# The Easier Way to
# Manage Service Integrations

ONEiO

# Introduction

Welcome to this instructional guide for IT pros who are trying to get all their internal and supplier service management tools talking to each other and sharing data. Fully integrated software and services sounds great in theory, but what does it look like in real life? We know IT has been tussling for decades to get this stuff right, so what's changed?

At ONEiO we meet IT teams all the time who are fiddling around APIs, code and handmade integrations, in order somehow to automate the otherwise manual process of getting data out of one system and into another. Let's be honest, not much has changed in this space for a long time. Integrations tools and IPaaS solutions are very heavy lifting and cumbersome solutions. And the 'manual way' custom code and endless data sets... is now fun at all. Inevitably this also means that every day more of integrations break or fail and you get bigger and bigger headaches.

This is now officially the old way of doing things.
It's been around for some time and everybody has had their fair share of integration pain; and now it's over!

# CONTENTS

# Case Study:
## Large Service Desk

"We run a service desk tool, which my team and I use to generate tickets every day. We probably log about 100 support calls a day, but only about 20 of those of are things we can fix ourselves. The other 80 we need to collaborate with third parties, who either just take over the whole ticket from us till resolution, or we act as a middle-man, going back and forth between our customer and the supplier. E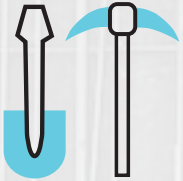ither way, a majority of tickets end up as multiple support calls across different suppliers and we need visibility of them all."

**Service Desk Manager, Large Retail Sector Service Desk**

## Doing it the 'Old Way'

"For some of the suppliers, we have to email them to report issues. They then send us a weekly report of ticket updates, which I go through and update our own ticket logs with. This is silly because it means we never really have up-to-date data, but it is the least disruptive way of getting a full view of everything.

For other suppliers, we have to separate out the incident reports from the bugs and change requests and log them in spreadsheets. At the end of each day, one of the people in my team uses a script he wrote to convert this data into a file, which then gets uploaded into our suppliers service management tool. We then get send the ticket data by email to confirm the requests have been raised. The supplier does the same thing the next day and our tickets get updated. This goes wrong A LOT and we spend hours a month trouble shooting the API. Alternatively; we just call them up to report an issue, they give us a ticket number and we wait."
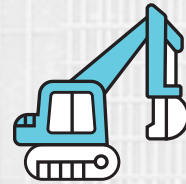
## Doing it the 'New Way'

"So that is how it used to be. Now we have a single integrated approach to working with all our suppliers. We have real-time connections from our service desk tool into the ticketing tools of our suppliers. So, we can at just the click of a button escalate a ticket within our own software and all the customer info, call log, categorization and service level data is automatically added to a new ticket within the suppliers software. Better yet, when that supplier updates the ticket those updates appear in our own ticket logs too (and vice versa). In reality there are of course multiple tickets open across several platforms. However, as far as the support staff are concerned everyone is working off one ticket.

We can also update configuration and policy data across systems too. So, if we want to change a service level agreement with a customer, when we amend that in our own system the same change is automatically updated on our supplier system too.

All the integrations we have are based on 'drag-and-drop' connections, so no more messy import/export of data or unreliable APIs!"

# How It Works
## (It's Pretty Easy)

Before we start setting up new integrations, let's take a look at some of the common language you'll see across ONEiO. You will see these terms appear a lot in this guide, so it's good to know what they mean.

## Messages

ONEiO is essentially built on our unique messaging system. Every ticket, alert, notification or change that is raised in either your own or your vendor's systems will be communicated across in a message. The message is essentially a packet of data. Because messages are all managed by virtual brokers, messages never get lost. Even if the receiving software is offline, messages will be queued till it's back up and running.
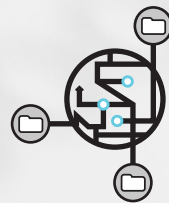
## Brokers

Brokers are light weight applications within ONEiO that route messages. Brokers know all your rules and configuration settings, so when a message arrives with a service ticket inside, the brokers knows exactly which of your multiple suppliers that ticket needs to go to. The same applies when ticket data is sent back from a supplier, the broker will find the original ticket data in your own system and update everything automatically.

