

Supercharge your app
with Cloud Functions



Zamzam Farzamipooya

Tech Lead, Veo

@zamzampooya

#WomenLifeFreedom



Peter Friese

Developer Advocate, Firebase

@peterfriese



Goals for today

Intro to Cloud Functions

- What are they
- How to use them in your apps

Hands-on use-cases

- Simple processing
- Calling 3P APIs
- Reacting to changes in your app
- Sending emails to users

Types of Cloud Functions

- HTTP Functions
- Callable Functions
- Triggers (Firestore / Time-based)

Beyond Cloud Functions

- Firebase Extensions
- AI 



What we're going to build today



Information Overload

The image is a collage of screenshots from various tech websites, primarily focused on Firebase and Google Play Console, illustrating the concept of 'Information Overload'.

Firebase Blog Screenshot: A dark-themed blog post titled "Add Remote a Jetpack app" by Steve Suppe. The post discusses how to add a Jetpack Compose app to a Firebase project. It includes sections like "TL; DR", "Firestore", and "previous".

Google Play Console Screenshot: A smartphone displaying the Google Play Store interface, showing search results for "Hello" in multiple languages.

Google Developers Blog Screenshot: A light-themed blog post titled "Control when changes to your app are sent for review with new Play Console features" by Steve Suppe, Manuel Wang, Ashley Marshall, and Ashley Marshall. It features a hand interacting with a smartphone screen showing a checkmark icon.

Common Elements: All screenshots feature a top navigation bar with links to "Products", "Use cases", "Pricing", "Docs", "Support", and "Go to console". The "Go to console" link is present in the top right corner of each screenshot.

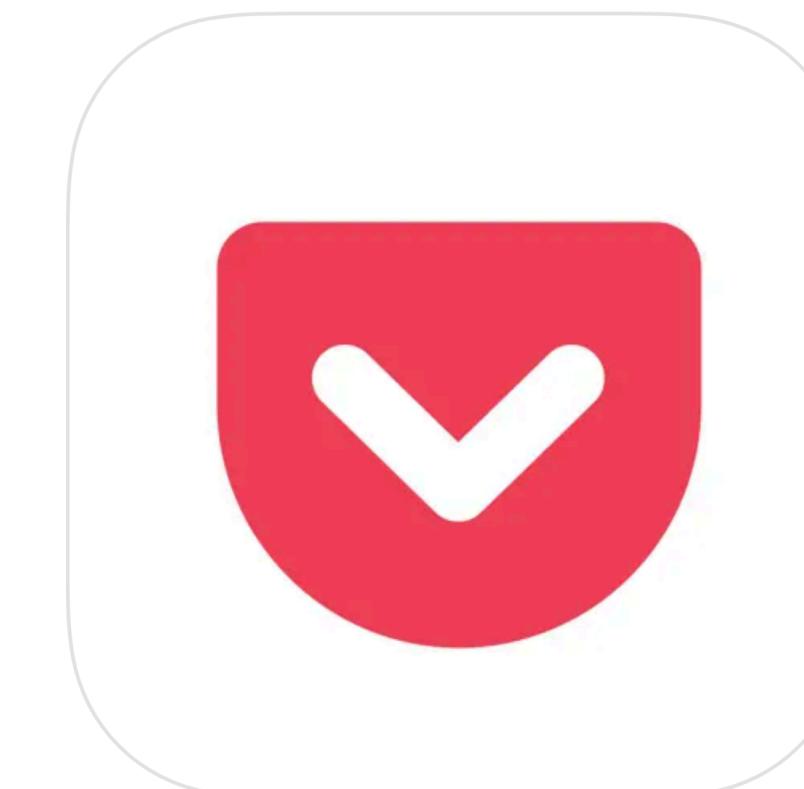
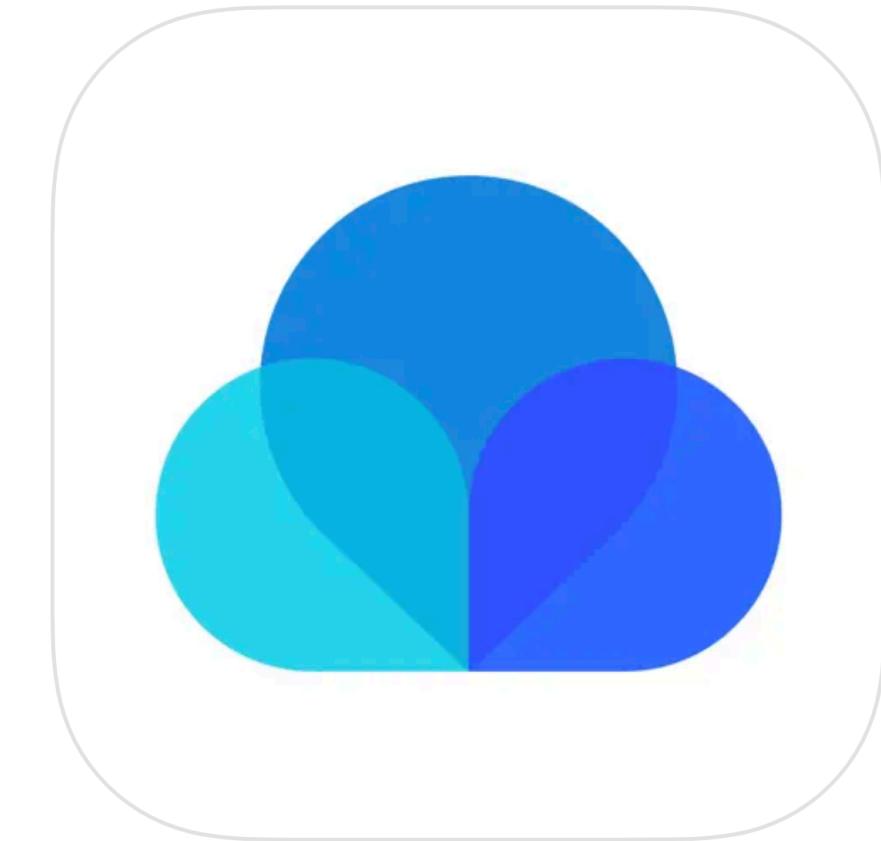
Text from the Google Developers Blog Post:

