Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

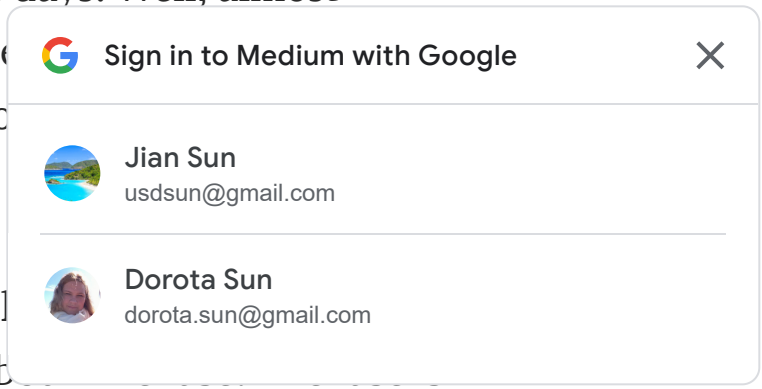
Dorota Sun
dorota.sun@gmail.com

What is Firebase? The complete story, abridged.

Doug Stevenson Follow
Sep 24, 2018 · 21 min read

Hello! I'm Firebase!

It seems like there's an app for everything these days. Well, almost everything. I haven't found an app that helps me
It happens to me sometimes, and it's super obnc
build it! I'll contribute to your Kickstarter.



If you're the enterprising sort of person that tacl
needs with a mobile app, you'll want to know ab
Google's mobile application development platform that helps you build, improve, and grow your app.

Here it is again in bigger letters, for impact:

> Firebase is Google's mobile application development platform that helps you build, improve, and grow your app.

And now you know what Firebase is. In theory, this blog post could be done!

Beyond the marketing copy

I have mixed feelings when someone asks me “What is Firebase?” (which happens a lot, since I work on it). On the one ha
interest! Thanks for asking! On the other hand,
do I even begin? The definition above is accurate
all.

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

I like retro games, so I’ll try to explain Firebase a

This makes a lot of sense — if you are me! But I understand that many of you are not me, so I'm sorry if you weren't helpe

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

All joking aside (for the remainder of this parag
toolset to "build, improve, and grow your app", a
cover a large portion of the services that develop
build themselves, but don't really want to build,
focusing on the app experience itself. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud, and scale with little to no effort on the part of the developer.

When I say "hosted in the cloud", I mean that the products have backend components that are fully maintained and operated by Google. Client SDKs provided by Firebase interact with these backend services *directly*, with no need to establish any middleware between your app and the service. So, if you're using one of the Firebase database options, you typically write code to query the database *in your client app*.

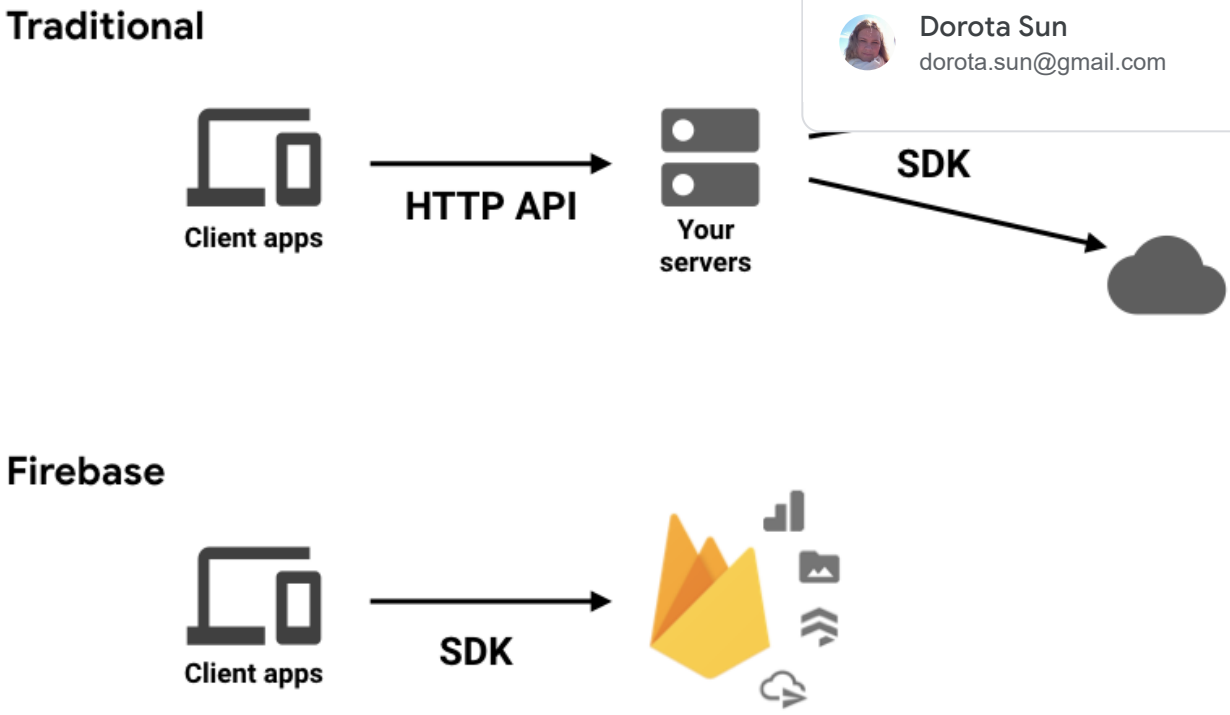
This is different than traditional app development, which typically involves writing *both* frontend and backend software. The frontend code just invokes API endpoints exposed by the backend, and the backend code actually does the work. However, with Firebase products, the traditional backend is

bypassed, putting the work into the client. Administrative access to each of these products is provided by the Firebase consc

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com



You can make your backend vanish!

If you identify as a “backend engineer”, you might be hearing this and thinking that your job is being eliminated! ***“OMG, no more backends — now I have to learn frontend development!”*** This isn’t really true, as there are some things that simply *ought* to be on the backend for a variety of

reasons. Firebase recognizes this, and offers a way to do some backend development, where it makes sense for the app
worry, your job is safe, and I'll talk more about t

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

Sorry if I gave you a scare, backend devs.

Because of the way Firebase products work, some people might call Firebase a "platform as a service" or a "backend

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

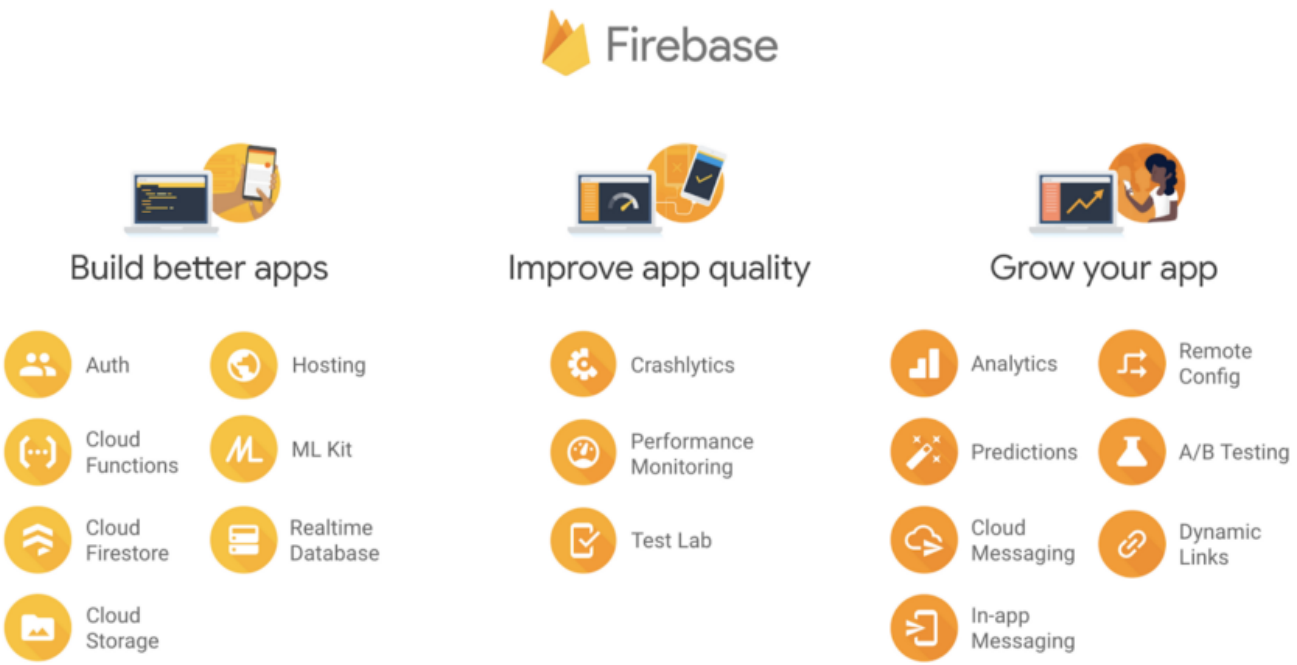
really felt comfortable wedging Firebase fully in

Firebase is Firebase. (I know, this statement obv

what Firebase is, which is supposed to be the pu

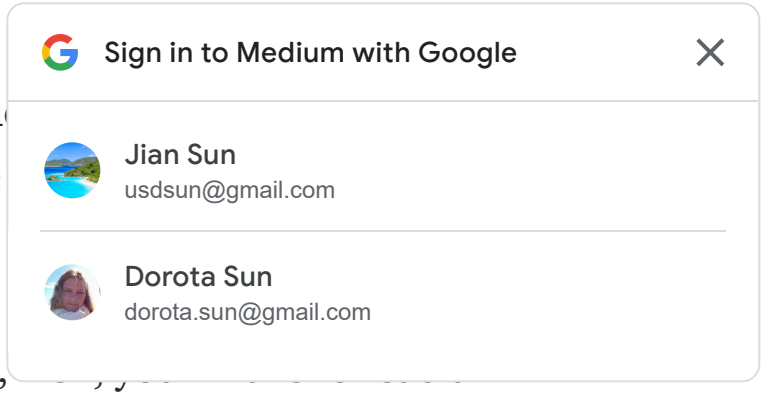
Anyway, at the time of this writing, I count 17 in

Firebase suite. Here's another helpful picture:



The Firebase suite. Sweet!

