



KAHUNA MARKETPLACES: BEHIND THE SCENES

By: Jacob Taylor | October 29, 2018 | Kahuna

Kahuna is a Hawaiian term taken to mean a wise person. In the context of software, Kahuna is a SaaS solution for digital marketplace operators that empowers them to engage with their buyers and sellers on a one-to-one level. As a company, Kahuna is a combination of the two. We're a trusted partner for many companies across the globe, many of which are industry leaders in their respective markets.

Kahuna originally started as a marketing automation provider that enabled companies to engage with users via personalized communications. Rather than simply sending emails and push notifications en-masse, Kahuna customers are able to leverage our machine learning optimizations to send communications when users were most likely to engage with them.

Since then, the Kahuna team decided to [focus on marketplaces](#) and created features to [optimize marketplace operations](#). This focus led us to create a [first of its kind industry report](#) about the state and future of the digital marketplace landscape.

While I could go on about Kahuna at a high level, that's not the purpose of this article. Instead, I want to provide some insights into the inner workings of our push and email messaging system—a core component of our solution.

Enabling one-to-one communications, at scale

Our primary objective at Kahuna is to help marketplaces have a personal conversation with their buyers and sellers, regardless of size, be it 10 users or a billion users. In order

to achieve this level of scalability, Kahuna is built on [Google Cloud Platform](#) (GCP) and is engineered to scale horizontally by having redundancy and no hotspots.

In the context of processing data for push/email communications, hotspots are situations where many parallel activities focus in on a small number of records. With hotspots, your application will perform only as well as your worst hotspot.

Kahuna was built to work on huge user bases that require horizontal scaling. For example, a web application might have a traffic spike of 10,000 concurrent users when the application is built to handle only 1,000 users.

While you're probably saying the solution is to spin up more servers, with hotspots, that isn't an option. Without hotspots, you can elastically scale your servers as the need arises.

Without the right expertise, avoiding hotspots is a difficult challenge. As the Co-Founder of SugarCRM, preventing hotspots has been a focus of mine for awhile.

SugarCRM stood apart from other software solutions because it was built for high performance on commodity/stock hardware. On two Dell servers, we were able to support 3,000 users with server response times in less than 1 second, over 99.8% of the time. SugarCRM represented a huge performance gain while using less than 0.4% of the server resources.

Fun fact: Although Kahuna started as a general marketing automation solution, a handful of our marketplace customers are using our solution as a CRM to manage buyer and seller engagement, but that's a topic for another article.

Going from zero to 80 million, in one day

Going back to the topic at hand, Kahuna is built with performance and reliability as top priorities. The solution is built to make it easy to detect and isolate issues. When issues are detected within the platform, our engineers can isolate them in a matter of minutes.

Downtime due to maintenance or resource consumption is never an issue for our customers since Kahuna can be scaled up or down instantly, and upgraded, while in production. Scaling a system to over a billion users is one of my favorite parts of Kahuna.

In order to prevent downtime from occurring, we leverage task queues within the platform. The task queues allow parts of the system to be worked on independently. If a component suffers from a problem, which is very rare, depending on the scope of the issue, queues or individual customers can be paused. We can verify that the requests are



well formed. Once the problem is resolved, Kahuna is able to quickly catch up to live traffic.

One of the biggest challenges to building out Kahuna was corrupted requests being sent from our SDKs. It appeared that HTTP requests were being corrupted between the collection in the SDK and the request being built.

The initial fix to this solution was fairly simple: We added a hash to detect corruption. Although effective, we still received a few corrupted HTTP requests, mainly from low-cost Android phones. For context, even if these corrupted requests occurred once in every million requests, that would still amount to multiple issues per hour. We take data quality very seriously.

Building on our first attempt to fix the issue, we added a second hash—containing the contents of the first hash—to all HTTP requests processed through Kahuna. If the two hashes didn't match, a notice would be sent to the client pretending that the server had a problem and they should resend the data.

The system was asking for a resend a high percentage of the time. The other portion was requests saying the information was accepted. This configuration did two things:

1. It allowed the client to create multiple server requests.
2. It allowed for the possibility of corrupted data downstream, in which case, it would be told to delete the offending data.

After our first solution was built, we started tracking the amount of times these issues occurred. Seven months later, the mechanism had never asked the client to send data a third time; the second attempt always delivered good data.

When Kahuna was first starting out, we had one Saturday where our customers Yelp and Tumblr simultaneously went live on our platform. Between the two accounts, Kahuna had to onboard 80,000,000 new users that day. It went smoothly, as we didn't face any system alerts, data loss, or long server response times.

Kahuna's scalability also came into play on the day Robin Williams passed away. After Robin Williams' death, several of our major news outlet customers broadcast push notifications to all their users notifying them of the event, prompting them to click for more details. In the race to be the first to break the news, the push messages arrived before they could put an article together, resulting in users hitting refresh until the article loaded.

Having readers constantly hitting refresh, resulted in a 5x surge in global traffic processed through Kahuna. This event triggered some internal alerts, but Kahuna self-corrected itself within seconds, without any action required from our teams. Going to the point about rapid scaling, Kahuna was able to scale elastically to handle the load and rapidly decrease resource consumption once the traffic passed.

