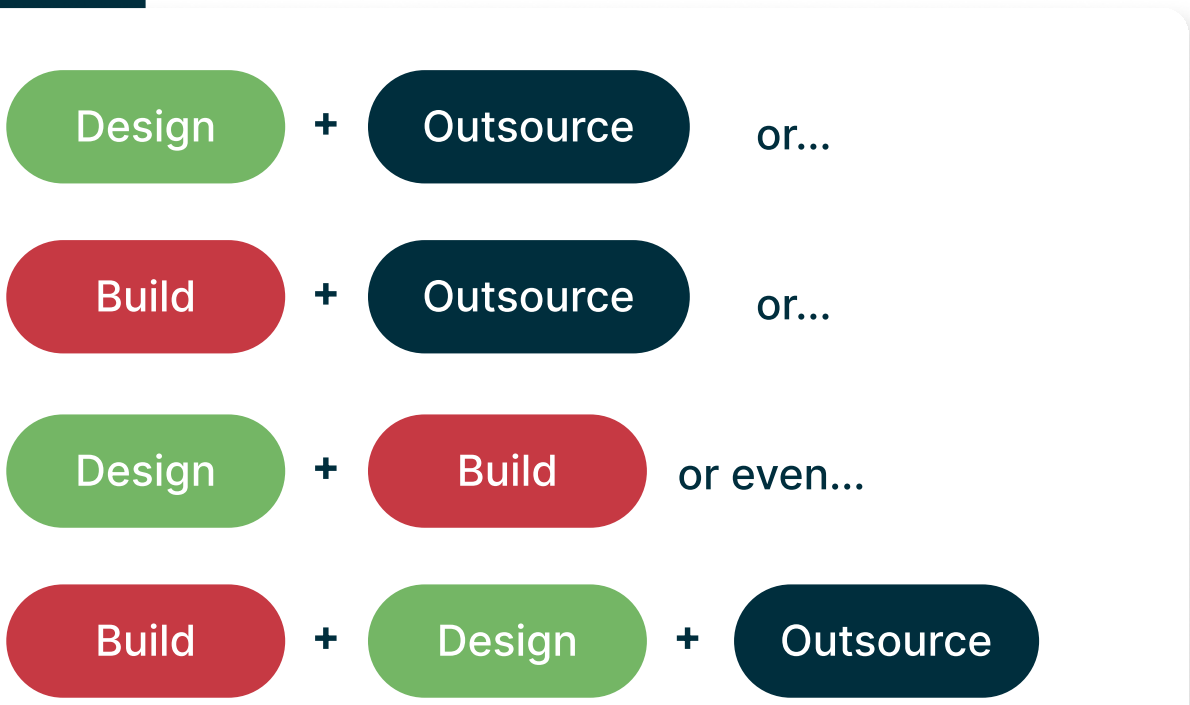
## A Combined Approach

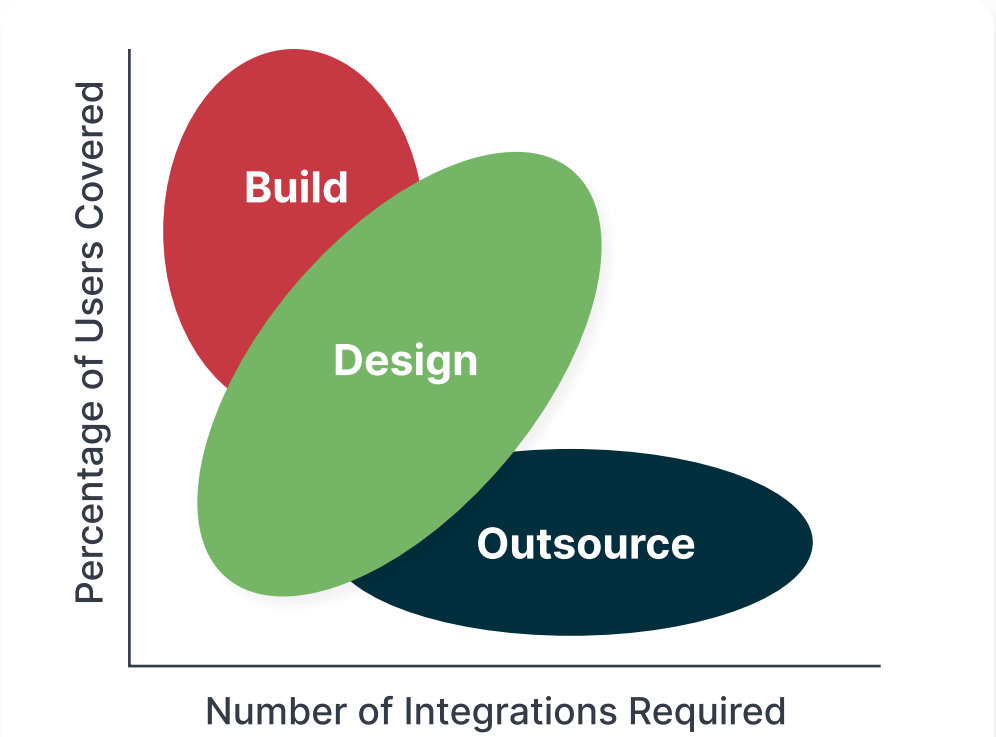
The 80:20 rule is a mindset to tackling integrations. This approach can help with efficiency and managing integration complexity.

