SACRA

EXPERT INTERVIEW

UPDATED
08/17/2023

Sara Du, co-founder and CEO of Alloy, on iPaas vs. universal APIs

TEAM

Jan-Erik Asplund
Co-Founder
jan@sacra.com

DISCLAIMERS

This report is for information purposes only and is not to be used or considered as an offer or the solicitation of an offer to sell or to buy or subscribe for securities or other financial instruments. Nothing in this report constitutes investment, legal, accounting or tax advice or a representation that any investment or strategy is suitable or appropriate to your individual circumstances or otherwise constitutes a personal trade recommendation to you.

This research report has been prepared solely by Sacra and should not be considered a product of any person or entity that makes such report available, if any.

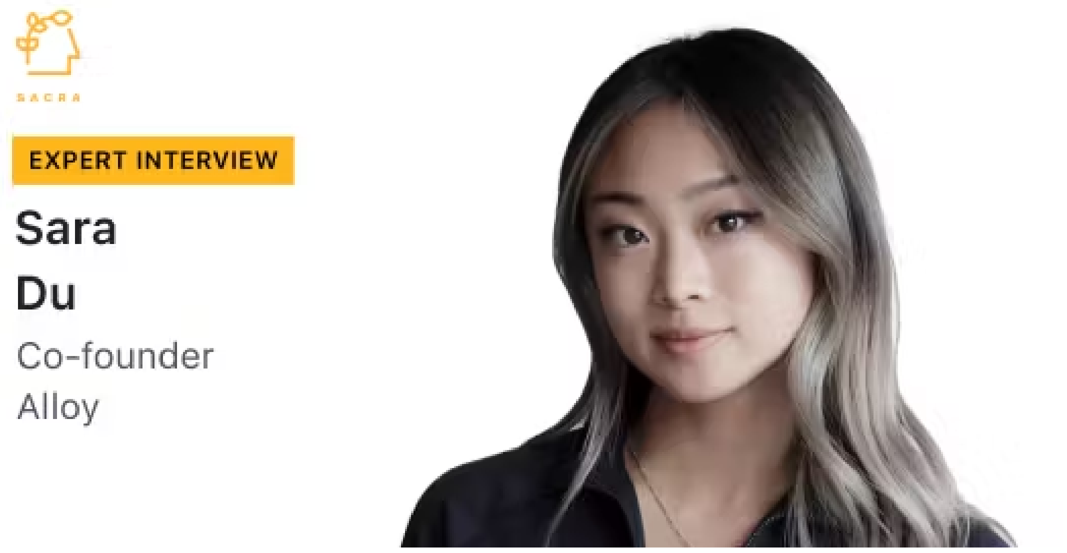
Information and opinions presented in the sections of the report were obtained or derived from sources Sacra believes are reliable, but Sacra makes no representation as to their accuracy or completeness. Past performance should not be taken as an indication or guarantee of future performance, and no representation or warranty, express or implied, is made regarding future performance. Information, opinions and estimates contained in this report reflect a determination at its original date of publication by Sacra and are subject to change without notice.

Sacra accepts no liability for loss arising from the use of the material presented in this report, except that this exclusion of liability does not apply to the extent that liability arises under specific statutes or regulations applicable to Sacra. Sacra may have issued, and may in the future issue, other reports that are inconsistent with, and reach different conclusions from, the information presented in this report. Those reports reflect different assumptions, views and analytical methods of the analysts who prepared them and Sacra is under no obligation to ensure that such other reports are brought to the attention of any recipient of this report.

Published on **Aug 17th, 2023**

Sara Du, co-founder and CEO of Alloy, on iPaas vs. universal APIs

By **Jan-Erik Asplund**



Background

Sara Du is co-founder and CEO of Alloy, an integration platform as a service (iPaaS) that consists of a consumer-facing, no-code product, a developer API, and an embedded solution all in one. We talked to Sara to make sense of the increasingly complex integrations landscape—between iPaaS, embedded iPaaS, universal APIs and native integrations—how AI is changing the world of integrations, and the future of custom software.

Interview

You're a B2B SaaS company in 2023. How do you build integrations and how has that process changed over the last 5 years? What will it look like over the next 5 years?

We've now at a point where, to play in an ecosystem you have to have at least a dozen key integrations. For example, pretty much any sales software is going to have to integrate with not just Salesforce and HubSpot, but a suite of related applications.