- Post Title:** Control when changes to your app are sent for review with new Play Console features
- Date:** 24 January 2023
- Authors:** Posted by Steve Suppe, Product Manager, Google Play and Manuel Wang, Product Manager, Google Play; Ashley Marshall, UX Writing Lead, Google Play
- Text:** Google Play Console is constantly evolving to improve how you manage and publish your apps. We know that launch moments are really important to you. Whether that's launching a new version of your app, or updating your store listing – you need the right tools to help you launch with confidence.
- Text:** One challenge you've shared with us is a lack of predictability and control over the app review process. Previously, it was hard to predict which changes would be sent to Google for review, and which changes would be published immediately. There was also no way to send multiple changes for review together, for example, if you wanted to update your app at the same time as one of your store listing screenshots.
- Text:** As a result of your feedback, we're making some changes to give you more flexibility and control over the app review process.

A black and white photograph of Steve Jobs, wearing round-rimmed glasses and a dark turtleneck, gesturing with his hands open towards the camera. He is set against a dark background.

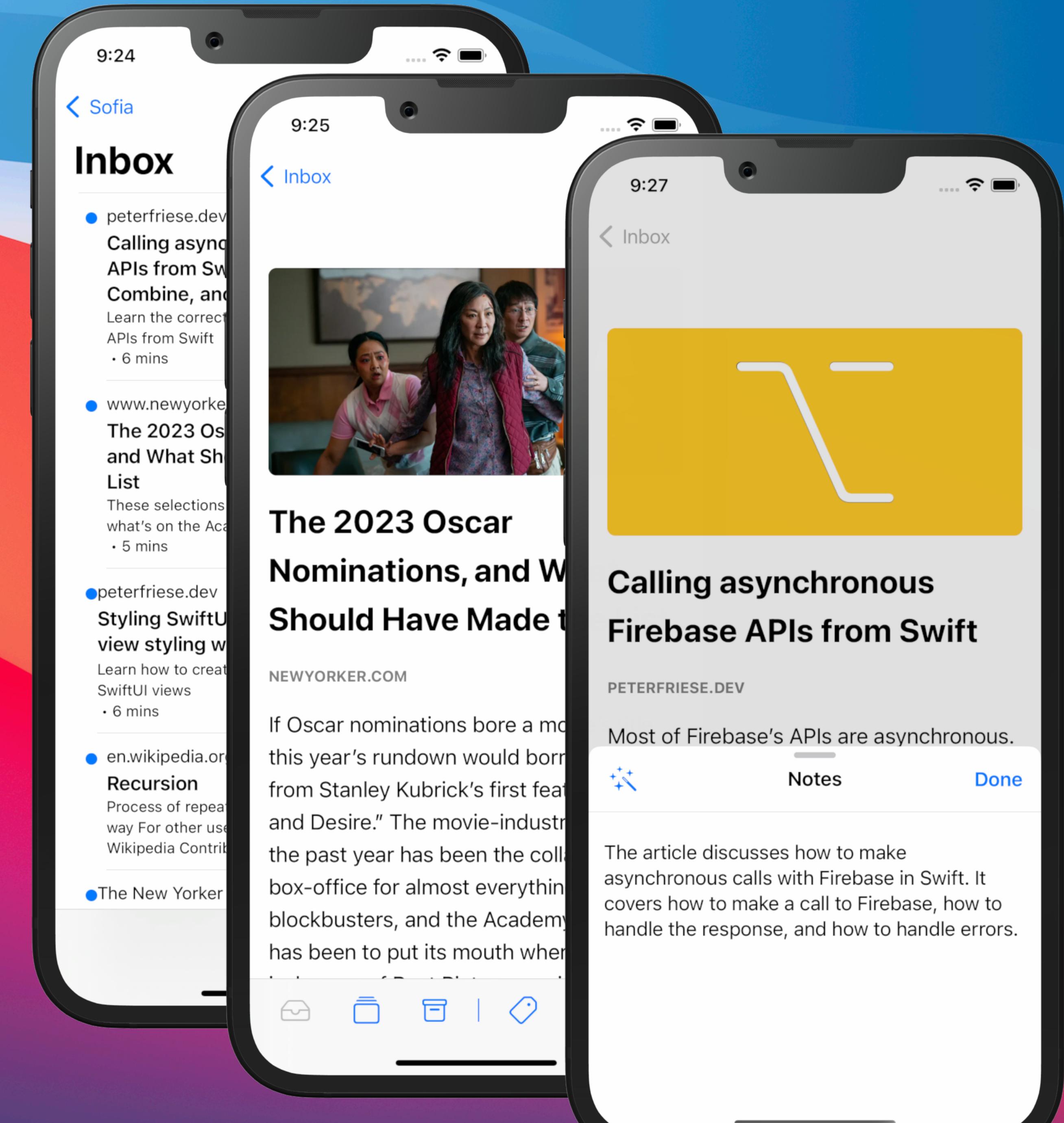
Isn't there an app for that?





