exalate

# SaaS Integration Challenges and Benefits

# Table of contents

*This blog post is based on this episode of Integration Talks, a podcast on everything integration.*

https://www.youtube.com/watch?v=s8UYqJ1UaWk

Companies are increasingly using SaaS platforms to work in the cloud. Connecting these platforms (SaaS integration) lets you share information and work together more easily.

Let's look at some of the challenges and benefits you'll encounter when integrating SaaS platforms.

Learning from others' experiences is always valuable, as integrations can get complex.

Here are a few insights from an experienced professional, Boris Berenberg.

- There are complex edge cases when dealing with data owned by other teams.
- You don't own other teams' data so can't make all decisions.
- Integration is a wide concept and it can get messy.
- Customers often shift from 1st-party to 3rd-party solutions.

There are several questions to ask to help you successfully complete an integration project.

## Do you Need a SaaS Integration in the First Place?

In some cases, it might be easier just to move from one system to another, since one instance contains all logic, and it's obviously easier to manage.

There are of course some downsides:

- You have to retrain all customers and agents.
- Other integrations and plugins still need to work.
- All these tools need to be shifted, which can snowball.
- It could potentially be very disruptive to business practices.

An integration needs to offer something to make shifting worth it. Ultimately, it's about business value.

How about the major upsides of using an integration?

You get reporting across multiple systems - this is of course a challenge on multiple systems. But reporting is a magical unicorn that every executive wants.

Often, senior stakeholders want cross-platform reporting. The idea is by centralizing inputs to one system, you can tell if engineers are performing.

There are some inevitable challenges with cross-company integrations as well:

- You can't ask companies to switch to other systems.
- There are security and privacy issues that won't customers have access to other systems.
- You have to limit the data they have access to.

So in this situation, permissions become complex and disruptive to everybody, information sharing has to be limited to what vendors should see, and you have to comply with everybody's data classification policies.

## What's the Minimum Functionality for A SaaS Integration?

Integrations tend to get complex with all the business rules.

So there are two potential approaches; doing everything from the start and iterating (starting small and improving along the way).

You should also think about the MVP (minimum viable product) for an integration and the capabilities of an integration provider.

Some tools, such as Zapier, Integromat, and other no-code tools, can build connections quickly.

These can be used by those with an understanding of business needs and basic technical skills, such as project managers or data scientists. They can deliver MVPs quickly.

Those MVPs have some issues though. Such products can have a short lifespan.

It can become difficult to manage. If you want to troubleshoot you have to rerun everything, so eventually it gets very messy to maintain.

They also often fail security compliance - especially with multi customer systems.

Teams need organizational autonomy over security policies and data classification.

So such solutions have difficulties handling this model.

A cross-company specialized solution is needed to take things further and to make sure that the SaaS integration is kept up and running in the long run.

In the Atlassian ecosystem, Atlas Authority uses Exalate, which is a cross-company integration solution.

## The MVP (Minimum Viable Product) of the Integration

The MVP only gives you access to the info.

Next, you need to see it natively in the tool, then let users work in the UI of the product.

From there you can add scheduling events and bi-directional sync, but this is where you need to stop loops.

Note that you need to think about heuristics to stop loops in bidirectional sync. You should also be aware that different configurations might easily be broken.

### The Human Factor

When teams work together the tool is only 5% of the effort. Understanding each other's goals takes time.

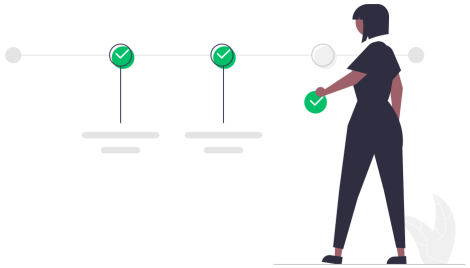
The strength of a SaaS integration solution like Exalate is that it features a rich bi-directional configuration allowing both sides to have autonomy and full control over their incoming and outgoing data.

A downside is that a configuration change needs system owners on both sides to push it out together.