Along with the explosion of vertical SaaS companies, we've seen that people building those integrations in-house are slowly starting to realize it's becoming unmanageable. Whether it's through Alloy Automation or another embedded iPaaS vendor, we think that more and more people will be buying solutions for integrations versus building them in-house.

There's an analogy to servers here. Back in the day, people who were building applications had to build out and maintain their own servers. Then the cloud came about, and the market realized that buying AWS would just be more efficient. It's important to move quickly as an organization and keep up with new tech trends, and that often means giving up on doing some stuff in house.

The same thing is going to happen with integrations. Buyers of software are not going to think about what integrates with what, they will just expect that from their vendors.

Even recently, I bought Outreach.io instead of Salesloft because they had a HubSpot integration. It'll be nice as a buyer to not have to think about those things in the future.

How do customers decide between buying iPaaS, embedded iPaaS, universal APIs and native integrations? How do we think about and make sense of these products?

For the most part, iPaaS is for internal business process automation, and automating workflows across the apps you use internally.

Embedded iPaaS is for software companies because you're basically embedding the iPaaS so that you can release configurable user-facing integrations.

Universal API is a subset of that functionality, but more efficient in some ways for building quick integrations, because you can read from and write to multiple sources at once. However, there's little to no end user configurability. You basically provide a schema so it's faster to do these integrations, but it really only covers 20-30% of functionality in integration. Those are the key differences with embedded iPaaS.

The way to think about it is that a lot of companies like to start with universal API, because universal API defines a standard schema for data across multiple integrations. It makes it really fast for you to stand up light integrations to, say, five different ticketing platforms.

The issue is it's mostly for read and write integrations, and you can do some logic stuff in your own backend, but you'll get to a point where your end users need to do some configuration.

If they have a CRM with custom fields, most of those are not captured in the universal API schema. That's when we see a lot of companies essentially upgrade onto embedded iPaaS. Universal APIs are great for straightforward integrations.

Salesforce's acquisition of MuleSoft was one of its most successful, while Piesync was a flop for HubSpot. How do you think about what made the difference there and the role a MuleSoft or Piesync plays in an enterprise integrations ecosystem? What lessons can we take away?

It's interesting because every major ecosystem wants its own integration accelerant. Salesforce bought MuleSoft so it could manage both its existing ecosystem and expand its ecosystem with net new integrations.

PieSync, as the name suggests, is really focused on syncing use cases. Syncing use cases don’t handle a lot of the complexity of more enterprise integrations, especially as HubSpot moved in that direction.

The reason HubSpot shut Piesync down is probably because this tool didn't actually have the workflow engine behind the scenes. MuleSoft is actually very powerful because you can spin up internal APIs, add in custom third party APIs, do integrations with their pre-built connectors, and then stitch them all together. Salesforce integrated it pretty well into its ecosystems, so I think it was one of their more successful acquisitions as well.

App stores, like the Shopify app store, incentivize developers to integrate with those platforms in exchange for access to customers. What's the interplay between that and iPaaS in an enterprise integrations ecosystem?

Companies with app store marketplaces always start out with the goal of letting their partners build into them because they don't have enough capacity internally to fulfill the integrations their users want. That's great—but you go 4-5 years down the line, and the integrations fall off a cliff in terms of quality. They're poorly maintained, or at least not at the same bar as the ones built internally by the company. There's a lot more to an integration than meets the eye, and they need to be updated constantly to fulfill user requests.

That's an issue that a lot of the larger players that start building ecosystems start to face. Embedded iPaaS is the solution, if the company cares about controlling integration quality and being able to quickly release updates.

For example, I'm a user of Outreach. I try to use the Outreach and HubSpot integration, and it's unclear who maintains that. There's also this weird politics thing, where the Outreach integration is on the HubSpot marketplace and it has terrible reviews, while the HubSpot integration for Outreach is only in Outreach's support article center. I tried to reach out to both teams, and no one seemed to be taking responsibility for fixing it.

At some point, a lot of companies will do a revenue analysis and realize, "Oh, we are losing a lot because of these broken integrations, and we finally want to own them ourselves."

Tell us about Alloy's move into iPaaS and what advantages Alloy has as a result of having gone to market with a vertical-specific, consumer facing product?