It's said that a picture is worth a thousand words. I counted 37 in the picture, including two things that are not really words. Guess I'm out of luck today, because I'm going to try to use up th words in this article to help explain that picture going to count them.



And if you want to find out what Firebase *is not*, the way to the end of this post. **No skipping ahead! You'll miss all the jokes!**

What sort of apps is Firebase good for?

There's really no limit to the *types* of apps that can be helped by Firebase products. There are only limits to the platforms it can be used on. iOS and Android are the primary targets for the Firebase SDKs, and there's increasing support for web, Flutter, Unity, and C++. You should also know there's an Admin SDK available for a variety of languages, to be used with any backend components you might require.

On top of those SDKs, there's a library called FirebaseUI (Android, iOS, web) that provides a bunch of helpful utilities to make development with Firebase even easier. And there are also projects such as AngularFire that

wrap the web SDKs for use with Angular. These are open source. Firebase likes open source.

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

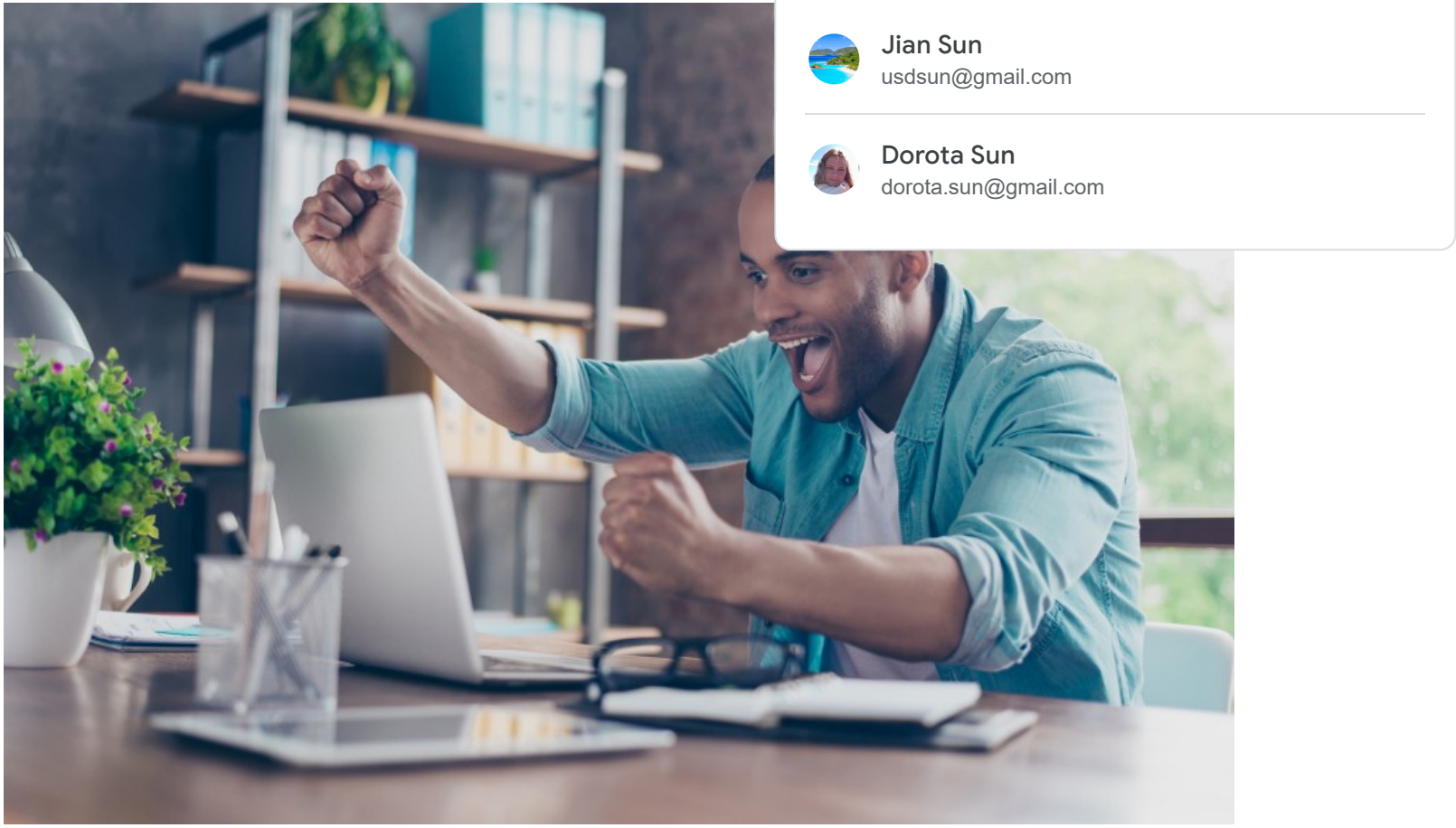
Dorota Sun
dorota.sun@gmail.com

Here are a couple examples of developers using

Greta builds mobile games with Unity:

This is Greta, and definitely not a stock photo of someone else.

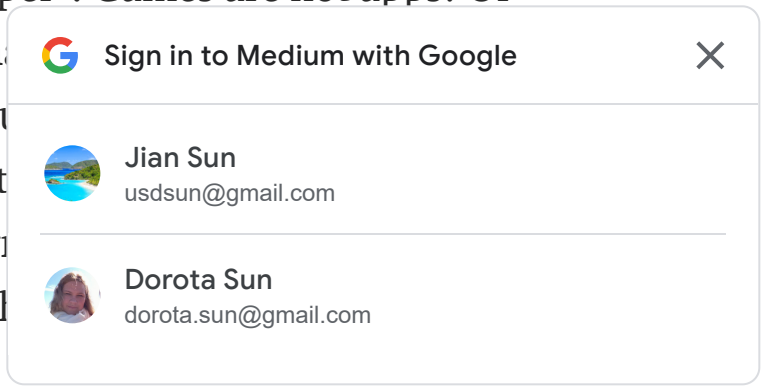
And Shawn is building a social networking app:



This is Shawn, also not a stock photo. These are totally real developers.

You can tell from the looks on their faces that they're *having a blast* building apps with Firebase.

Greta doesn't think of herself as an "app developer". Games are not apps! Or are they? I don't know. But games and tradition needs that could be met by Firebase. For fun, I ju apps with a highly customized UI, that happen t gamification strategy. Anyway, Greta and Shawn despite the fact they're building very different th



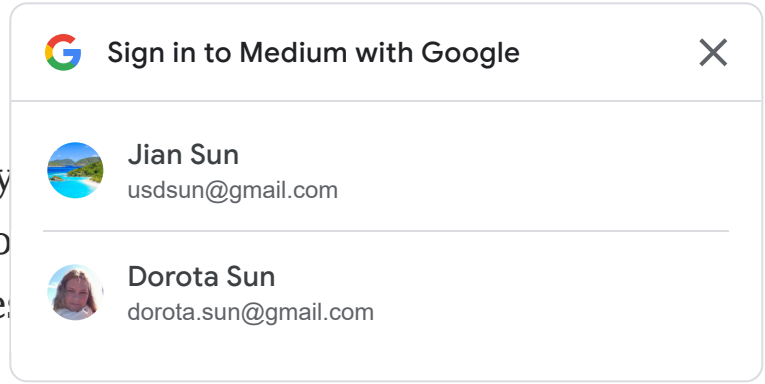
To get a sense of what Firebase products actually do in an app, I'll go through the individual products from that image above. You saw, in that image way up there, three main groups of products: "build", "improve", and "grow" (but these categorizations are not strict). I'll talk about the "build" group first, and give some specific cases where Greta and Shawn make use of each product.

Build your app — creating the "guts"

The "build" group of products are these:

Authentication — user login and identity
Realtime Database — realtime, cloud hosted, NoSQL database
Cloud Firestore — realtime, cloud hosted, NoSQL database
Cloud Storage — massively scalable file storage
Cloud Functions — "serverless", event driven backend

Firebase Hosting — global web hosting
ML Kit —SDK for common ML tasks

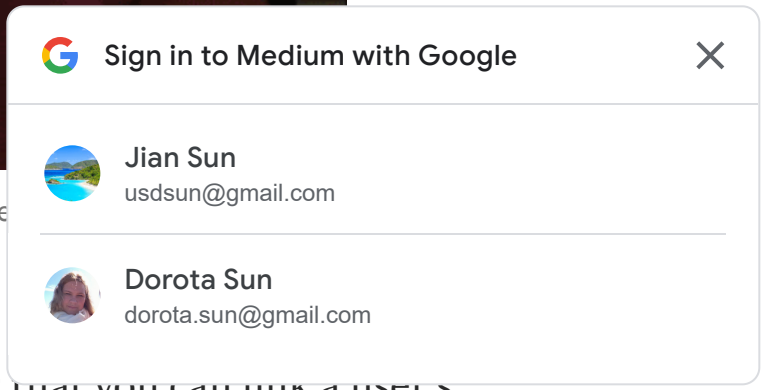


Firebase Authentication takes care of getting y
identified. This product is essential to getting so
configured properly, especially if you need to re
(which nearly every app will want to do).

What's special about Firebase Authentication is that it makes easy to perform secure logins, which is incredibly difficult to implement correctly on your own. And it's "federated", which is to say that the United Federation of Planets encourages its use. Here's what the Federation's Captain Picard thinks of implementing your own auth system:

Jean-Luc has experienced enough grief already — please don't make

password.