We believe that you should natively deliver 20% of integrations (by volume) that solve 80% of your clients' integration needs.

Then **outsource 80% of integrations** (by volume) that **resolve 20% of your clients' needs**.

The approach can be tackled in a variety of ways:






We created 5 key questions to help assess which combination is best for you:

1. Is it important that I resolve my customer's integration 'pain point'?
2. Is my application lightly or deeply enhanced by third-party applications?
3. Are integration requirements consistent across 80% of my customers?
4. What is most important, the source code or the end-customer solution?
5. Are my customer's integration needs relatively fixed or do I need agility?

With these questions in mind, you can look at your platform, and your platform's users' needs to work out what combination would be best for you.

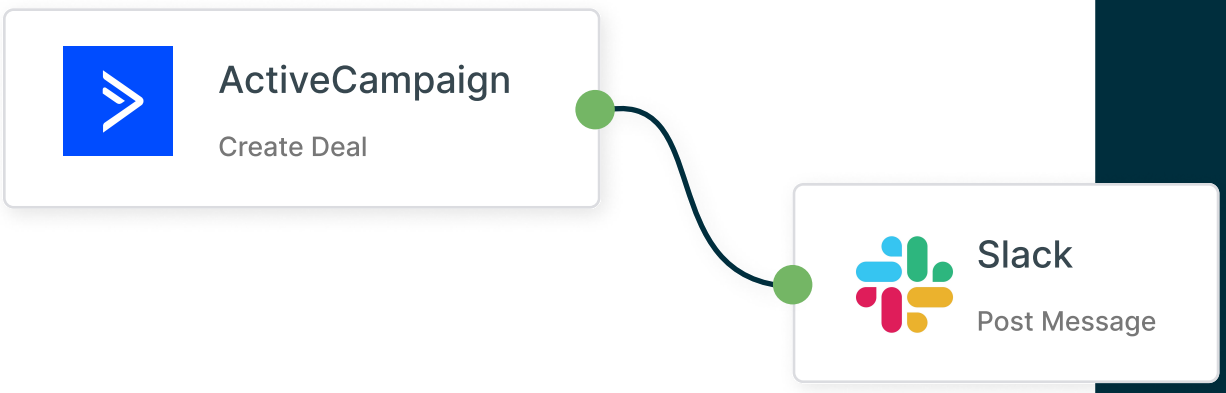


“You should consider the old 80:20 rule. Own the integration solution of 80% of your clients’ integration needs and outsource for 20% of your clients’ integration needs. It is all about efficiency and managing complexity.”

## Volumes of Required Integrations

Depending on the number of integrations you need, you cannot always resolve every single client integration requirement at point of need.

Why? Complexity of presenting an option for every single possible integration within your native UI. There is a simple rule of thumb on which choices we would make.




## The 80:20 Rule of Success

Embedded iPaaS offers an intelligent way to scale your native integrations. However, they by no means have to underpin the entirety of your integration ecosystem. You should consider the old 80:20 rule.

Natively deliver the 20% of integrations that solve 80% of your clients' integration needs and outsource the 80% of integrations that resolve 20% of your clients' needs. It is all about efficiency and managing complexity.

| | Outsource | Design | Build |
|---|---|---|---|
| It is important to resolve the customer integration at ‘pain point’ | ✗ | ✓ | ✓ |
| My application is deeply enhanced by third party applications | ✗ | ✓ | ✓ |
| My application is lightly enhanced by third party applications | ✓ | ✗ | ✗ |
| Integration requirements are consistent across 80%+ of my customers (focused) | ✗ | ✓ | ✓ |
| Integration requirements are inconsistent across 80%+ of my customers (long-tail) | ✓ | ✓ | ✗ |
| The source code is most important to me | ✗ | ✗ | ✓ |
| The end customer solution is most important to me | ✓ | ✓ | ✗ |
| My integrations need to be agile | ✓ | ✓ | ✗ |