There are also some Challenges that we need to address when it comes to cross-company integration in general:

- Scheduling conflicts
- Negotiating field value matching
- Information share violating security promises

It's also challenging from a product management perspective, you have to balance usability and how much you can help teams collaborate on configuration without compromising security guarantees.



In this case, product management tools can't solve these issues: You need a shared vocabulary and clear discussions to have a correct understanding of processes.

### How can you achieve this?

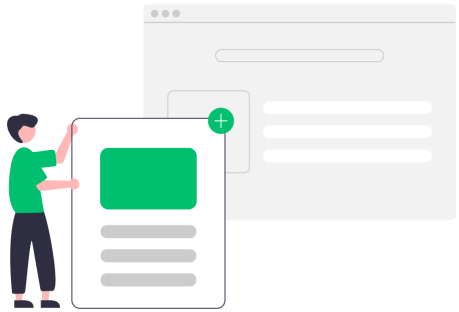
Have the right stakeholders draw the processes. You usually discover a lot. The biggest cause of wasted work is a lack of shared understanding.

For example, what is an issue? It actually means different things in different contexts, right?

So it's important to have everyone speak the same language especially when it is outside borders and across different companies.

## Solutions:

Well, unfortunately, there is no silver bullet! But you can set boundaries so you keep moving forward without derailing.



Get the right stakeholders - not the easy ones. The right stakeholders are the ones with executive authority, who can make decisions (not all CEOs can!). They should be able to provide structured input.

You can then give those stakeholders specific questions and options, and figure out how the processes should be implemented.

***Tip***: Avoid group brainstorming, it can only postpone discussing what's actually important at this stage.

Scheduling can also be quite tough. But standing meetings can easily work. Book a fixed slot on the stakeholders' schedules and meet regularly. Note that you need to make it known that the meeting can be canceled if needed. And by respecting everyone's time, you can build trust.

## The Iterative Approach

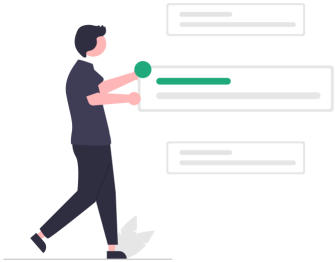
The iterative approach will actually work in this case and will work better than the classic waterfall approach (which eventually leads to an iterative one).

Because:

- Most people don't know their own requirements.
- You can dynamically discover/build.
- Goals change during discovery and implementation.

You better set some initial guidelines and define an iterative process for the best ROI. This way, the right expectations will be created even though you don't necessarily know what you will encounter on the way.

Of course, with networks between companies, you can't predict all the permutations between individual teams. So it can become an uphill battle.



But remember that you can make anybody happy if you set expectations correctly.

There are different ways of managing the delivery gap and different ways to understand what people are saying.

For example, if you integrate incidents in ServiceNow with Jira tickets but not changes with Jira tickets, people can either be unhappy about the changes that don't apply to their system, or they can be happy with what the integration provides.

Now to set expectations:

- Deploy one team at a time.
- The spectrum of change acceptance - some like it, some don't
- Be clear that when changing, things can get worse before they improve.
- Start with stakeholders that are actively engaged in discovery.

You better also prioritize enthusiastic teams first, willing second, and not keen, last.

There are different ways of getting teams to buy in. The first two can validate the process for the third. People who don't get it first, often want it more. And you can set a fixed time period with the option of reverting - well most won't.

### Estimating the turnaround time

The turnaround time depends on:

- Scope
- Number of teams
- Types of entities being synced

For instance, a Jira to Jira integration is easier than Jira to other platforms because the entity concepts are the same and there are no requests for features in A which B doesn't support.

Perfect bi-directional sync between two Jiras with no user mapping could take an hour or two. But if you add features, the scope can expand. What's important, is to tell clients how changes affect them.

## Conclusion

This podcast covered many interesting topics surrounding SaaS integration. There's a huge amount to consider, so expertise is critical. Exalate and Atlas Authority both have years of experience making integrations work and can help you connect your teams as painlessly as possible.