It was good for us in the early days because of sales and marketing efficiency, targeting keywords, all these things that you do for user acquisition, it helps to be verticalized. Then, it also helped us build partnerships and trust more quickly than if we took a more scattered approach. That was really good for us.

We got to a point where we felt a bit pigeonholed by the commerce-focused brand though. So undoing that commerce-only messaging was a huge move for us recently. Because, when we started we built a generic integration and workflow engine infrastructure. Now, we really want people to know that they can use us regardless of if they're in commerce.

What did the process of building integrations look like before Alloy and what does it look like now after?

Alloy drastically simplifies the process of building integrations. We have a visual development tool that gets you to 80% of the integration. A lot of integration building is the data mapping between your internal API and systems and these third-party apps, and then there's also some of the logic, and then also the authentication part of integrations. It's just a lot of work to build, to handle OAuth, to handle those redirects, API key, whatever that third-party app requires. It takes a lot to build out the infrastructure.

We completely abstract that part away, making it really easy for PMs and developers to collaborate on the logic of the data mapping. What we don't do is the front end and the user experience—we let our customers basically define and make it look completely native. That's why I say 80% no code and visual because the 20%, I think, that's what a lot of the SaaS companies want to have control over.

Being developer-first is a big trend, how do you think about what that means and how to differentiate being developer-first?

Really it’s all about thorough documentation and control over edge cases. As a developer myself, I don't want to buy something unless I feel like when things go bad, I can have those failure handlers.

Ultimately, we have a lot of different layers of abstraction. We basically help you choose which one you're going to go down, especially the ones who want to get all the way down to the bare metal, they can do that with this tool.

How do you position Alloy with Zapier, Workato and Tray.io in iPaaS and embedded iPaaS? What's the opportunity that you see that a Workato or Tray.io isn't meeting in embedded iPaaS?

Since Zapier doesn't have a real, true embedded iPaaS option, companies who may have referred their users to Zapier are coming to us because they want to own integrations. Then with the others that have embedded iPaaS, our approach has always been to be just more developer-first. With our

integration set, we've been pretty focused on more enterprise integrations, so we're not for everyone. We’re pretty targeted about our SEO and acquisitions. It depends on the use case, but that's how we'll usually win.
I think a lot of it is still the personas we speak to, and then also the experience. It's hard to communicate what the product difference is unless you use it and experience the agility of building with our embedded iPaaS. We try to make it as fast as possible to roll out integrations. We've always optimized for the time between sales and partnerships getting integration requests to product and engineering being able to deliver on the business logic and frontend needs of those integrations.

Why does it make sense for a company like Alloy to have a consumer-facing product, developer API and embedded solution all in 1 company, versus solely focusing on one as a Zapier does?

For us, they’re all built on the same infrastructure, so we knew that we didn’t have to turn one thing off to turn the other thing on, or split our product team up.

We've been focused on selling to software companies as our customers, and oftentimes they're first buying our embedded solution. Sometimes, those very enterprise customers ultimately need full workflow control instead of using integrations natively inside a platform. They can actually spin out a user account from Embedded into our business process automation product—Flow. This is the benefit of all of our products being connected. Although technically, you can't go from Flow to Embedded, but you can go from Embedded and spin things out into Flow.

How do you think about horizontal versus vertical universal API companies, specifically as Alloy started out verticalized for ecommerce?

I feel like there's a really good advantage to being vertical which is that you have the industry context and knowledge to do schemas.

With universal API, the schema is actually a very opinionated model of data. Different universal API companies might do it slightly differently. Different parts of the industry might think of, say, orders and line items inside orders as different. Or

fulfillment objects and delivery objects, all this data is pretty subjective. Being vertical gives you the advantage of being close to the industry; as people's perspectives change, the schema might need to change as well. It's critical to be on top of that.
I think the disadvantage is, with the vertical approach, you get boxed in. Where, if you say you have a commerce universal API, basically, commerce APIs have a lot of use cases in fintech and other areas. If you say commerce, people just automatically disqualify themselves. But industries don't really live in isolation, and there's a lot of cross-pollination that can happen between industries. So being horizontal helps build a flywheel.

There's been a recent boom in universal API companies. In addition to Merge, Finch and Alloy, we have Rutter, Vessel, Recall and a number of others. How do you explain the boom, why is it an attractive market to be in and how do you see it shaking out?