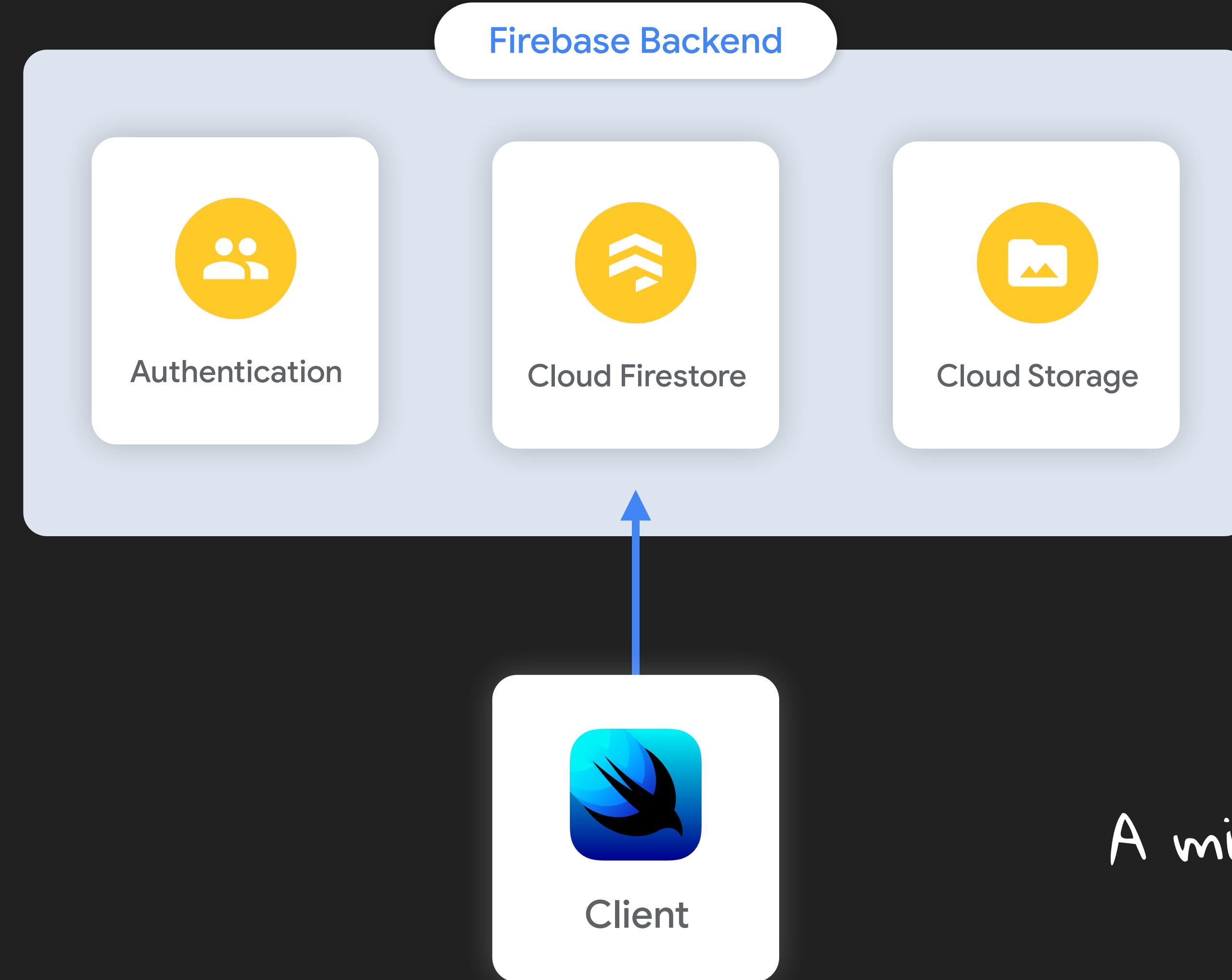
Read It Later

- ★ Add links via iOS Share Extension
- ★ Readability
- ★ Extract OpenGraph Metadata
- ★ Summarise (via GPT-3 / PaLM)
- ★ Send weekly digest by email