Others may say that "federated identity" means that you can link a user's accounts from the various identity providers (Facebook, Twitter, Google, GitHub) into a single account on Firebase Authentication. But I like my definition better.

In any event, I strongly recommend learning Authentication and integrating it into your app *first*, which will hopefully prompt you to think about the security of per-user data that you might store using some of the other "build" group products.

Firebase Realtime Database and **Cloud Firestore** provide database services. I listed them both as "realtime, cloud hosted, NoSQL databases". They have individual strengths and weaknesses, and you may need to do some research to figure out which one is better for your needs. Hint: start with Cloud Firestore, as it likely addresses more of your needs (and it's also

massively scalable). You can use either one, or both together, if that suits your app.

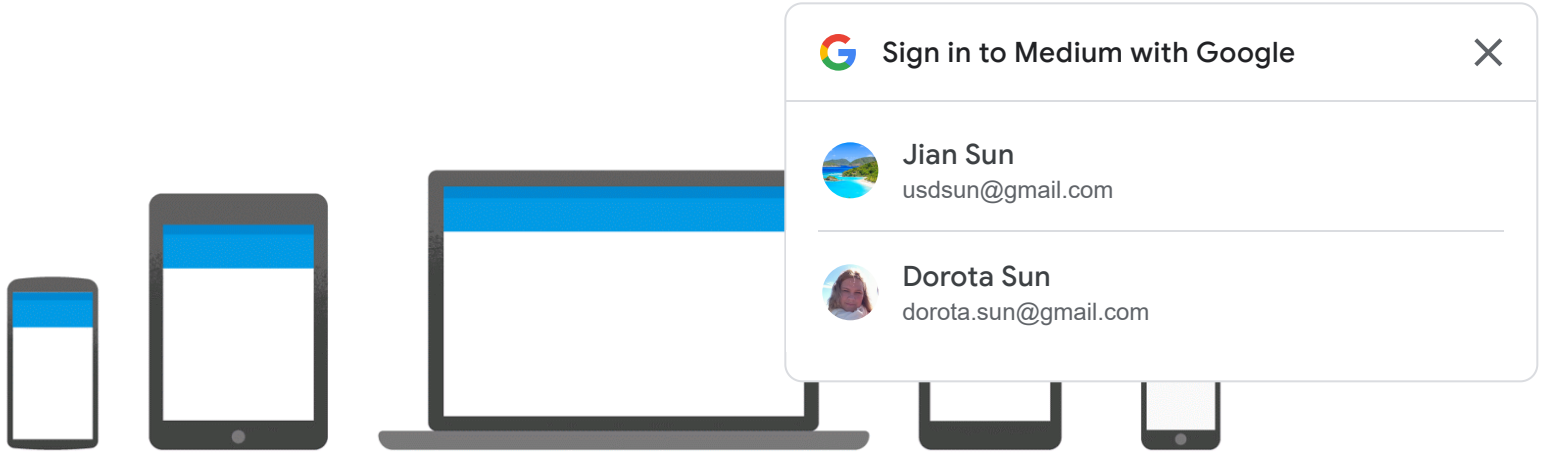
Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

It's worth noting that Firestore is technically a G
Firebase product. Why is it listed with Firebase?
in your mobile app to make direct data access p
for that pesky middleware component. There ar
with a similar relationship with Google Cloud, which I'll also note.

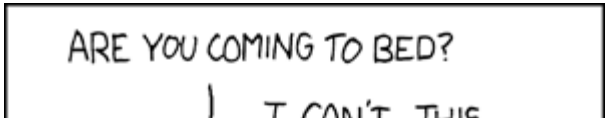
What's really special about these databases is that they give you "realtime" updates to data as it changes in the database. You use the client SDK to set up a "listener" at the location of the data your app wants to use, and the listener gets invoked with that data repeatedly, every time a change is observed. This lets you keep your app's display fresh, without having to poll the data of interest.

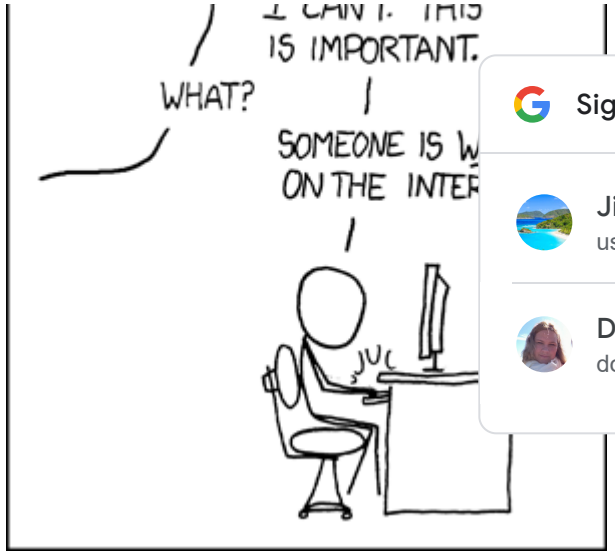


I'm paid a cash reward to put this GIF in every presentation where I mention "realtime data".

With realtime data like this, our friend Greta uses realtime data in her games to maintain live leaderboards of in-game events for everyone to see. Shawn uses them to provide messaging between friends on his social network. Because we just don't have enough chat apps out there!

Fun fact: Realtime Database was the original "Firebase" before it joined Google in 2014. To this day, people still colloquially (but incorrectly) refer to Realtime Database as just "Firebase". But you shouldn't do that, because it's wrong.



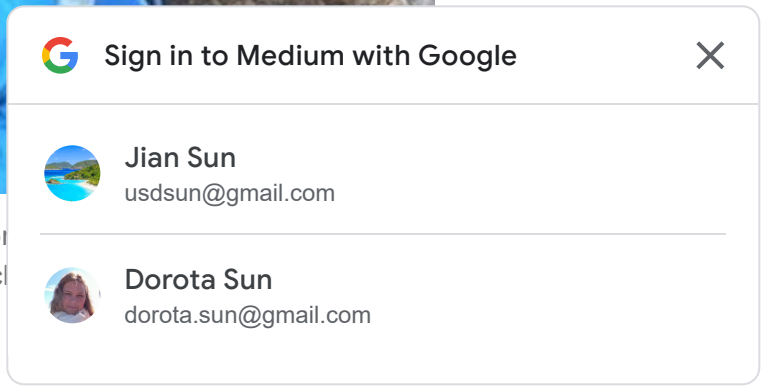


Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

You should always be right on the internet, and argue fiercely. Firebase is a **platform**, folks, not a database!
(view this comic on XKCD)

Cloud Storage provides massively scalable file storage. It's also technically a Google Cloud product, not a Firebase product. With Cloud Storage *for Firebase*, you get client SDKs to use in your app that enable you to upload and download files directly to and from your Cloud Storage "bucket".

This “lolrus” has a Cloud Storage “bucket” that stores “files”. Unfor
protect the contents of the buc



Greta’s games use Cloud Storage to let people upload custom avatars for display in the game, and Shawn’s social network lets people share their photos with each other. Neither of them worry about running out of space, because Cloud Storage scales to *exabytes* of data. Have you ever stored an exabyte of data? Do you even have a mental model of how much data that is? I don’t! But I ran some numbers, and let’s just say it’s enough for every person on the planet to store 1000 high quality photos. Ping me if you publish the app that achieves this feat!

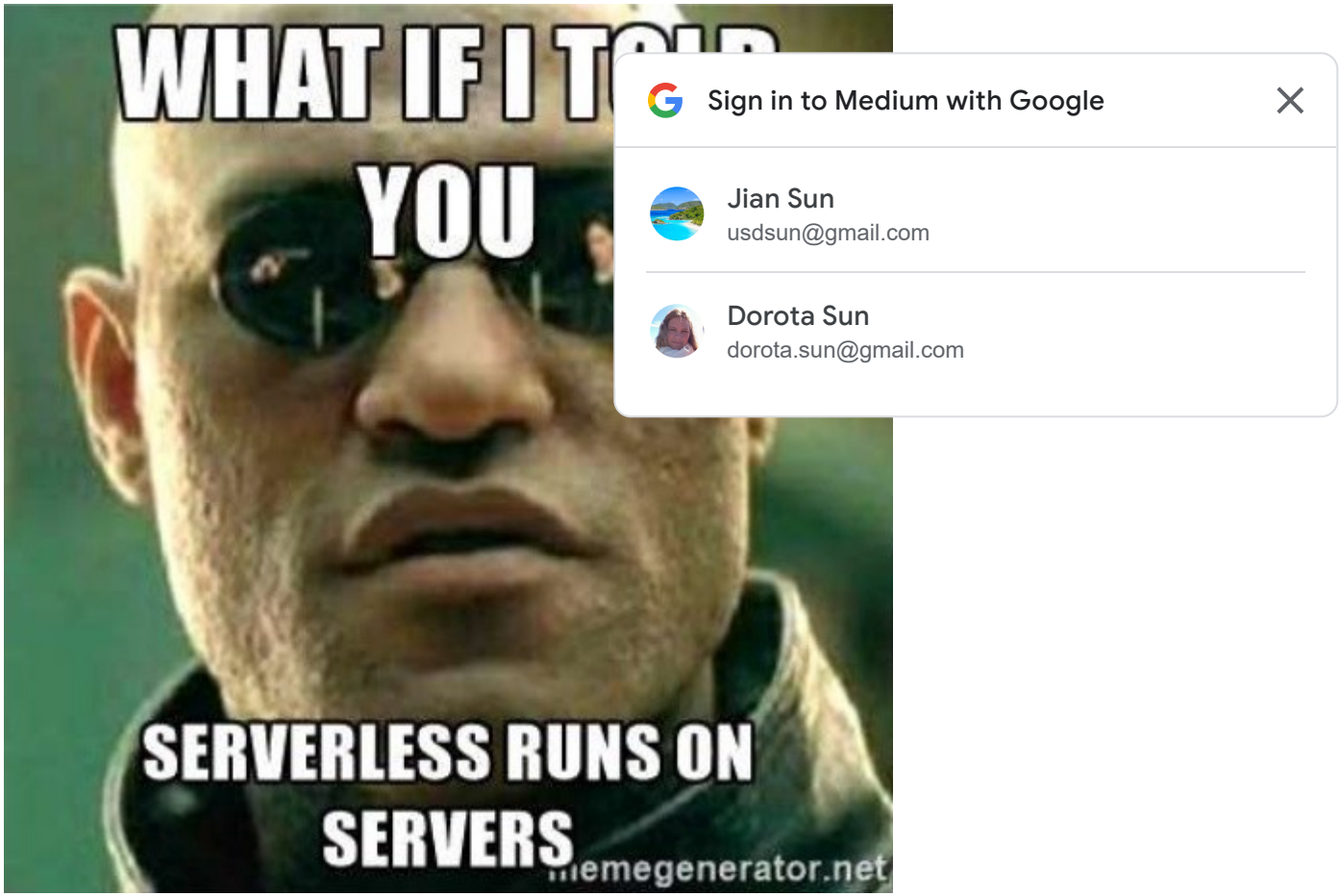
Authentication works extremely well with these three products with the use of **security rules** (for Realtime Database, Firestore, and Cloud Storage) that you can use to set access control to your data at the source. This ensures that clients can access that data *only* in the ways you allow, avoiding the tragic situation with the lolrus above. Users signed into an app with Authentication will automatically provide an identification token that you can use in your rules to protect who can read and write which items of

