

CHOOSING THE RIGHT CUSTOMER ID TO MAKE GROWTH FASTER

One of the most dangerously overlooked aspects of activating customers is choosing the right customer identifier. We've seen this simple decision cause so many headaches that it is worth having its own section here.

Steve Blank (well-known investor and our former instructor and professor at UC Berkeley) famously said that a startup is an organization searching for a repeatable business model. With an emphasis on "searching"—meaning that startups should optimize for opinionated flexibility. In service of this ethos, a lot of the scale advice for startups in the last five years has been admonishment to do some things manually and defer scale until they find that optimum product-market fit—you nail it and then you scale it.

The downside is that you may make decisions early on that make scaling extremely difficult. We know many fast-growing companies who made bad architectural decisions that haunted them for years. And for some, those foundational mistakes took them off the chess board.

One such problem we have seen happen in recurring fashion, across multiple startups now, is the choice of a primary identifying key for your customers. A lot of companies use email as the primary identifier for customers. Thus customers use email to log in and identify themselves. In many cases, they can only use one email identifier at a time, which serves as their database ID. Examples include Atlassian, Amazon, and many, many other business tools you use and love.

Don't do it. We don't care how common it is or how many startup bootstrap frameworks work like this. Using email as an identifier will hamstring you and cause massive headaches down the line. We have seen multiple massive, world-renowned companies spend tens of millions of dollars just to re-engineer their identifier architecture. Eventually, every company needs more flexibility in how they organize customers and the identity framework underpins everything.

In B2B SaaS specifically, a lot of customers use their company email as an ID. As they move to different companies, they take good tools with them. But because company email is the ID, they have to ditch their old account and sign up with a new email at the next job. As a result, the customer loses their history and person-

alizations of the software. The product company also loses a way to connect and communicate with the customer and influence their retention during this delicate interregnum where they could always choose a competitor.

The other problematic pattern we have seen is that in general, people have multiple email addresses and numbers. Some of these IDs are backed with real data that can and should be connected to your software to deliver a better service. Accepting only one of these as an ID can be suboptimal for a software company and its products.

You should design customer ID systems to maximize these and other opportunities for your software and your product funnel. Here's our suggestion:

The best way to construct an ID system for your customers is by using a unique unguessable random number¹⁴. We generally call these globally unique IDs or GUIDs. Then you can attach all kinds of other identifiers to a customer's GUID to create a flexible system of identification. Multiple emails, phone numbers, and even other GUIDs can be associated with *and* dissociated from the primary GUID. Your customer doesn't even need to know their GUID or interact with it in any way—they can just use any of the associated IDs to log in and configure the software to their liking.

If your customer joins a different company, they can simply add their new email and jettison the old one. It makes no difference to your identifier system because their GUID does not change. Your customer doesn't need to provide an email at all—they can use a username, phone number, PIN, or any other identifier. It's better for the customer and better for your company.

Here is what it could look like (among other possible schemes):

GUID (primary identifier)	
Email #1	can login
Phone #1	can login
Email #2	cannot login
Phone #2	cannot login
GUID #2	cannot login
Name (Fname, Lname)	can login
Address	cannot login

In this example, you have two emails connected to an account, as well as a phone, simple username, etc. One email¹⁵ can be used to log in, as well as the phone number. The benefit of this setup is this: If the product uses calendar or email data (or can send email on behalf of the customer), your system can use the data represented by any of the email IDs to deliver on the customer promise—even though the customer uses only one of those to log in. For example, I can send email from

multiple Gmail addresses within the same account, even though I can only log in with one of the emails.

This is not rocket science, but I still see so many companies make this error. Don't be one of them.

Customer onboarding and activation is a game of trade-offs. It's a balance of personalization and opinionated UX; of simplicity and power; time-to-value and maximum value. It's a never-ending process that the best product teams return to again and again as they learn more about their target customers and how they use the product.

Once a customer is activated, the fight for their dollar has just begun. Next is how product-led companies think about converting customers into paying fans.

[11](#) Read the effing manual.

[12](#) Very common during the CIO buying era.

[13](#) Not forever, just until the customer is deeper in the actual relevant activation flow.

[14](#) Unguessable by machines/computers, not just humans.

[15](#) As a matter of good security, using multiple *email addresses* to authenticate and authorize may be a bad idea because one may be compromised when you're not using it regularly. Best to use one of each to authorize at a time. And the customer can decide which in their security settings.