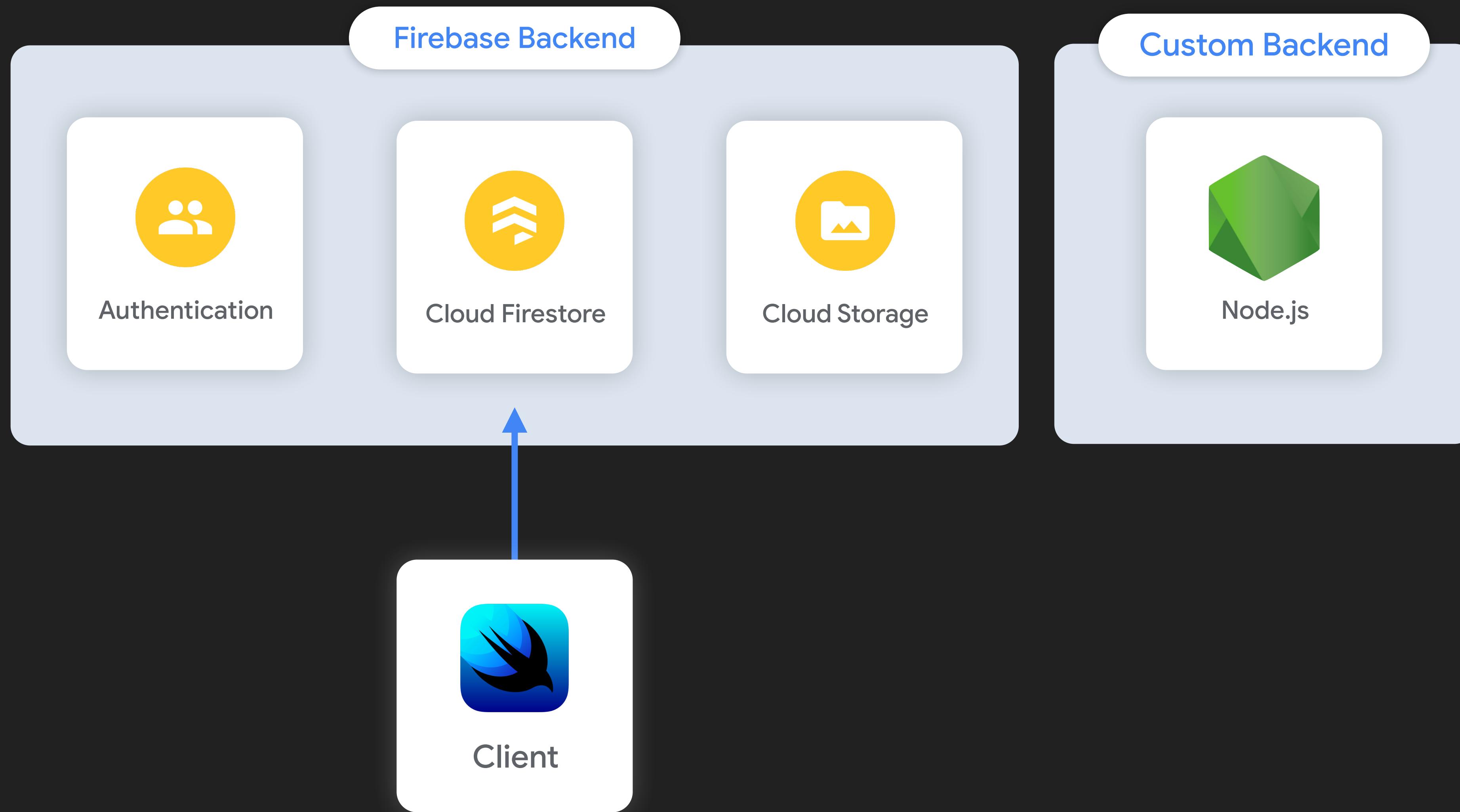
Serverless Architecture



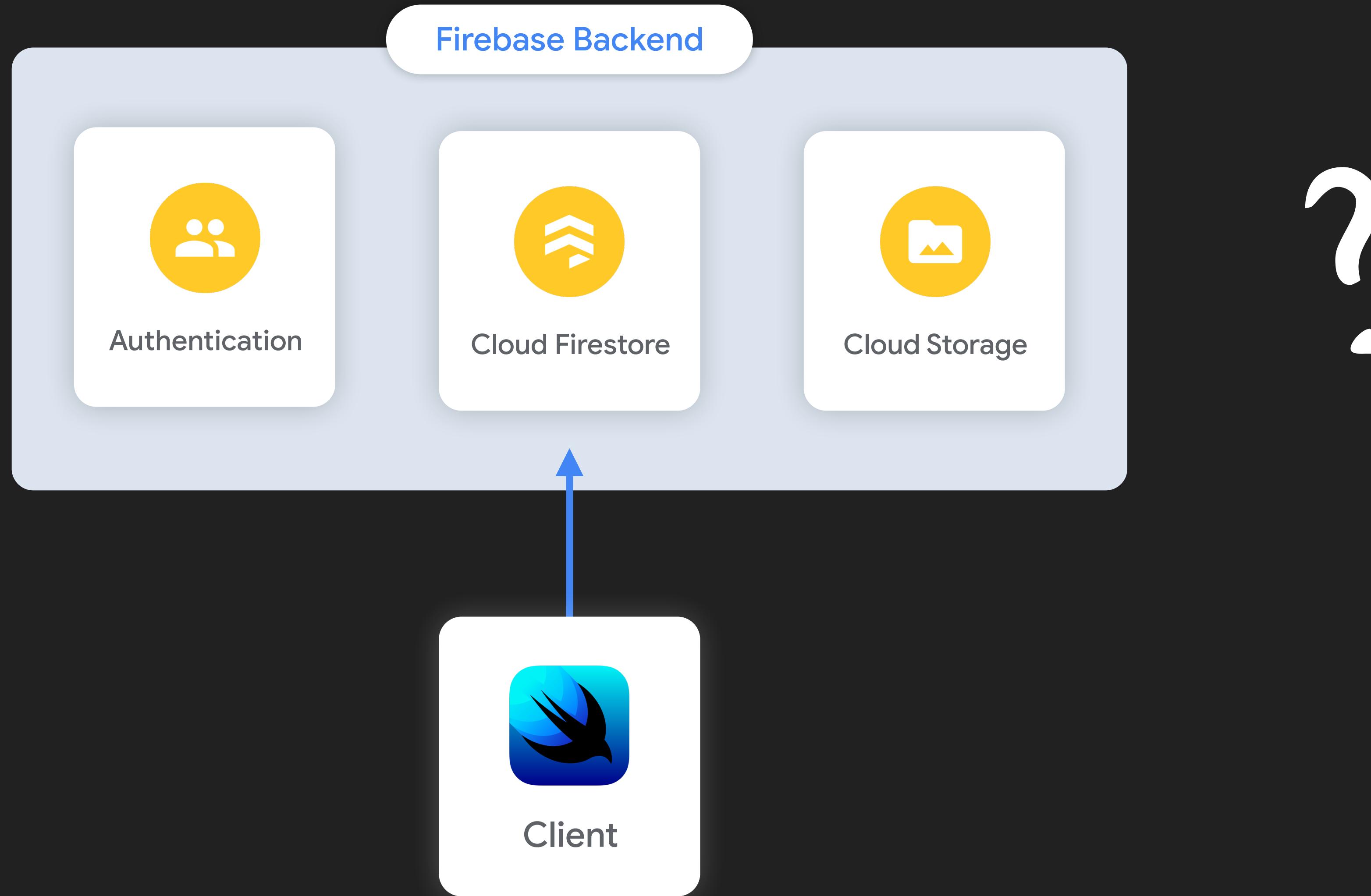
A miracle happens here



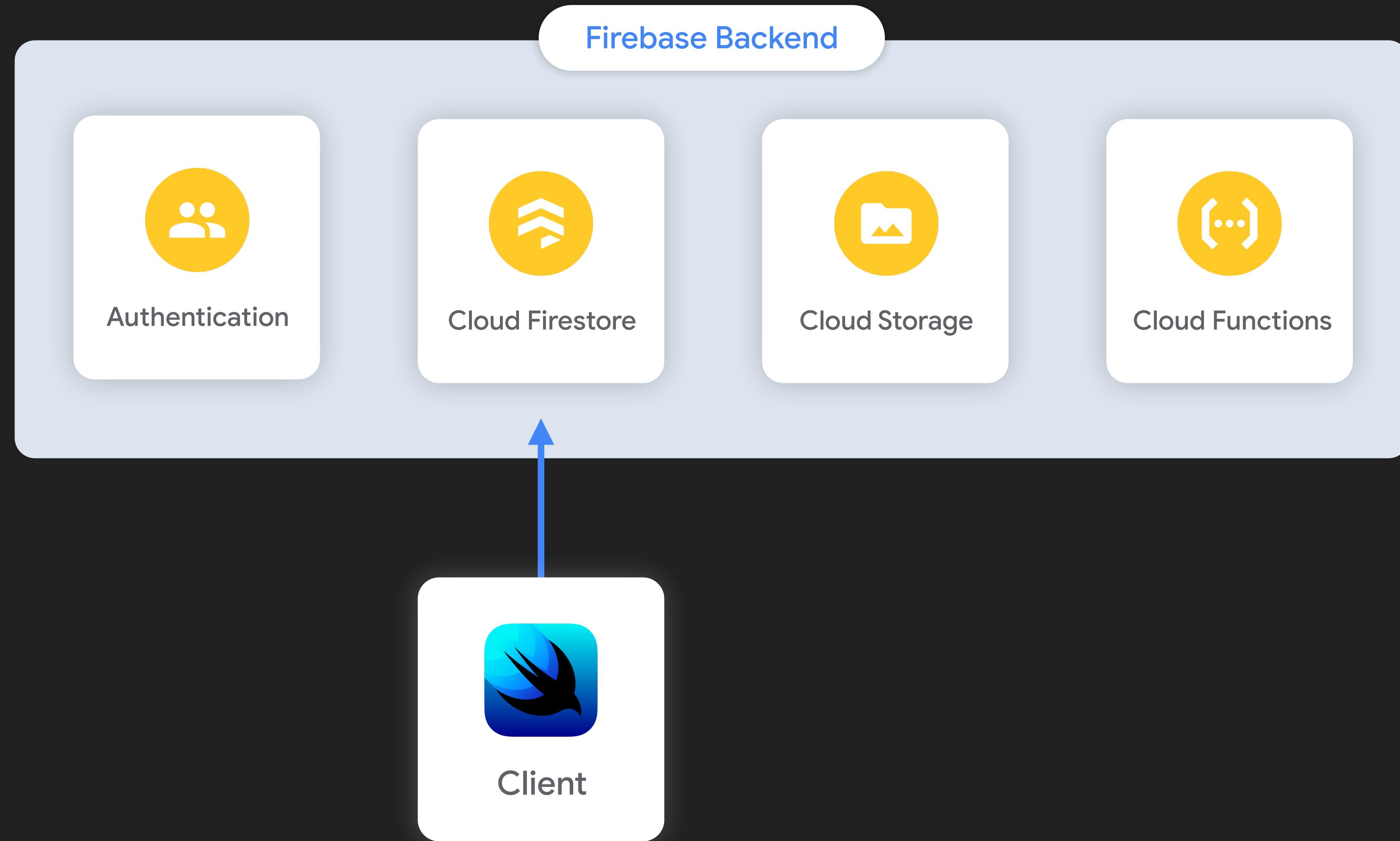
Serverless Architecture



Serverless Architecture



Serverless Architecture











Cloud Functions for Firebase





Cloud Functions for Firebase

- Serverless
- JavaScript / TypeScript / Python
- Automatic scaling
- Run in trusted environment
- Zero maintenance
- Easy deployment
- Can run intensive / long running tasks



What's new in Cloud Functions?