A holistic understanding of buyers and sellers within marketplaces

Kahuna was built to help customers of any size interact with their users. It starts with a welcome campaign that introduces a feature the user hasn't tried, encouraging a user to complete a process they started (checkout after filling the shopping cart), reaching out to users that have gone dormant, and so on.

In order to improve communications between businesses and their consumers, Kahuna creates a holistic view of each user as a person. The suite recognizes the fact that users aren't devices, phones, or forms. Rather, they're individuals that can be reached via a variety of channels. They have multiple devices and have different histories.

Kahuna collects submitted user information and stores it in the user's profile, which enables us to optimize the user experience (more on that in a minute). Kahuna tracks anonymous users. Once those anonymous users create an account with the provider, Kahuna is able to merge the anonymous data into the known profile.

Typical Kahuna integrations track 3 dimensions of data: behavioral, transactional, and situational. Behavioral events track things the user is doing. Transactional data tracks what people are purchasing, server-side updates, and product/service categories the user is interested in. Situational events tell us about the environment of the user and allow us to reach out to them based on where in the world they are located.

One of the most important capabilities we pride ourselves on at Kahuna is the fact that we're able to process user activities in real time. This prevents situations such as flawed cart abandonment campaigns, where a user is sent discounts for a product/service, even after they've made a purchase.

Once Kahuna has accepted a setup of events from one of our data ingestion entry points, Kahuna will update the user's profile within 4 seconds. These data ingestion capabilities were designed to not require details about how messages are formatted. Our data ingestion engine also scales well. In testing, I've hit it with tens of thousands of requests per second for over 20 minutes straight and didn't face any issues.

Another primary benefit of Kahuna and its SDK is our ability to cache events and forward events/attributes where appropriate. For example, when users create campaigns to reduce cart abandonment, Kahuna is aware that you are waiting for an `add_to_cart` event, with an objective of triggering `purchase_complete`.



As we know those two events are important to you, we'll expedite the `add_to_cart` function. Similarly we'll send the `purchase_complete` event the instant a purchase is complete. This ensures Kahuna is able to determine when the user would normally have transacted and optimizes outside of that window using a plethora of optimizations. We are seeing significant lift from cart abandonment campaigns. If a purchase is made, as mentioned earlier, we'll block the campaign from reaching out to that user.

Remote management of the Kahuna SDK settings takes place in the background and is implemented to allow us to send data to Kahuna as efficiently as possible while still keeping your campaigns accurate and profiles up to date.

As mentioned earlier, Kahuna is undergoing significant shifts. What started as a marketing automation solution for application developers has since evolved into a comprehensive suite that enables marketplace operators to gain an unfair advantage over their competitors. This advantage comes in a variety of forms.

Going beyond push and email communications

Kahuna can help in a variety of forms from [intelligently messaging](#) buyers and sellers, to monitoring the [health of their marketplace](#) in a central location, to creating [effective subject lines](#). Kahuna has come a long way from the days of delivering push notifications and emails when users were most likely to engage with them.

As a whole, Kahuna is the brains behind the digital communications efforts of brands and marketplaces across the globe. For marketplaces in particular, it's a core component of their [technology stack](#).

To learn more about how the marketplace business model can help improve your profitability, check out The Kahuna Blog. Simply click the button below!

The Kahuna Blog
Key Insights and Best Practices for Marketplace Marketers



SIGN UP TO RECEIVE THE BIG KAHUNA

A Bi-Weekly Newsletter Focused Exclusively on Online Marketplaces

*



Author: Jacob Taylor

Jacob Taylor is Head of Data Science and Co-Founder at Kahuna, where he's passionate about helping companies have meaningful conversations at scale. He believes brands should "Treat people as people, not numbers." Before starting Kahuna, Jacob worked on building high-performing scalable systems at Symantec, Octane Software, and Epiphany

(now Infor) and was the Co-Founding CTO of SugarCRM, helping drive it from an initial vision to millions of installations around the world. Jacob holds a Bachelor's in Computer Science and Engineering from University of California, Los Angeles and a Master's in Artificial Intelligence.



Fun fact:

Jacob loves increasing performance and decreasing costs of products. In fact, in just one year, while adding new features, Kahuna's traffic increased by 10x and the cost per billion events decreased by 10x!

[View all posts from this author](#)

RECENT POSTS

By: Charles Costa | December 3, 2018

Four Ways to Boost Marketplace Transaction Volumes During the Holiday Season and Beyond



By: Jeff Nolan | November 29, 2018

Car Next Door and the 2nd Law of Marketplaces – Trust is The Product

By: Adrien Nussenbaum | November 28, 2018

Understanding the Services Revolution



[About](#) [Careers](#) [Partners](#) [Press](#) [Contact Us](#) [Log In](#)



+1.844.465.2486 • hello@kahuna.com

© 2018 Kahuna All rights reserved

• [Legal](#) • [Privacy Policy](#) • [Documentation](#)