Right now, there were a lot of fundraising announcements around a similar time, so that built the hype. I think that hype is temporary.

There's definitely a market that’s growing because of this build vs. buy climate shift, because as there's more vertical SaaS, people realize they don't want to do this in-house, so there is more of a buy mindset that's happening.

I think there's a couple incumbents, but there's no clear winner. There can be winners that coexist in different parts of the market, but ultimately, I think there's a lot of startups that tackle something super niche that probably just won't exist because they'll be supplanted by more universal platforms.

There are some companies that just shouldn't be VC-backed and be very vertical APIs. They should still exist and they make a lot of money, which is great for certain industries. But with horizontal companies the advantage is scale. Eventually, the people translating docs into JSON or whatever format these companies use, it's just going to be LLMs. Then, the more data you have, the easier it is to generate integrations. At one point, it won't really be about prowess and building out integrations, more around functionality, platform functionality.

How do you see the rise of LLMs changing how SaaS companies think about and build integrations?

With AI, one of the biggest things is that you can generate API endpoints much more quickly. A partner or customer can be like “Hey, Shopify, you don't have this endpoint,” and they can just spin it up super fast.

They can use some kind of LLM to translate that business need and then request the right resources or, in the code base, put them together and expose a public endpoint.

Ultimately, I think there's going to be a lot more API endpoints because everyone has these existing code bases and it’s just a matter of chopping it up in the right ways for functionality to be used by external parties.

It's going to be tough, because you'll have to manage all these endpoints. Agents will have to learn how to navigate iPaaS and all these universal API tools like limbs to the brain of LLM. On the other end of that, there's going to be more opportunities for agents trained on these types of tools.

AI plus iPaaS enables AI to take action in business apps and has the promise of automating a huge number of everyday tasks. How will this affect RPA and how do you anticipate RPA evolving?

They will continue to be that. They'll just have to add in functionality to observe and, basically, replicate behaviors, just like Adept. I’m on the iPaaS side and the way I see it is they're two sides of the same coin. It's like RPA, it's just an imitation thing. But for further accuracy, you need iPaaS and accessing the machinery inside. I think they'll keep coexisting, and hopefully players like UiPath can add in AI functionality.

How does the proliferation of iPaaS like Alloy affect CDP companies like Segment, ETL companies like Fivetran and reverse ETL companies like Census and Hightouch, aka third parties wholly dedicated to helping data move between apps?

ETL and data stack companies are selling to a very different persona—the data and the marketing team.

Embedded iPaaS, on the other hand, is for engineers and product teams, because it's for the user-facing configurable integrations that are in the product. It's two very different form factors or use cases of integrations.

A lot of CFOs or people who look from the top-down are like, “Why do we have two buckets of integration spend?”

Ultimately, long-term and currently, a lot of ETL tools and iPaaS tools are jointly educating the market on what our differences are and why we shouldn't be bucketed together.

Here's an analogy: There are all these different types of email tools. A sales team uses outbound tools like Outreach, marketing teams use campaign tools like Customer.io and HubSpot.

At this point, people don't question the fact that there are multiple buckets of email tool spend, and there's also the backend, a code-driven email APIs. I think the same should happen with ETL versus embedded iPaaS, because they’re very different functions and sit in very different parts of the organization.

What does the future of custom software look like with AI and how is Alloy positioning itself to enable that over a 5 year time frame?

There's definitely going to be a lot more vertical software in whatever shape or form. AI and LLMs decreasing the barrier to entry is going to create so much more software for super niche use cases. And I think it fits into that whole theme of, there's going to just be more need for managing all these, the APIs and the SaaS. So to the extent that we can stay on top of it as iPaaS, I think it'll be good for us for there to be more software in the world.

Disclaimers

This transcript is for information purposes only and does not constitute advice of any type or trade recommendation and should not form the basis of any investment decision. Sacra accepts no liability for the transcript or for any errors, omissions or inaccuracies in respect of it. The views of the *experts expressed in the transcript are those of the experts and they are not endorsed by, nor do they represent the opinion of Sacra. Sacra reserves all copyright, intellectual property rights in the transcript. Any modification, copying, displaying, distributing, transmitting, publishing, licensing, creating derivative works from, or selling any transcript is strictly prohibited.*