

Why embedded integration is a game changer for SaaS vendors

Table of contents

- 1 Introduction
- 2 So what's the problem?
- 2 A victim of its own success
- 3 Integrations are for life
- 3 Why embedded integration is a win/win
- 4 iPaaS vs. embedded iPaaS
- 4 Introducing webMethods.io Embed
- 5 Conclusion

White paper



The triumph of the cloud is complete, and there are no signs that cloud innovation is slowing down, although developers might soon run out of as-a-service acronyms. The evolution from SaaS to PaaS to IaaS and its offspring, iPaaS, is beginning to get confusing, but businesses are not fazed by this and are ramping up cloud investments.

Gartner projects that global revenue from public cloud technologies will reach [\\$330bn by 2022](#), up from \$180bn in 2018. SaaS will take the lion's share with \$144bn, an increase of 80%. Even more mouth-watering for SaaS vendors is the projected total spend on cloud software over the next few years: a colossal \$350bn.

So what's the problem?

Paradoxically, vendors have never faced greater challenges. Rival apps are coming out of nowhere as businesses recalibrate their software stacks by ditching unproductive apps. How can vendors respond to this double squeeze and differentiate themselves?

Above all, the market is seeking speed, flexibility and a coherent user experience. This is why headless architectures for CMS and e-commerce are making such an impact. By decoupling the backend from the UI, developers for both can make rapid changes and innovate flexibly without getting in each other's way.

A recent extension of this approach is "headless integration" where SaaS developers embed integrations into their front end without having to program them from scratch. Known as "embedded integration," it solves a problem for both SaaS vendors and end-users: vendors achieve the speed-to-market they need to remain competitive, while end-users become more productive as they access and experience their favorite apps from one UI.

In the first part of this white paper, we will examine why integrations present such a challenge, and why it is not feasible for vendors to build these integrations themselves. This paper will also explain how embedded integration works, and why it is a breakthrough solution, both for SaaS vendors and the different markets they serve.

A victim of its own success

The more apps you have, the trickier it gets to move swiftly and meaningfully between them. How many apps the "average" business operates is difficult to determine, as most CIOs have incomplete control over what sales, marketing, HR and finance subscribe to independently. But we all know it is a lot, and probably too many.

A recent [survey](#) concludes that businesses with between 200 to 500 employees have 120 apps. Smaller businesses or start-ups without "legacy" have fewer, obviously, but they too accumulate SaaS products pretty quickly and juggle an average of 40 cloud applications.

SaaS spread like wildfire because it makes end-users more productive. However, lack of integration, which forces end-users to go back and forth endlessly between the apps they need and like, is disruptive because it corrodes productivity.

We are wasting a lot more time than we think. Almost [70%](#) of employees say they spend up to an hour a day navigating between apps—a horrible 32 working days a year. Time is lost, but so is concentration as the jolt of dealing with so many different UIs makes the experience of all apps—and that includes yours—fatiguing, disjointed, even disorientating. A third of employees say they lose their train of thought as they move restlessly from app to app.

Just as significant is the extent to which we resent having to do this. Switching between apps is seen as more annoying than doing household chores or trying to lose weight by respectively 53% and 50% of employees.

And just as we are too lazy to tidy up or order pizza even though we're dieting, end-users will eventually give up on apps that slow them down too much, even if the app itself is very useful.

An app is only as brilliant as its commercial viability and in the end, SaaS developers have to give end-users what they want. What they emphatically do not want is an integration project gift-wrapped as an app. Apps packaged with a bunch of APIs or a referral to an integration partner are no longer acceptable to end-users who are turning the very rationale of SaaS against the vendors. "If speed-to-market and agility are key benefits of SaaS, how come you are making us waste time on integrations?" is the essence of that line argument.

No, the burden of integration rests squarely on the shoulders of the SaaS developers.

Benoit Lheureux, VP at Gartner, reaches the same [conclusion](#). "Redirecting your customers to a third party for integration solutions when integration is a requirement in every IT project of substance is falling short of meeting your customer requirements ... SaaS providers should have bitten the bullet and given their customers integration capabilities, rather than forcing customers to buy them from a third party." This advice is valid for all SaaS vendors, irrespective of their target market. Is it too late for vendors to give their apps these "integration capabilities"? How can vendors bite the bullet of integration?

Integrations are for life

It is never too late. Vendors can build their own integrations or embed them in their applications, as we saw.

The attractions of owning and customizing your own integrations soon collapse under the weight of what that involves. You have to drag your developers away from their real jobs to write and maintain code, not once, but over and over again for each and every integration, and you need connectivity with 10, 20 or more third-party apps to make your product stand out. That's a lot of code. And as we said, each app, and so each integration, is unique—there are no real economies of scale. Integrations never end; they will continue to divert time, talent and money from where your focus should really lie: your core product. One distraction associated with integration maintenance is that APIs have to be monitored regularly to make sure your (multiple) integrations do not break, a routine task for which most developers have little natural affinity, and absolutely no enthusiasm.

Then there is our old friend, speed-to-market. Integrations take time, and a vendor that can sidestep them gains a considerable head start on its rivals, perhaps three months or more. Often, those three months is what separates a successful app from an app that drops off the cloud or is never adopted.