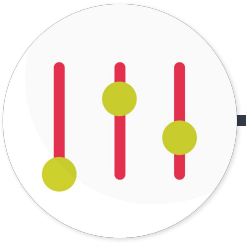
| Total number of customer integrations | How many to build | How many to design | How many to outsource | Recommendation |
|---|---|---|---|---|
| 1-2 | 1-2 | 0 | 0 | Build |
| 3-10 | 1-5 | 1-5 | 0 | Build + Design |
| 11-20 | 1-5 | 6-20 | 5+ | Build + Design + Outsource |
| 21-50 | 0 | 1-25 | 5+ | Design + Outsource |
| 51+ | 0 | 1-50 | 5+ | Design + Outsource |

# 6 Reasons Why You Should Use an Embedded iPaaS

## Native integrations offer the best customer experience

Customers want their problems answered ‘right here, right now’. If they want to integrate your application they want to be able to achieve this simply, and without leaving your application environment. In other words they want the integration to be ‘native’.

Non-native integrations are when the customer has to leave your application to achieve an integration result, either provided by the application at the other end of the integration flow, or provided by an independent (off-platform) provider of integration solutions.



## You retain control of your customer relationship

When a customer has to leave your SaaS application to achieve an integration you are losing a part of the ‘control’ you have over them. You are introducing the possibility for the customer to be invited to switch from your application to another and/or for the trusted relationship to move from you to a third party.

Say you are a CRM application and a third party application delivers the integration to your CRM, they may also offer other CRM integrations and therefore the ability for the customer to migrate/switch from your application to another. If the integration is native and/or delivered via embedded platform then this risk is reduced.

## Your time to market for new integrations is vastly diminished

Embedded iPaaS platforms are like a box of Lego bricks, with pre-built components, an environment designed for creativity and the ability to build whatever you choose within limited boundaries.

An Embedded iPaaS platform allows you to focus on the solution to your customer needs rather than the need to build, manage and maintain the component building blocks of the solution.

With an Embedded iPaaS platform you should be able to switch on new integrations rapidly, simply through activating them within your integration platform's environment. No need for scheduling builds and ongoing maintenance. The Embedded iPaaS provider should resolve this.

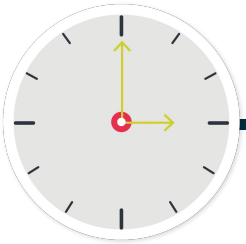
## Integrations can be delivered by developers or commercial teams

Why buy a toolkit to save developer time and only have it usable by developers? A well designed Embedded iPaaS toolkit should be appropriate and accessible to both developers (technical) and commercial staff (non-technical).

Integrations are a critical part of customer satisfaction and 'winning deals'. Therefore the ability for integrations to be rapidly delivered by those that understand the problem (the data flow) and are in communication directly with the client is critical to the resolution.

Truly resolving integration speed to market and responsiveness relies upon integration being able to be owned by the appropriate team at the appropriate time. Shy away from a solution that is designed for one single internal audience.

# 6 Reasons Why You Should Use an Embedded iPaaS

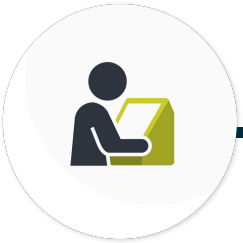


## It saves substantial development time and cost

SaaS companies are just that: companies. Developing integrations from scratch can be costly from a build and maintenance perspective. Both in time and therefore money.

Developing integrations requires a team, or a proportion of an existing team. Using an embedded iPaaS platform can substantially save on the resource required to deliver integrations and/or amplify the effectiveness of the existing integration team, enabling them to manage a much larger, more responsive, integration suite.

An Embedded iPaaS platform should be competitively priced against the direct cost of providing such solutions yourself.



## A single platform for in-app self-service and custom integrations

Why use two methodologies for fulfilling your users' integration requirements? While self-service integrations are a huge benefit to both users and development teams, there are always going to be exceptions when it comes to integration requests.

iPaaS platforms ensure you're covered in both eventualities; so you can create and publish self-service integrations as well as empowering your team to create bespoke, private integrations for your users. This opens customisation revenue opportunities, while ensuring that all of your team use a single platform.

“An embedded iPaaS platform allows you to focus on the solution to your customer needs rather than the need to build, manage and maintain the component building blocks of the solution.”

## Get in Touch

If you would like to talk about your integrations give us a call **+44 (0) 330 010 2525** or email **info@cyclr.com**

★★★★★
"Great product with amazing service"

★★★★
"Easy to adopt and adapt"

★★★★★
"Zapier on Steroids!"

★★★★★
"Integrations made easy!"

★★★★★
"Fantastic team to work with"


CYCLR

No. 1 Croydon, Office 6-11, Sussex Innovation Centre, 12-16 Addiscombe Rd, Croydon CR0 0XT

+44 (0) 330 010 2525 - info@cyclr.com