data. So, if you store personal data about your users, *definitely* use Firebase Authentication with security rules to limit acces
even get a gentle reminder from Firebase if your
permissive.

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

You can't be ignorant about "personal data". I mean, you can, but that's totally uncool. Use Firebase Security Rules! (view this comic on XKCD)

Cloud Functions is yet another a Google Cloud product that works well with other Firebase and Cloud products. Using the Firebase SDKs for Cloud Functions, you can write and deploy code, running on Google "serverless" infrastructure, that automatically responds to events coming from other Firebase products. That's right, it's serverless!



Thanks for the tip, Morpheus.

When people say “serverless”, they don’t suggest a lack of servers. With a serverless backend architecture, there are still servers in play, you just don’t have to know much about them. You don’t provision, maintain, scale, or do any of the devops required in a traditional (or “serverful”, my word) architecture. You just write and deploy code, and Google does the rest.

Cloud Functions for Firebase is the *one* product of the entire Firebase suite that actually has you writing backend code. In n
code *should* be running in a controlled backend
should be giving those backend devs a job, beca
earlier.

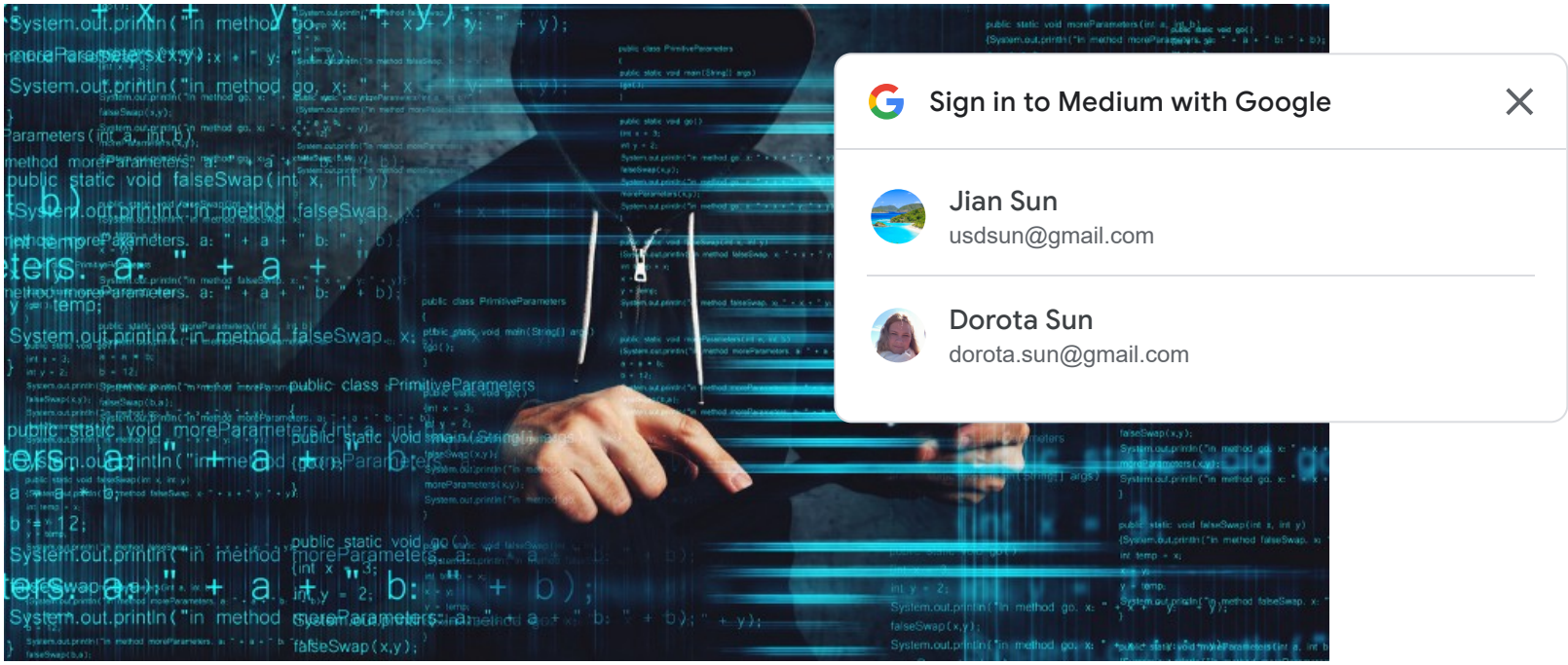
Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

The list of things you can do with Cloud Functio
look at all these samples! But I'll boil it down to one main concept: Firebase products (database, storage, auth, etc) emit events when data changes within the product, and your code deployed to Cloud Functions is triggered in response to those events.

Shawn uses Cloud Functions to automatically delete data from his database and storage when someone deletes their account (because user privacy must be protected in many situations, and getting billed for unused data is terrible). Greta uses Cloud Functions to execute game logic and scoring on a secure backend, because she knows that hackers may try to cheat by reverse engineering her game code.



This really is what it looks like when someone hacks your app. Hollywood got it right — don't hate on them so much!

Firebase Hosting is a secure, global web hosting CDN (Content Delivery Network). It's really good at quickly delivering static content (HTML, CSS, JS, images) using servers that are close to your users. And you can get it set up quickly, with or without your custom domain, along with a provisioned SSL certificate that costs you nothing.

Firebase Hosting has one important point of integration with the rest of Firebase, and that's through Cloud Functions. Firebase Hosting lets you proxy the request and response to and from Cloud Functions when writing

HTTP type functions. And, even better, it'll cache the responses from your functions, if you configure them properly. What

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

"RESTful" API!

Truth be told, I can't tell you exactly what "RESTful" means. I'm assuming you know. Feel free to rant in the comments.

ML Kit for Firebase lets you take advantage of a wealth of machine learning expertise from Google, without having to know anything about ML. This is great for me, because I don't know anything about ML! But what

I get out of ML Kit is the ability to recognize things that my device camera captures, such as text, faces, and landmarks. An device with very limited computing power. For t advanced understanding of ML (again, not me), TensorFlow model for more sophisticated use ca machine learning products at Firebase will be fu

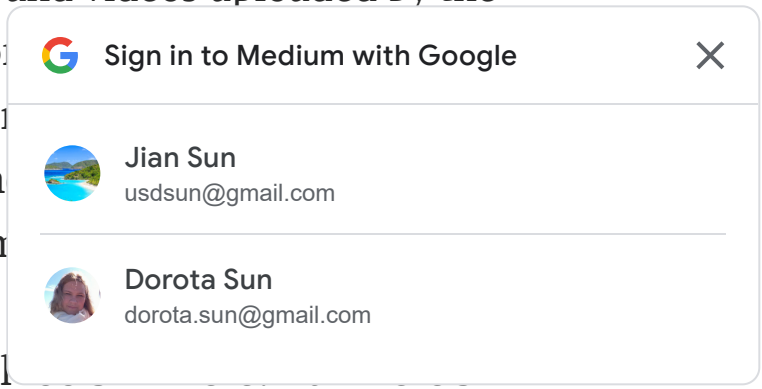
Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

Dr. Noonian Soong is going to be Google's employee #540138478.

Shawn uses ML Kit to locate faces in the photos and videos uploaded by the users of his social network, then he performs so
them. Greta doesn't use ML Kit yet, because the
implementation for the SDK yet. Dang! But if th
let people scan QR "coupon" codes for free pro



And that's it for the "build" group. Lots of useful
still much left to discuss about Firebase.

Read next: What would be possible if all our thoughts were connected and easily accessible? Meet Journal →

Grow your app — attract and retain users

The "grow" group of products are these:

Analytics — understand your users, and how they use your app
Predictions — apply machine learning to analytics to predict user behavior
Cloud Messaging — send messages and notifications to users
Remote Config — customize your app without deploying a new version; monitor the changes
A/B Testing — run marketing and usability experiments to see what works

best

Dynamic Links — enable native app conversion
marketing campaigns

App Indexing — re-engage users with Google S

In-App Messaging — engage your active users

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

Google Analytics for Firebase is the core of the
need to better know your users, and how they make use of your app, Analytics can show you that. When you publish an app for the first time, you might have an idea who your user base is going to be, where they live, and how they might use your app. And those ideas might be completely wrong in practice! The only way to know for sure is to collect data, and that's where Analytics helps.