2nd gen runtime

- Fewer cold starts
- Concurrency

Faster deployments

- Codebases
- Skip no-op deploys

Parameterized configs

- Type-safe configurations
- Support for Cloud Secret Manager

Try them now

- Use alongside v1 functions
- Emulator support



Different ways to call your functions



Different ways to call your functions

HTTPS

Listen for HTTP requests

Callable

Call a function from your app
with extra context

Different ways to call your functions

Scheduled

Using crontab or AppEngine
cron.yaml syntax

In a queue

For resource-intensive, long-
running tasks

Different ways to call your functions

React to events

Data written

New user created

User signing in

Image uploaded

Crashlytics alert

Analytics Conversions

File deleted

Data deleted

Test run completed

Data updated

New configuration

Configuration rollback

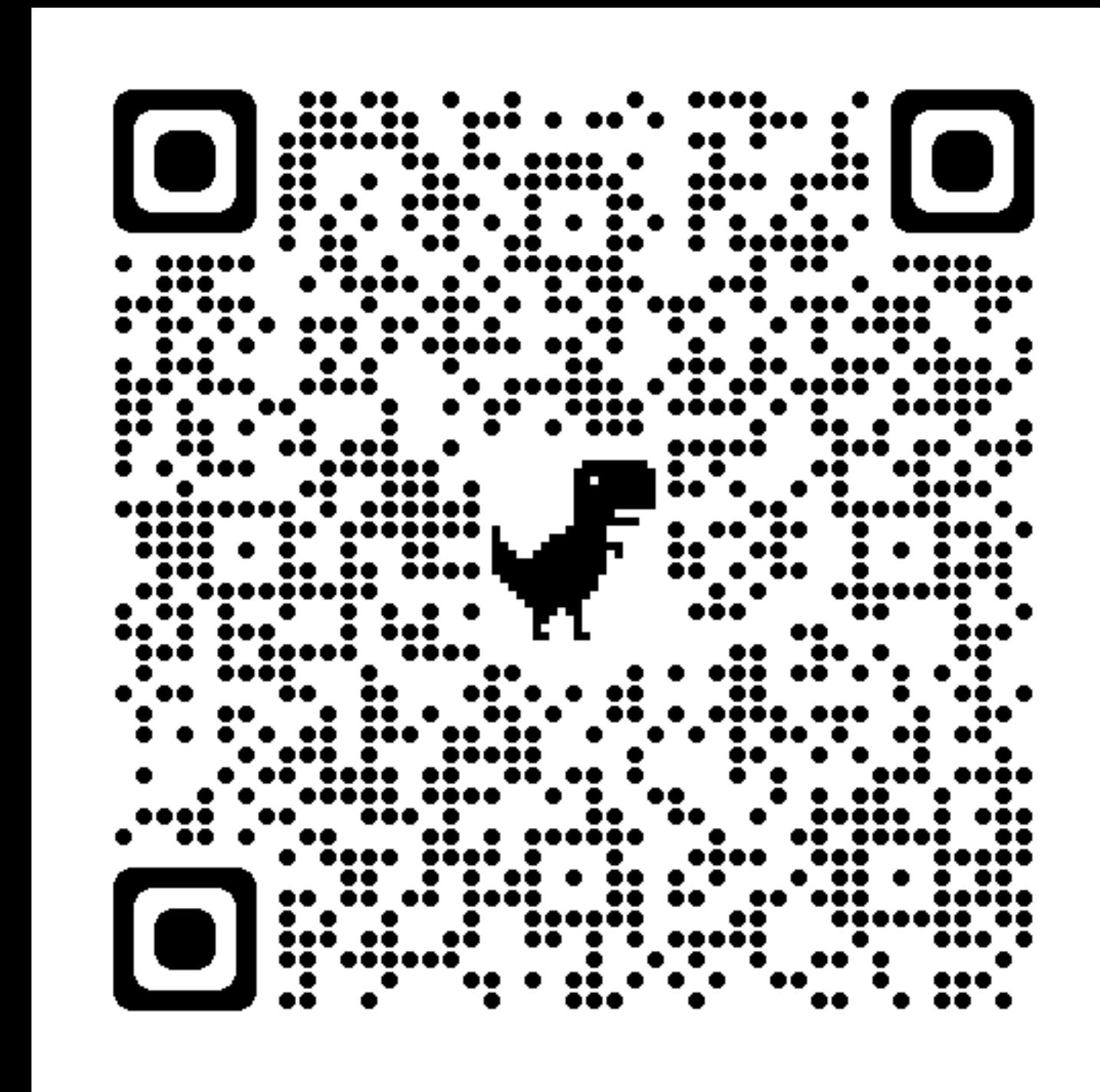
Tutorial



Goals for today

- 1: Simple HTTPS Functions
- 2: Cloud Firestore Triggers
- 3: Callable Functions
- 4: Firebase Extensions

Launch the tutorials



<https://bit.ly/41BpQxi>

Useful commands

```
# Login to your account  
firebase login
```

```
# Add another account  
firebase login:add
```

```
# Show available project aliases  
firebase use
```

```
# Add a new alias  
firebase use --add
```

```
# Switch to a specific project alias  
firebase use production
```

```
# Paste stuff into a text field (Android)  
adb shell input text "some text"
```

Chapter 1

Simple HTTPS Functions

HTTPS Functions



Execute a Cloud Function by sending a HTTP request

- Supports GET, POST, PUT, DELETE, OPTIONS
- Can be used to connect with Express apps
- No access to signed in user
- Callable Functions are a better fit for mobile apps

Calling an HTTP Function



```
// Construct the URL with the name as a query parameter
let url = URL(string: "YOUR_ENDPOINT/functionName?parameter=\(value)")!

// Create the URLRequest
var request = URLRequest(url: url)
request.httpMethod = "GET"

// Send the request and wait for the response
let (data, _) = try await URLSession.shared.data(for: request)

// Decode the response using the GreetResponse struct
return try JSONDecoder().decode(SomeType.self, from: data)
```