## Rules

Every integration you setup has a set of rules to ensure that all tickets are routed to the right place. In addition to routing data, rules contain information around SLAs, event thresholds and ticket categorization. This means that you can easily report on real-time ticket data from your supplier's systems, and have all the automated notifications that you would normally have setup in your own tools.
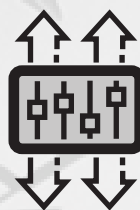
## Adapters

Adapters are the final piece of the puzzle. Adapters are basically data ninjas that take whatever coding, data and configuration is inside your messages and translates into the exact information the system at the other end needs. Adapters sit on both your side and your suppliers side of the broker, not only translating data into system specific languages, but into the single language used by the ONEiO broker.
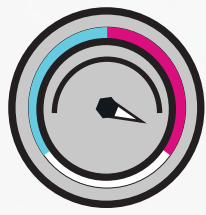
# Setting Up a New Integration

To setup a new integration you first need to subscribe to the ONEiO network. This take a few seconds and you can do it for free. You only pay for what you use, and your first two weeks don't cost anything at all.

Once you're in, you need to you select the apps, which will make up your first integration. You have a list of available apps, such as ServiceNow, Jira or ZenDesk, which you can start using right away, with no coding, wait time or developer involvement.

No need for API's either, it is all pre-built within ONEiO. So, you just have to select the apps you want, drag and drop them into your integration hub and click done!

The next step is to set your rules. Your rules define what happens when tickets arrive, get opened, closed, assigned, escalated and so on! For the most common apps, you can use our ONEiO "Rule Robot' who will setup the basics for you. For your first few weeks, this might well be enough to start using your integration.

The next step is to just pop open the popcorn... or just get on with some other work! And Let ONEiO look after everything else!
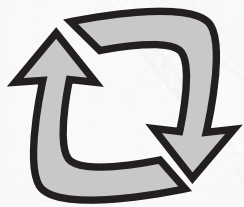
# How to Change an Integration (It's Super Easy)

A big challenge with heavily scripted, manual or API-led integrations is that when you want to amend something like a SLA, categorization or technician assignment rule, you could be creating days or weeks of work for yourself.

In ONEiO all the elements that connect the rules between your software and services are automated. It doesn't matter if you use different terms or spellings for an 'incident' or a 'service request' - ONEiO creates a single language inside the adapter - so you never have to spend hours hunting for the field you need again.

To change an integration you simply login to ONEiO and select that integration. You can then head over to the rules tab and edit any of the rules attributed to that app. Any and every change is saved, and you can roll-back to a previous configuration at any time. So no more accidently deleting a line of code... and freaking out!

Best of all, your rules and settings are managed in one language within ONEiO and changes are replicated to the relevant systems automatically and in real-time, so you only have to make your changes once.

# Common Questions

## How do you explain this to your boss?

Most of the time, managers don't want to hear about how much code you do or do not have to write, they just want to hear you talk about the time and money involved. With regard to setting up new integrations, you can explain that you have developed a more efficient way to leverage the value you get from a supplier by automating ticket exchange and saving a lot work time (and therefore money).

This provides greater transparency and control and enhances SLA reporting from both your own pro- cess and your suppliers. This is important because although you can outsource a service, you still retain the accountability for the value it creates for your business.

## Can I disconnect an integration temporarily?

Sometimes when you are making complex changes, or switching mappings from one system to the next, you will want to take everything off-line for a while. Traditionally the danger here is that downtime means any attempted communications made during that time get lost.