Turns out, 9 out of 10 adults prefer Pokemon ov

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

There's *waaay* more to Google Analytics for Fire
summarized here, but there's one important thi
concept of an "audience". An audience is a group of users who have taken some predefined actions in your app, share some user properties, or have common device characteristics. You define the requirements for someone to become a member of an audience, and Analytics will figure out which users belong to it, by analyzing the stream of events that your app generates over time. This concept of an audience segmentation is powerful, because you can *target that audience* with other Firebase products in the "grow" category. Keep this concept of audience in mind as you read on!

Greta's game defines an audience of "players who have completed level 10, and made an in-game purchase" (core players). Shawn's social network app defines an audience of "people between the ages of 18–24, who have at least 50 friends on the network" (social young adults). Once these audiences have been defined in the Firebase console, and apps are updated to send the correct events, the audiences will collect members. Greta and

Shawn can then target the audience members with information and offers of interest to these groups.

Sign in to Medium with Google

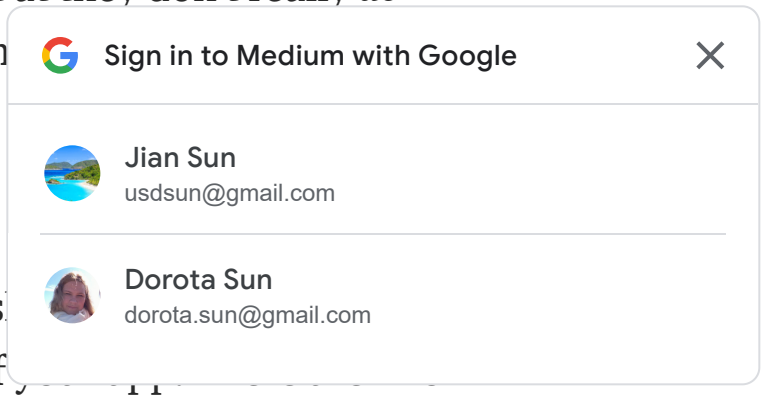
Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

Firebase Predictions builds on top of the data c
make predictions (no surprises there) about wh
likely to *churn* (not open your app), and which v
app). These two new categories of users are kinc
audiences, except you aren't required to do anything to define how a user ends up in one of these groups. This is the magic of machine learning! It's just like the magic of the Sorting Hat from Harry Potter, except no one frets about getting sorted into Hufflepuff.

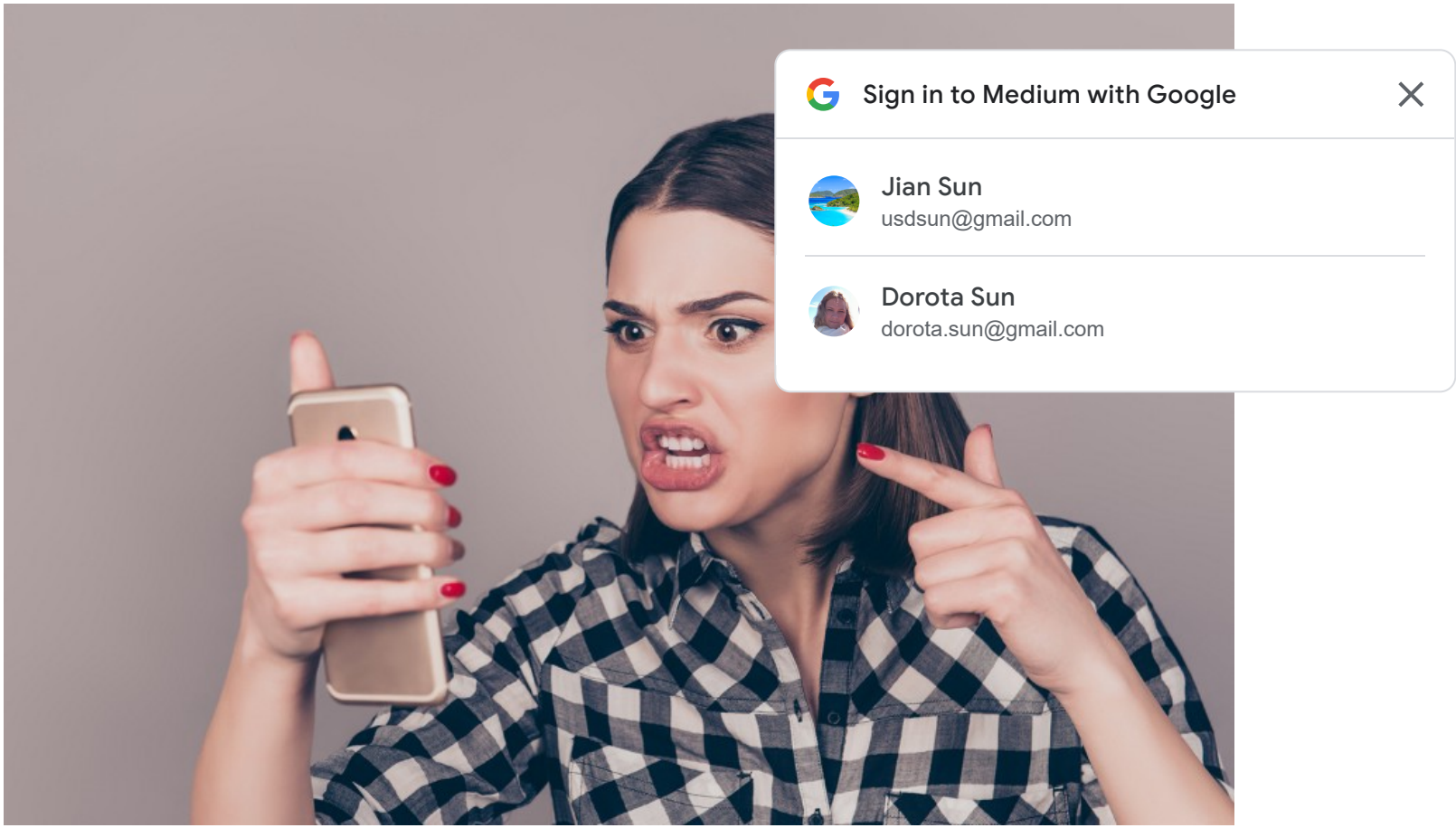
No one would "spend" to watch Harry Potter "churn".

Analytics and Predictions are both interesting, but they don't really *do anything* for your app. But when you team them
"grow" products, you can achieve truly magical
works.



Firebase Cloud Messaging lets you deliver pus
something of interest to your app, or the user of
ways to send a message. First, you can write code on your backend to *ping your app* when something gets updated that your app might want to respond to (for example, a chat room notification). Second, you can compose a message in the Firebase console to *ping your users* with information of interest. It's the second case — direct user notifications — that I'm more interested in today.

One thing you can do with user messaging, since it's integrated with Analytics and Predictions, is send a message to members of a particular Analytics audience or Predictions groups. This is valuable, because you can target users with information that they're more likely to be interested in and click on, keeping them engaged with your app. Targeting a specific group with relevant content is better than just blasting messages to everyone. The reason is clear enough. To illustrate that, here's a picture of someone getting too many push messages that are irrelevant to their interests:



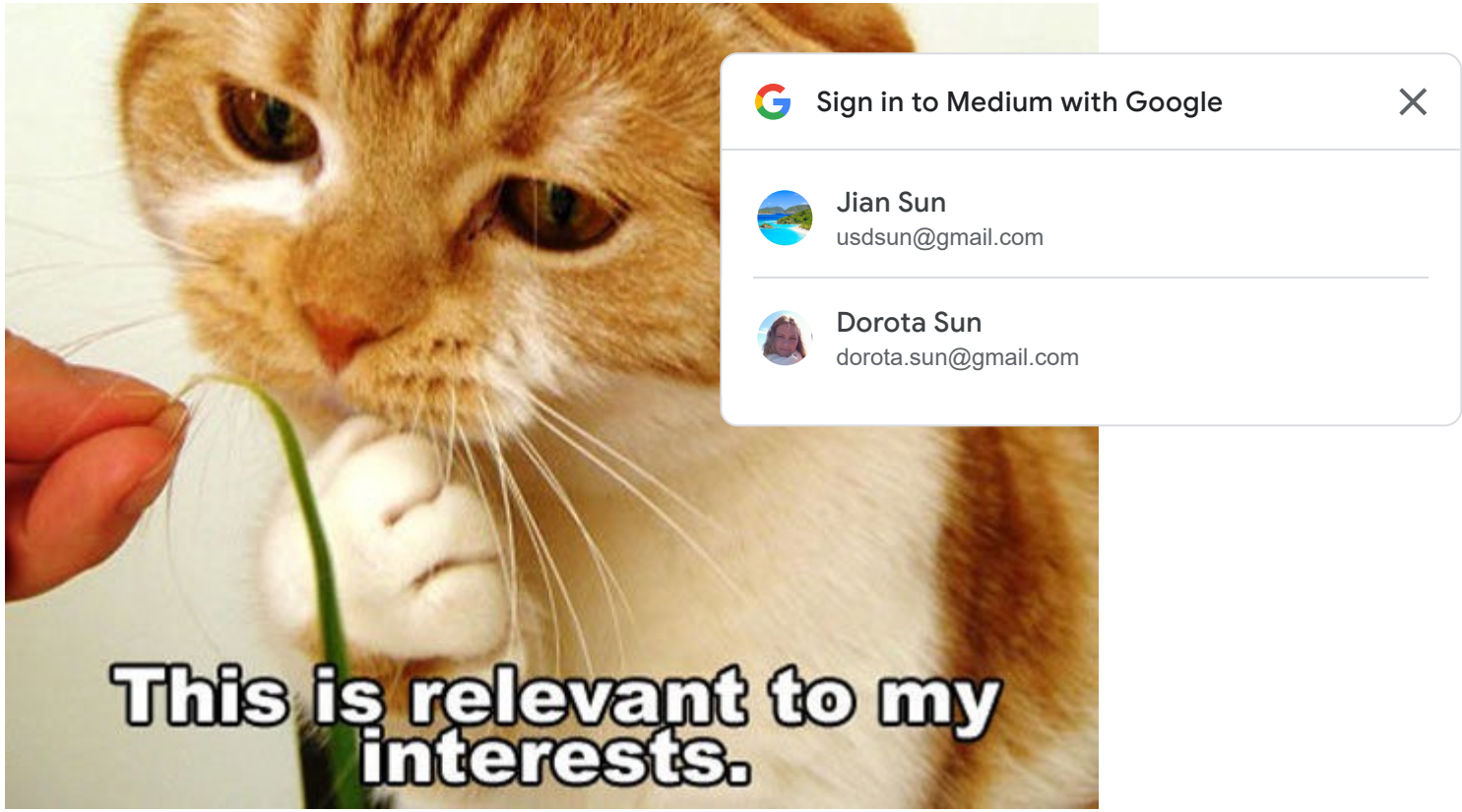
Seriously, don't be that spammy app — you'll get uninstalled. **Send only relevant and desired messages to your users**, at a reasonable frequency.