How do vendors offer integrations and avoid having to build and maintain them? By "prepackaging" them within their app through embedded integration.

Why embedded integration is a win/win

As we observed earlier, embedded integration solves a problem for both vendor and end-user.

For vendors, it solves the problem of building custom, native integrations. Embedded integration solutions usually provide out-of-the-box connectors to the most popular SaaS applications, so these integrations can be brought in-app very straightforwardly through an SDK.

This greatly reduces the development and maintenance costs related to integrations—sometimes by as much as 50%—and accelerates the vendor's time to market because the integrations are delivered so much faster.

The enthusiasm and creativity of SaaS development teams is brought to bear not on the unrelenting task of building and maintaining integrations, but on innovation of your core product.

Embedded integration makes the app that you are launching or updating much more attractive to your end-users, who do not have to step out of your UI to connect and work with their other favorite cloud apps.

Let's examine the typical trajectory with a real-life example. A leading technology company reacted to the challenge of the start-ups snapping at its heels by powering its own communication and collaboration tool with embedded integrations. Note that this company was large and experienced enough to build its own integrations, but it had to be agile, just like the start-ups that were beginning to erode its market share.

Through embedded integration, an iconic and long-established product became relevant again, its reach and flexibility greatly enhanced by the in-app integrations which end-users can add with just a single click.

The more third-party apps you can corral into your product, the greater the engagement of the end-user with your app. The point is, end-users want and need these integrations because the dizzying back and forth between applications has become unsustainable.

iPaaS vs. embedded iPaaS

Embedded integration is relatively new to the market. The approach is sometimes referred to as "embedded iPaaS," logically enough as it allows vendors to embed iPaaS capabilities within their applications.

All the same, it is somewhat misleading to view "embedded iPaaS" as an adaptation or a variant of "regular iPaaS," or iPaaS with extra mustard. With iPaaS you are orchestrating connectivity and workflows between apps, whereas "embedded iPaaS" expands connectivity within a single application.

A more interesting way to look at embedded integration is to define it as an integration where the backend is decoupled from the UI—and here, of course, there are functional and market parallels with headless software.

Headless CMS and headless e-commerce are increasingly popular because they raise the bar for flexibility and speed-to-market. Headless CMS architecture future-proofs a website build because the site can be redesigned or refreshed without having to re-implement the entire CMS. For e-commerce, headless finally fulfills the promise of omnichannel because it allows B2C to repurpose content across its website(s), mobile app(s), Alexa skill, kiosk, and more via APIs.

Headless gives the market what it wants and needs the most: speed, flexibility, multichannel interoperability – and does so in a way that is at once elegant and unexpected. This is why headless has proved so transformative. We believe embedded integration does the same for SaaS vendors across a wide range of sectors and we see similar potential for our solution in this space, webMethods.io Embed.

Introducing webMethods.io Embed

webMethods.io Embed is a cloud-based embedded integration engine that makes it much easier for vendors to give end-users the harmonious integration with third-party apps they are demanding. With webMethods.io Embed vendors reinvent existing apps or increase adoption of new products by empowering end-users to interact with go-to apps without having to go anywhere, but work uninterrupted from within your application experience.

With webMethods.io Embed, vendors embed productivity within their apps.

The new or reinvented app finds its way to market very quickly because most of the heavy lifting—and by that we mean the code—is done by webMethods.io Embed which also takes the maintenance and infrastructure of the integrations off your hands. This frees up time and creativity for engineers to do what they are passionate about (and what you hired them to do)—build the products of the future.

Conclusion

Embedded integration is a much more transformative development than the name might suggest. It resolves what you might call the standoff between SaaS vendors who need to innovate rapidly to stay alive and end-users who want their apps to be productive, not hold them back. Integration is key to this, but neither vendors nor end-users want to be distracted by time-consuming integration projects.

For SaaS developers, embedded integration means that they can plug, brand and customize their connectivity with third-party apps without having to write the entirety of the code or maintain the integrations going forward. This puts them well ahead of rival innovators who are building integrations from scratch—and who, once they are built, have to commit resources to hosting and updating them.

There may be good reasons to self-build, but these are of no concern to the end-users, who are moving to the cloud to be more productive, not to waste hours navigating between apps. SaaS products that do not address this issue of productivity will wither on the vine and become another churn statistic overnight.

Embedded integration is a breakthrough in its infancy, but one that will soon make an impact because it is so smart and hits a sweet spot for both vendors and end-users, whether they be Fortune 500 enterprises, B2C or startups looking for the quickest route to market.

When a problem integrates with a solution, transformation happens. And with webMethods.io Embed, we expect to be at the forefront of this transformation.



**Get more
information**

and request a demo:
www.SoftwareAG.com/embed

ABOUT SOFTWARE AG

Software AG began its journey in 1969, the year that technology helped put a man on the moon and the software industry was born. Today our infrastructure software makes a world of living connections possible. Every day, millions of lives around the world are connected by our technologies. A fluid flow of data fuels hybrid integration and the Industrial Internet of Things. By connecting applications on the ground and in cloud, businesses, governments and humanity can instantly see opportunities, make decisions and act immediately. Software AG connects the world to keep it living and thriving. For more information, visit www.softwareag.com.

© 2020 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

2020_05_WP_webmethods.io_Embed_SaaS_Vendors_EN