ONEiO's solution to this is that if the Broker isn't able to route a message it will simply queue it. When the adapter it is waiting for comes back online, it will check what the new routing rule is and send all the queued messages through the new mapping.
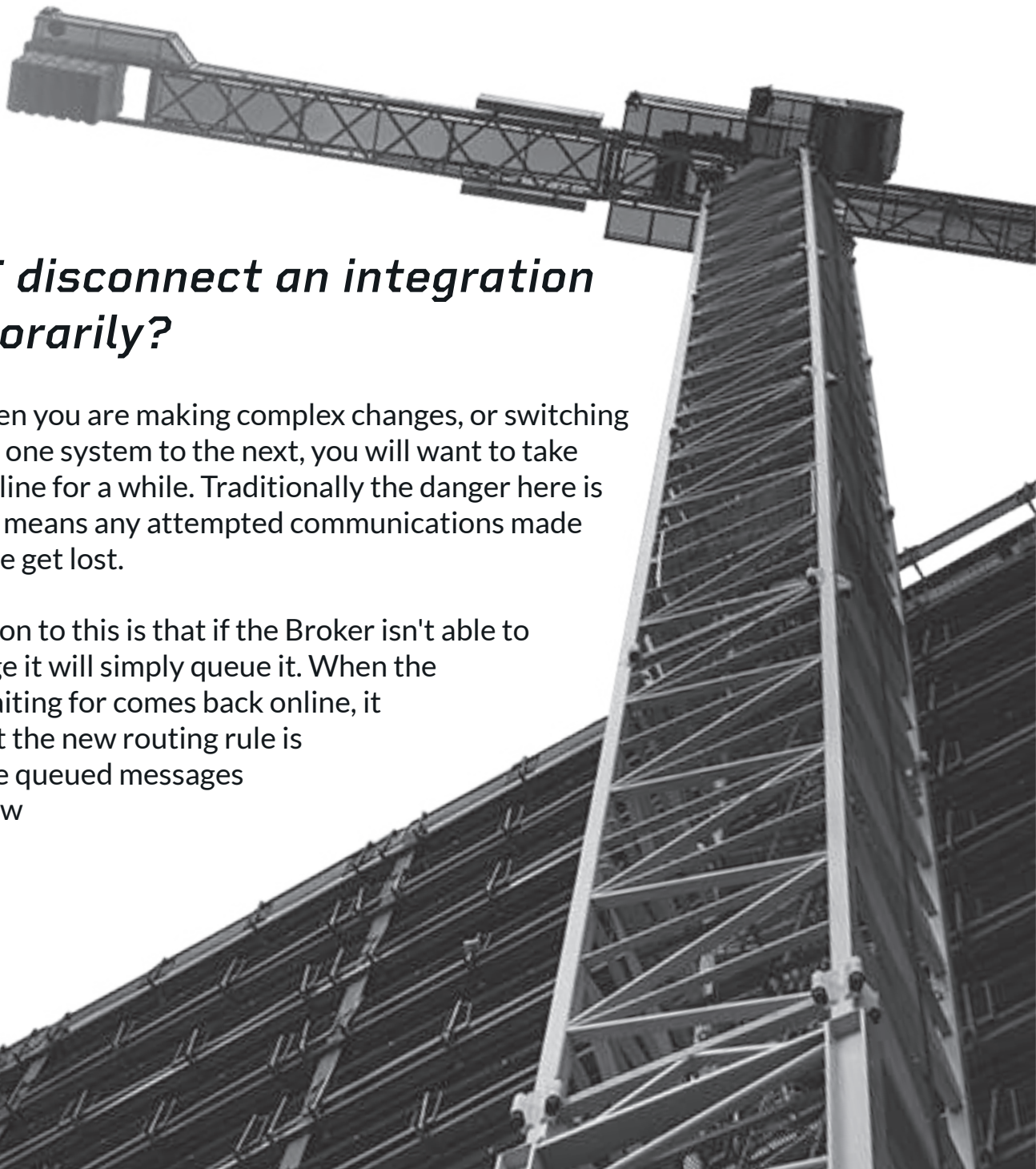
# Using a SaaS Solution

Every business is now choosing Software and a Service (SaaS) solutions to solve their IT and ITSM challenges. This is because SaaS tends to provide all the same features and benefits of traditional on premise solutions but with added flexibility and significantly lower costs.