Greta might notify her core players audience of new game items for sale. And Shawn might send a reminder to his social young adults to rate and discuss their favorite night clubs. In both cases, people are getting messages they're more likely to act on, and fewer people are getting spammed, with assistance from Analytics.

Firebase In App Messaging helps you show targeted, customizable messages to your users to engage with key featu
be wondering “well, how is this different than F
asked! It means you’re still paying attention!) Th
that messages from FCM originate from a serve
Firebase console), whereas messages from FIAM
app itself (but configured in the console). The m
displayed while your user is actually using the app. But FCM and FIAM are alike in that they’re both deeply integrated with Analytics and Predictions.

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

It’s entirely possible that a message sent by FCM simply won’t be of interest to the user, because of its timing or relevance, or it was even accidentally dismissed. And did you notice that angry-looking woman above with the low tolerance for push messaging?! With FIAM, the message is delivered *at the very moment* it becomes relevant, based on criteria you define, determined by the user’s behavior as measured by Analytics events and Predictions groups.



Relevant messages hold the attention of cats on the internet.

For example, Greta's games allow players to earn virtual currency as they play, and uses FIAM to remind people who *just* earned their first virtual dollar that they can spend it in the game's store. And Shawn's social network points out to users the option for sharing content if they've been using the app for a while but haven't yet discovered it.

Firebase Remote Config lets you make dynamic changes to your app's behavior and appearance without having to publish an update to your app.

The general idea with Remote Config is that you define a bunch of configuration parameters in the Firebase consol SDK to periodically fetch those values and make You can think of Remote Config as kind of like a key/value pairs. This may sound like a simple da you can do with it than you might initially imag

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

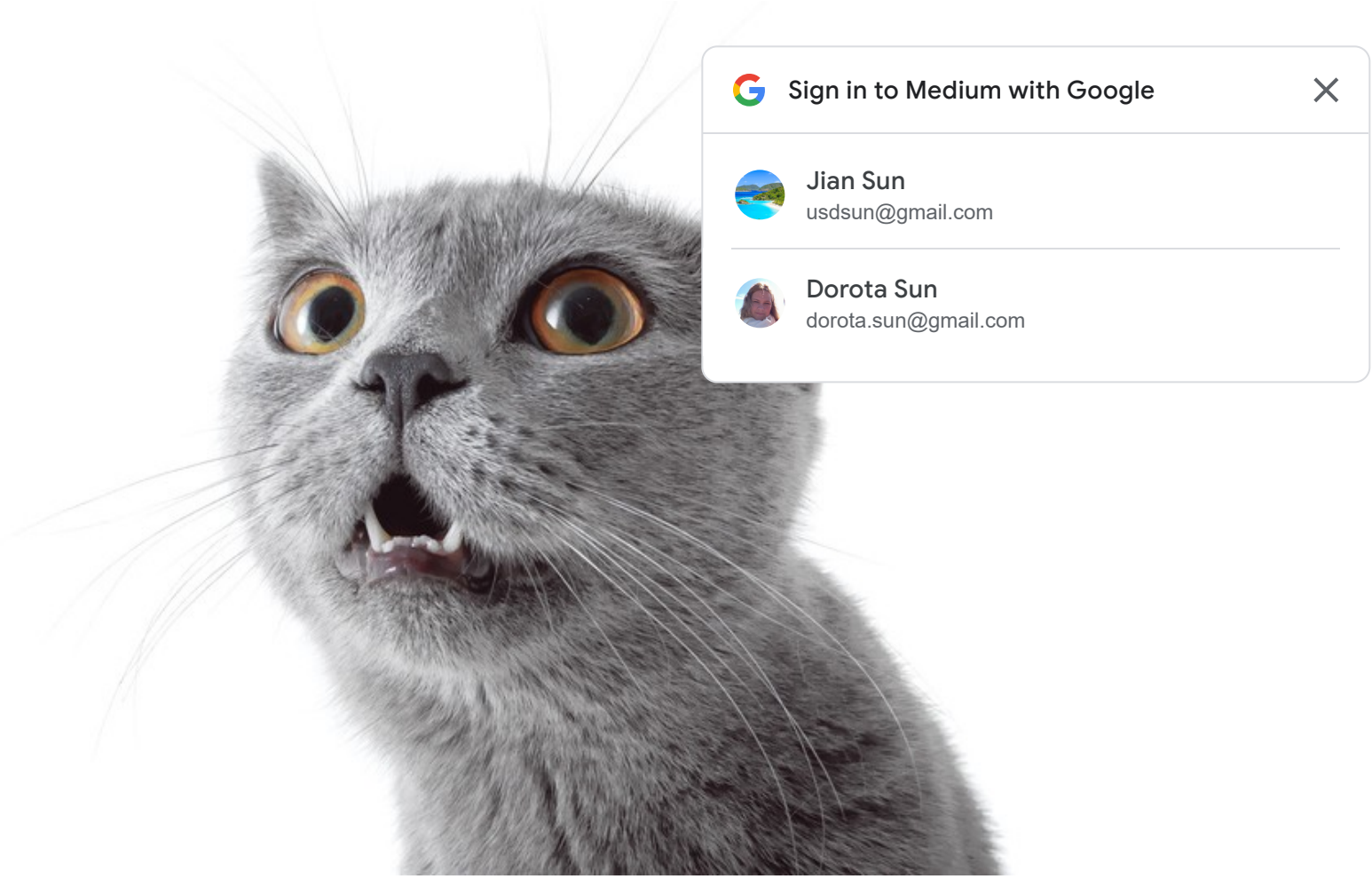
Remote Config lets you remotely control the way your app works. I use it to “Firebase and chill”.

The way that Remote Config really shines is through its ability to define conditions for each parameter. One type of conc
particular values to specific Analytics audiences
Predictions groups for “churn” or “spend”. This
number of useful features in your app, including
members a premium experience, or infrequent u

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

Shawn uses Remote Config to dynamically promote which social networking features should be promoted to certain users. Greta’s games use it to fine tune the difficulty of some levels and game mechanics, without having to build and publish a new version of the game.

Firebase A/B Testing takes the tight integration between Analytics, Remote Config, and FCM *even further*. I imagine you’re constantly making changes to your app, which is good. However, you probably don’t know for sure ahead of time if it’s going to help or hurt, unless you conduct your own studies. If you don’t have those kinds of resources, you can perform your own studies, and back them up with data. If you can figure out how to measure success, you can use Firebase A/B Testing to set up an experiment and conduct it on a handful of users before making a decision. This is wise, because making an uninformed decision about a change to your app could cause this to happen to your users:



OMG you did **what** to your app??

Firebase Dynamic Links builds on the existing concept of a “deep link” that launches your app to a particular screen or customized experience. Deep links work great if the user already has your app installed, but they don’t work well at all if they have to go install it first. Dynamic links improves on this by surviving the app installation process. When the user

clicks a dynamic link, and the app isn't already installed, they're directed to the appropriate app marketplace to install it. Th
the app for the first time, the context of the link
start with the experience that you originally inte
across platforms as well, so you don't need to ha
your Android, iOS, and web apps.

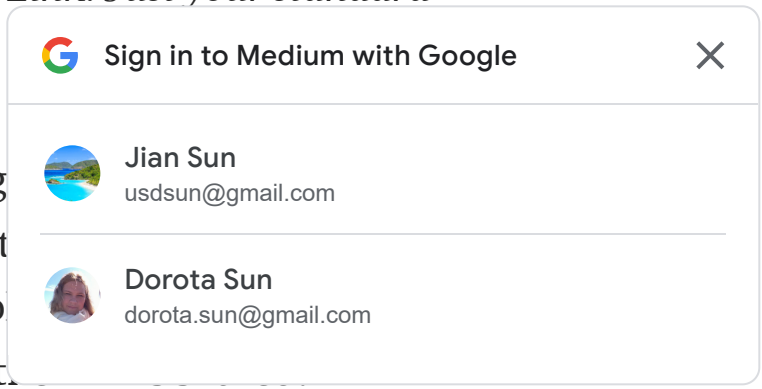
Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

That little green dude down there isn't a Dynamic Link. Just your standard Link.

That little green dude down there isn't a Dynamic Link. Just your standard Link.