Chapter 2

Cloud Firestore Triggers



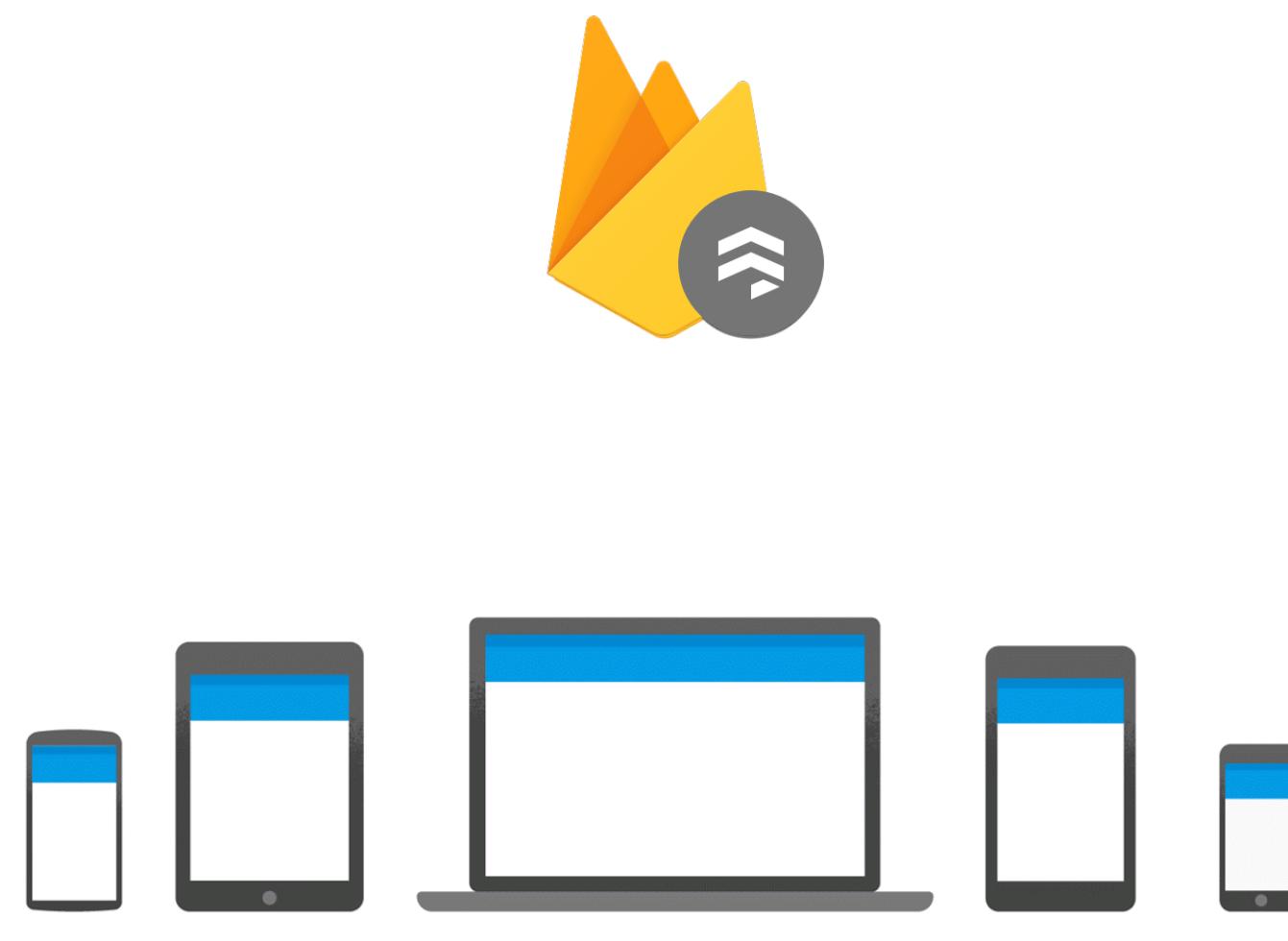
