



Case Study



Twilio turns to Split as a key component of their experimentation capability to drive faster product decisions

Twilio started using experimentation as a way to quantitatively improve product decision-making, to make decisions quickly as well as drive key metrics like sales opportunities and product adoption. As Twilio looked to expand experimentation enterprise-wide, and to more complex use cases, they turned to Split to solve granular targeting and providing necessary data around what variation of a feature a user experienced. Split seamlessly plugged into Twilio's internal employee dashboard, making it easy to adopt experimentation across the engineering team.

“Split’s full-stack experimentation platform has been built with engineering and product teams in mind. Its robust architecture and rich feature set integrates into our internal platforms and helps power experimentation across our entire engineering organization.”

Laura Schaffer, Product Manager

Situation

Over nearly a decade, Twilio has been one of the fastest growing software companies in the world. One of Twilio's strengths has been a highly innovative engineering team. To be an innovator at the forefront of building API services for developers, Twilio invested in an engineering team that builds many tools in-house, often before a product even exists to address that need within their stack.

In 2016, Twilio began to see the need for richer product experimentation. Use cases were on the rise in many key areas of the codebase. Typical use cases included:

- **High-risk releases** - Limit the exposure of a new feature that has the potential for significant impact. For example, changes to the customer sign-up flow could directly impact sign-ups.
- **Assessing the value of a change** - Quantify a change being made, perhaps a new feature, that has a specific goal of impacting a metric. For example, if a backend code change is intended to improve API response time by a certain amount, the goal would be to see if an attempt at doing so leads to the expected outcome.
- **Understanding the impact across segments** - Gain a deeper understanding of user behavior, by testing a feature across multiple users or customer segments. This can be done per customer account, or per individual user. The segments are typically different use cases including different geographies, different code language, and level of interest in the product.

At the time, Twilio began to build out the infrastructure needed to run experiments at scale. This included a cloud data warehouse, a data visualization tool, an in-house built data analytics tool and an in-house internal reporting dashboard. As Laura states, “We were lucky that Twilio, having a really innovative engineering team, had a lot of components needed for an experimentation platform already built in-house.”

With some initial successes of Twilio's efforts on the first few use cases, and a full team was formed with a product manager, software engineers and a data scientist to build an experimentation platform at scale that would be used across all of engineering. To scale their efforts they needed a robust system that could bucketize and randomize user assignments to experiment treatments, record what treatment of a feature a user experienced and pipe that event data to Twilio's data warehouse.

Choosing Split

As with many internal engineering needs, Twilio evaluated building a targeting engine in-house and reviewed a range of both open source and off-the-shelf tools. As part of this process, they discovered Split. Twilio had some unique requirements but found that Split had the capabilities and flexibility to meet their needs.

Split was built from the ground up for an engineering team use case. Split's Feature Experimentation Platform comes with SDKs for eleven different languages. However, what was more important, was that Twilio needed different teams to organize their experiments in Splits within one environment.

As Laura Schaffer, Product Manager at Twilio states, "It was clear that Split was built with the developer in mind. We felt that the Split team really understood our environment and the product aligned with our requirements."

Twilio also wanted to build a 3rd party targeting engine into their own internal dashboards and tools. This would serve two purposes: it insulates them from future changes in underlying platforms, and it enables their employees to have a single user experience. The Split API gave Twilio the flexibility their engineers needed to build the right experience.

Result

Building on Split enabled Twilio's experimentation team to move fast. With sophisticated feature flagging and targeting from Split, Twilio was able to have the full platform up and running six months faster than originally planned, and be able to run precisely targeted experiments.

With the platform up and running to its full capability, the experimentation team was able to support experiments on some of Twilio's toughest internal debates. One example was an internal debate around adding additional questions to the user sign-up flow. The sales team wanted additional questions to better understand the customer needs and environment before reaching out to them.

However, the marketing team was concerned that additional questions could irritate customers trying to sign up and use Twilio.

After one month of running an experiment, Twilio saw that the treatment they were testing (i.e. additional questions) was clearly outperforming the control. This indicated clearly to the team that the extra questions were not harming the signup experience, and were, in fact, making the overall experience better for potential customers.

This was a big win for the company overall, and a clear validation of the value of the experimentation Split helped power.

Future Plans

Going forward Twilio is looking at additional experimentation use cases. One test they are planning is around making tweaks to SMS messages sent for user verification using their two-factor authentication product. The goal would be to increase verification success rates.

Through the work of the experimentation platform team, full-stack experimentation is now an integral part of Twilio's continuous delivery software development process.