Greta uses Dynamic Links to conduct marketing
users get a free in-game item when they follow t
for the first time. And Shawn uses them to enab
on his social network, no matter where or how t

Improve your app — stability and performance

The "improve" group of products are these:

Test Lab — scalable and automated app testing on cloud-hosted devices
Crashlytics — get clear, actionable insight into your app's crashes
Performance Monitoring — gain insight into your app's performance issues

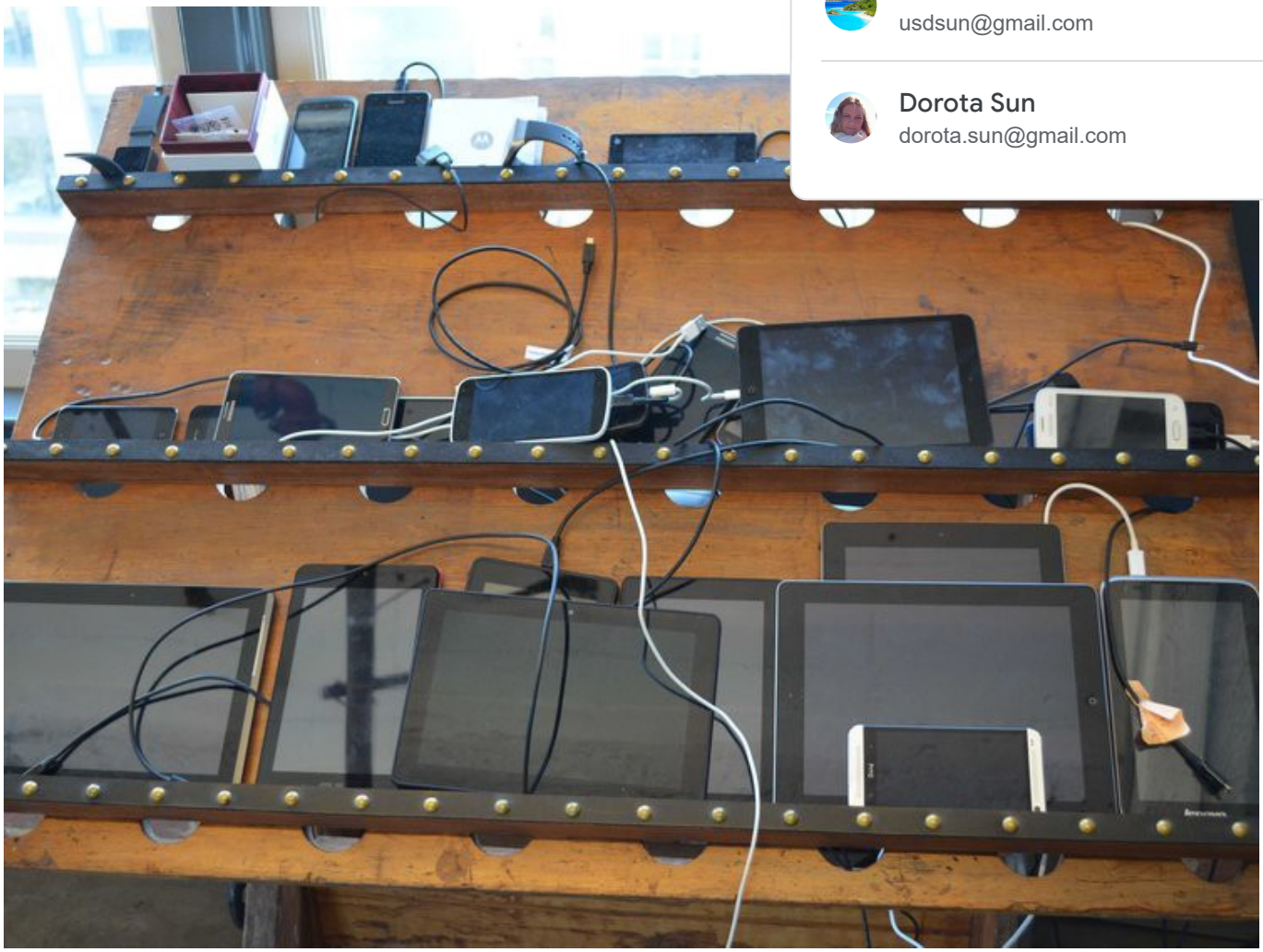
Firebase Test Lab provides you access to a large variety of iOS and Android devices, in additional to virtual Android devices, for testing your app. If you build apps for mobile devices, you probably have at least one device at your desk for development and testing. But this one device certainly isn't representative of what your users are actually using. Mobile devices come in all kinds of sizes, from many different manufacturers, with many different

versions of the OS in play. It's royally expensive and time consuming to try to maintain your own selection of devices, then

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com



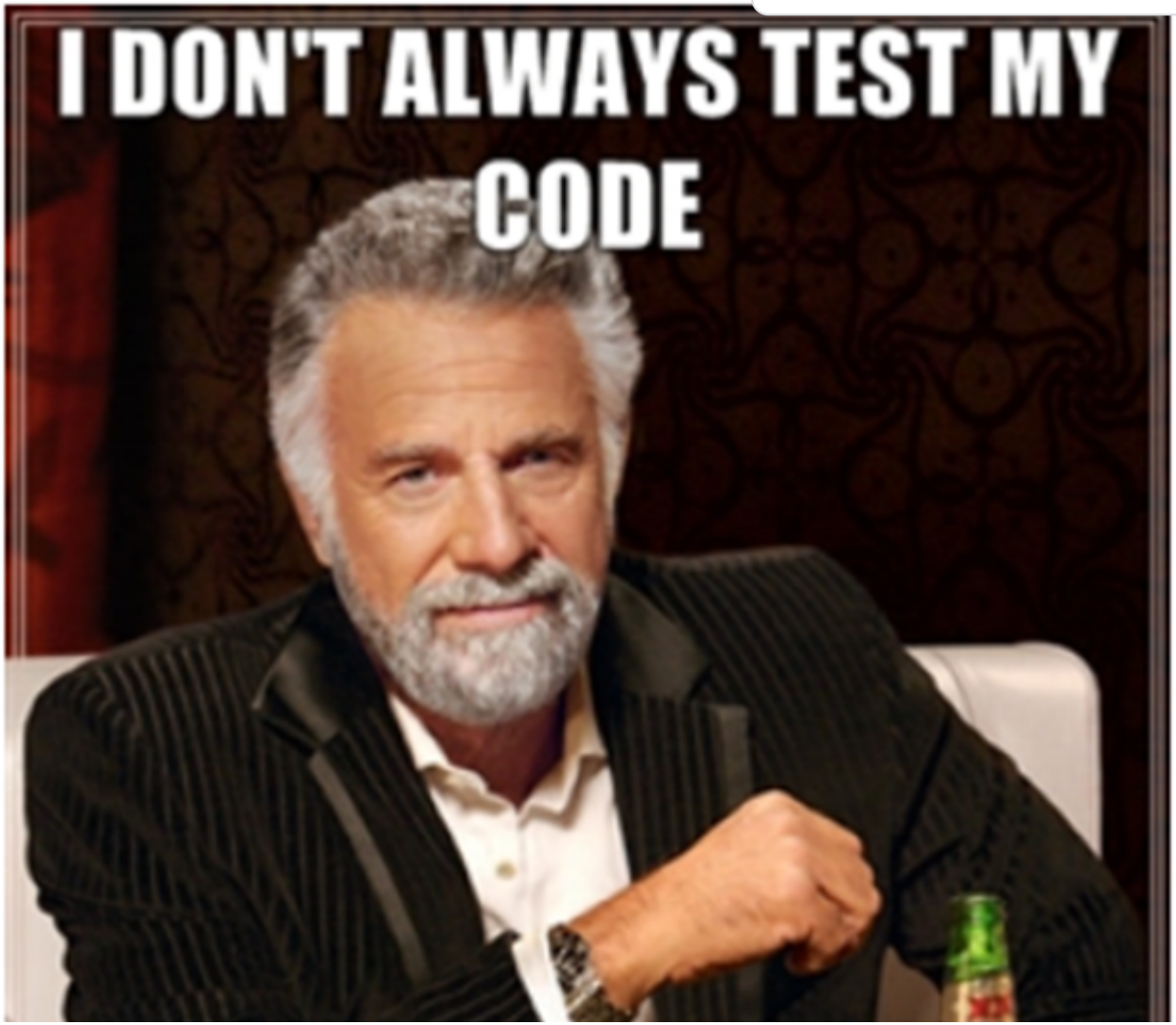
DIY device lab, DIY pain. Why would you do this to yourself?

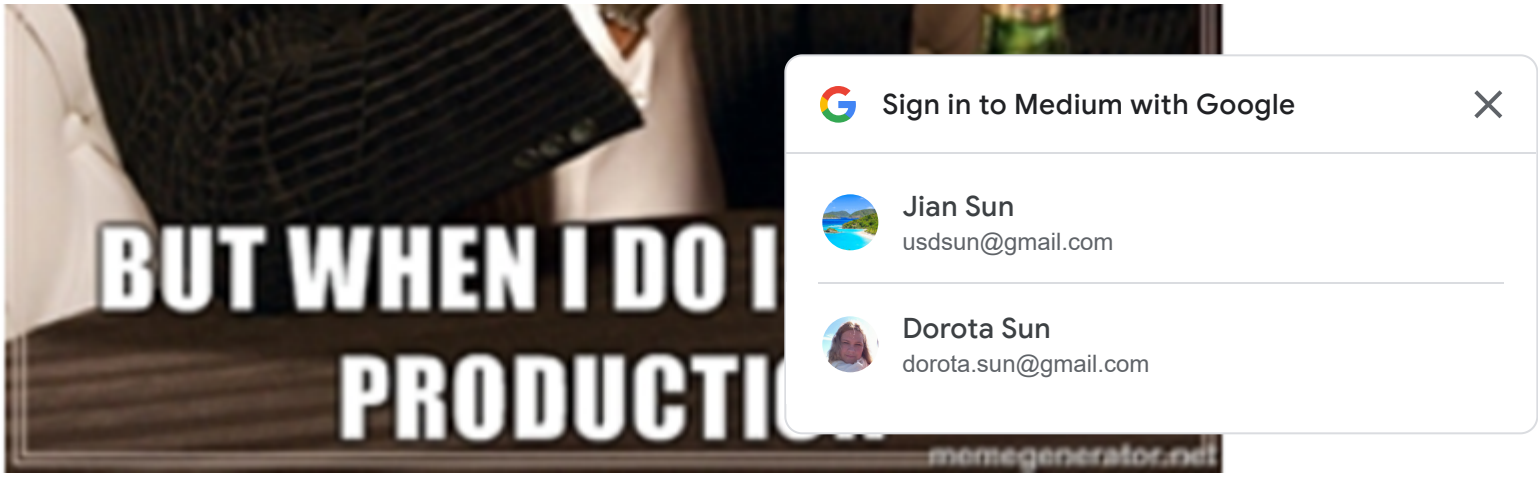
Test Lab solves this by hosting a large selection of actual, physical devices that will install your app and run your test suite (XCTest) against it. It also can perform a fully aut no additional coding, for the truly lazy among u amirite?).