Until recently there has been little to nothing available for IT teams who wanted to use a SaaS solution for multivendor integrations, particularly for multivendor integrations, particularly for companies pursuing a Service Integration and Management (SIAM) framework. This big gap in enabling simpler, faster and cheaper integrations is what actually motivated us to develop ONEiO. So, let's take a look at some game changers using SaaS for service integration.

## It's a subscription

Software and infrastructure are like new cars, as soon as you buy them they start to lose value. So why buy when you can rent? We also know from experience that when purchasing new software, the cost is never the cost. There will always be more to pay for such as implementation, configuration, maintenance, licensing, renewables and so on.

In addition to the software costs, you also need to consider hosting and infrastructure. Do you need to purchase new servers to host your software on? If so, perhaps you need to pay your staff over-time to build and deploy that server outside of core business hours.

If you have existing servers in place for your new software, what research and preparation must you do to ensure they are fit for purpose? What changes are required in order to get those servers ready for the installation, what financial and staff resources will that require and how will you carry out testing? The short response to this is that buying software is expensive and no matter how many times you have done it, there will always be something you didn't expect to pay for.

When subscription software is done well it removes large upfront costs and provides predictable and controllable spend. You can even create more flexibility around shifting your subscriptions across Capex and Opex spending.

## SaaS equals speed and agility

Even a standard change within your own IT environment can come with major fallout. Small changes within complex ecosystems can create wide spread and significant issues. SaaS enables you to relinquish your ownership of the platforms that house your software and thus the responsibilities and risks that come with ensuring that platform is robust enough to support any change you need to make.

# How to Get Started

Great productivity comes down to one thing; collaboration. But in the enterprise IT environment, it isn't just about having a better working relationship with the person sitting next to you, it is about the quality of collaboration across all your teams, suppliers and tools. You achieve this by building more seamless and agile connections between these different functions.

**If you would like to get started with ONEiO, you can do so today. You just have to follow our four step process:**

## ① Subscribe

Join the ONEiO network, for free and with no obligation to buy.

## ② Connect

Add your first apps to your integration hub.

## ③ Rules

Setup your initial set of rules for how these apps will work together (don't forget to use the Rule Robot!)

## ④ Sit Back

Kick off your shoes, open some popcorn and watch your business become better connected than ever before!

# Meet our team and get in touch

ONEiO was originally founded as Service-Flow by Juha, Janne and Kai. The idea was that the world of service integration was heavy, slow and outdated and it was time to flip the industry on its head. Built upon several decades of experience, the outcome was a first-to-market SaaS solution for service integration, which in 2019 was recognised to be the Next Generation of iPaaS.

After welcoming a family of international clients, investors and an amazing team of staff, ONEiO is now one of the most exciting IT brands to work with.

Here are some of the key contacts in our team, who would love to hear from you:

**Juha Berghäll**
CEO
juha.berghall@oneio.cloud

**Erkka Heinola**
Business Development
erkka.heinola@oneio.cloud

Let's start your journey towards a more integrated future today!

ONEiO is headquartered in Helsinki, with offices in the USA, London and Germany.

For any sales, consulting or product enquiries, you can email or call us:
E: flow@oneio.cloud
T: +358 10 325 4200

For any technical support enquires:
E: experience@oneio.cloud

And for any partner enquires:
partners@oneio.cloud

We are also always on the lookout for talented new staff. From sales and marketing, to software development and technical consulting. If you would like to find out more about how you might fit in to our fun, creative and hardworking team, you can get in touch via: tiina.eronen@oneio.cloud