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

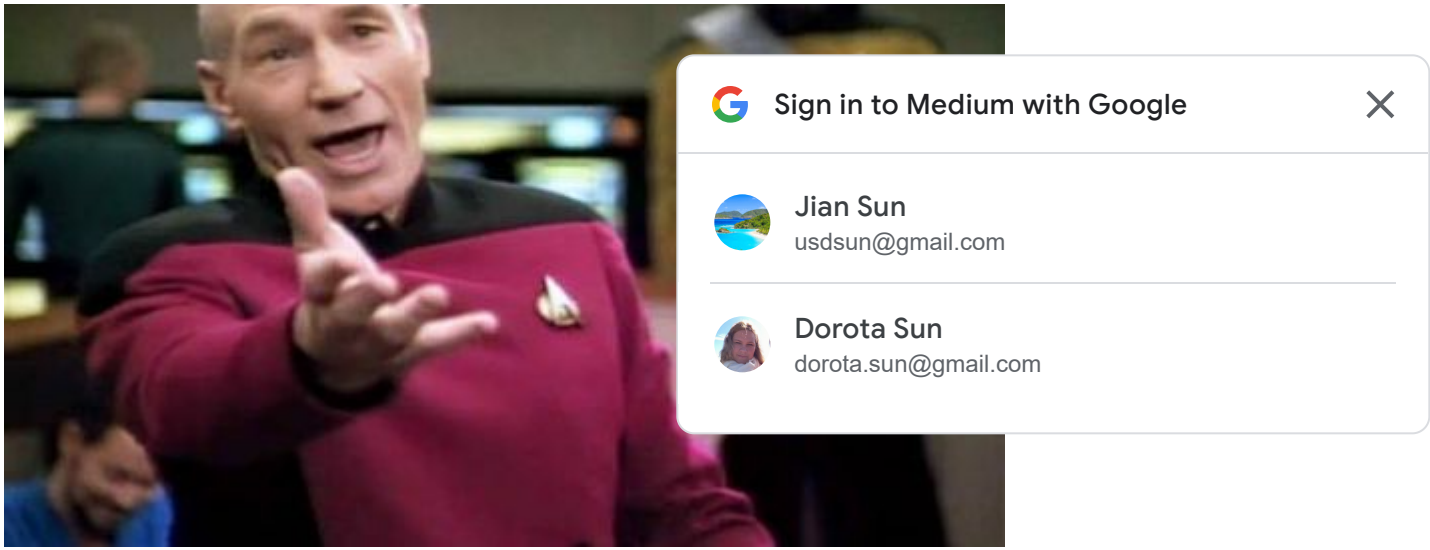




The most interesting man in the world is also the world's worst software engineer. **Don't test on your users, test in the lab.**

Firebase Crashlytics is the best crash reporting tool in the world. Seriously, it's the best. I don't know why I'm even bothering to type stuff about it. It's been around since forever. Just use it. It's even integrated with Analytics, so you can measure how crashes are affecting the way users use your app (possibly by uninstalling it). App crashes aren't even funny, so I won't bother with a funny picture to illustrate a concept that everyone understands.

Oh, who am I kidding? Here's the picture.



App crashes are not “federated”.

It’s the home stretch! Only one more product to discuss!

Firebase Performance Monitoring gives you insights into your app’s performance issues, from your users’ point of view, by measuring its HTTP requests, startup time, and other code using its API. The magic in this Firebase product is that it starts measuring the HTTP request and startup time without you having to write more than a couple lines of code, with the results visible in the Firebase console. Add a few more lines to time anything else your app might be doing.

I’ll underscore the importance of getting metrics *from your users’ point of view*. You probably develop your app using fast devices on fast wifi

networks. This setup almost certainly isn't similar to the conditions that your users face all over the world, on flakey mol devices. If you want to understand the pain that when using your app, try pulling out your nose pliers. Then, when you finally realize that was a integrate Performance Monitoring into your app Firebase console. You'll be able to see exactly ho might be in different parts of the world, on different networks, on different devices, with different versions of the OS. And you'll see which HTTP endpoints are causing the biggest delays in your app, giving you a target on which to focus your optimizations.

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

Sign in to Medium with Google
Jian Sun
usdsun@gmail.com
Dorota Sun
dorota.sun@gmail.com

What's not immediately obvious is how the thre
category work well together, even though they c
integrations like those in the other Firebase cate
run a special version of your app in Test Lab, and when that's done, you'll be able to see more detailed crash information in Crashlytics (if it crashes, of course) and better performance measurements in Performance Monitoring. I also know someone who was able to use a Crashlytics crash report to find a performance problem in Performance Monitoring. You can be clever like this too!

Whew, that was a lot of text and pictures

You made it to the end! Guess what? You just found out that **Firebase has a lot of stuff in it**. So what do you take away from all this? I'll boil it down to a few points so you can get back to building your app:

- Firebase is Google's mobile application development platform. *(YES, THAT AGAIN.)*

- You're gonna save a ton of time and money using Firebase products rather than trying to build them yourself.
- You can use all of it, or none of it, or just the
- All those bits are designed to work well toget console.
- You gonna use Firebase to build that earwax-busting app already?

Sign in to Medium with Google

Jian Sun
usdsun@gmail.com

Dorota Sun
dorota.sun@gmail.com

Addendum: What Firebase isn't

That's all great info up there about what Firebase *is*, but it's come to my attention that we also need a reference for everything that Firebase *is not*. For the record:

- Firebase is a platform, not (just) a database (any more). We already covered this.
- It's "Firebase", not "FireBase". If you write it that way, you will be punched in the throat.
- Similarly, it's "Firestore", not "FireStore", but the punch goes to the kidney.

- It's NEVER abbreviated FB. Do that, and you'll be yelled at until your ears bleed.
- If Firebase was an element, it would combat
- It's "realtime", not "real-time", despite the su checker.
- It's "Cloud Functions for Firebase", not "Firebase Functions". Go ahead, search the docs and prove me wrong. I dare you.
- It's "Cloud Firestore", not "Firebase Firestore". Seriously, who needs that much fire??
- Cloud Firestore is not Cloud Filestore is not Cloud Datastore is not Cloud Memorystore is not Cloud Storage. Got it?

Sign in to Medium with Google

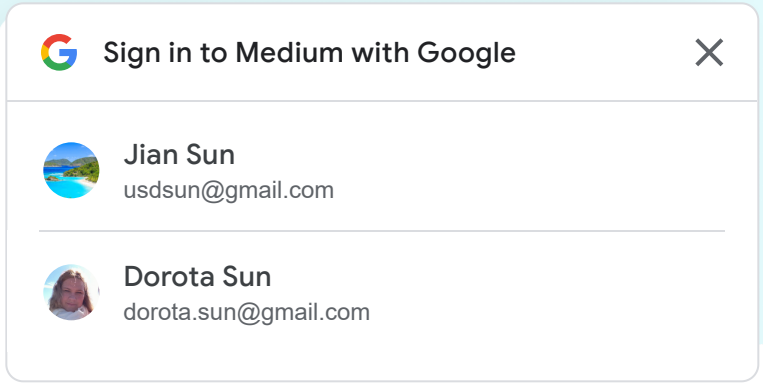

Jian Sun
usdsun@gmail.com


Dorota Sun
dorota.sun@gmail.com

. . .

Read next: What would be possible if all our thoughts were connected and easily accessible?

Meet Journal →



Sign up for The Firebase Developers Quarterly

By Firebase Developers

A summary of what has happened on the Firebase Developers publication in the past quarter. Sent once a quarter. Take a look.

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

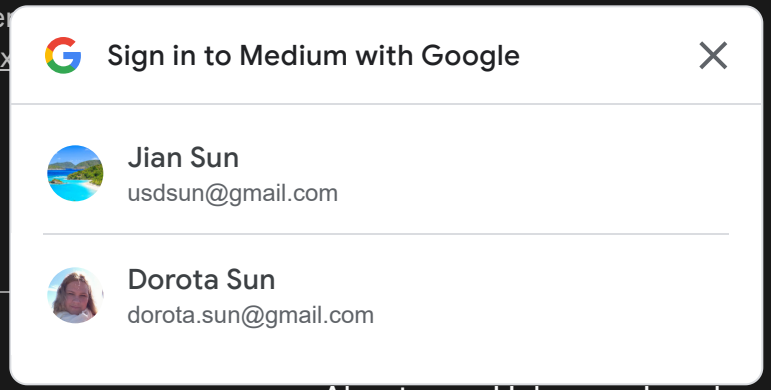
Firebase | Mobile App Development | Web Development | Android App Development | iOS App Development

Learn more. Make Medium yours. Share your thinking.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. Learn more

Follow the writers, publications, and topics that matter to you, and you'll see the
your homepage and in your inbox. Ex

If you have a story to tell, knowledge to



Medium

About Help Legal