Firestore



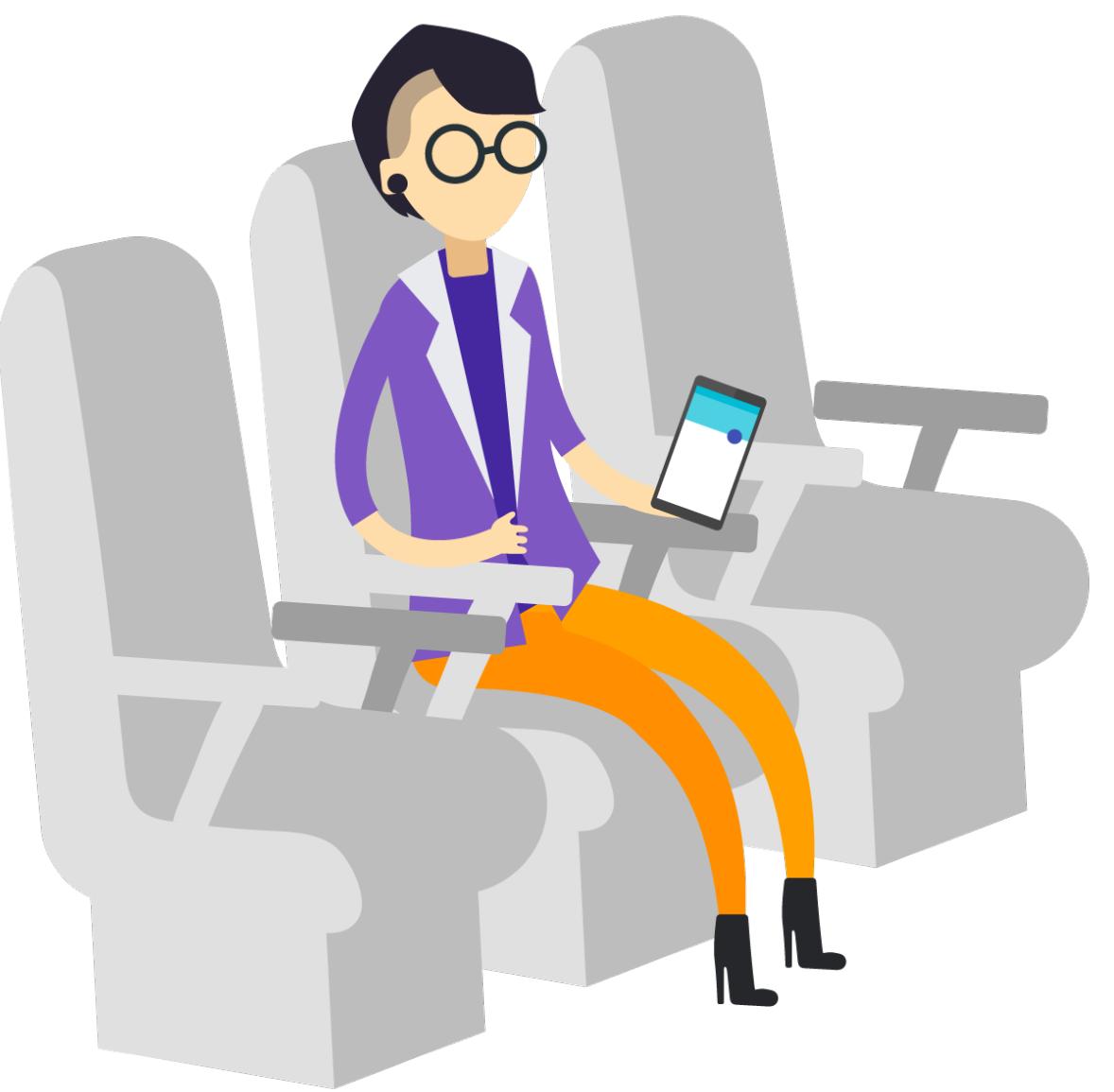




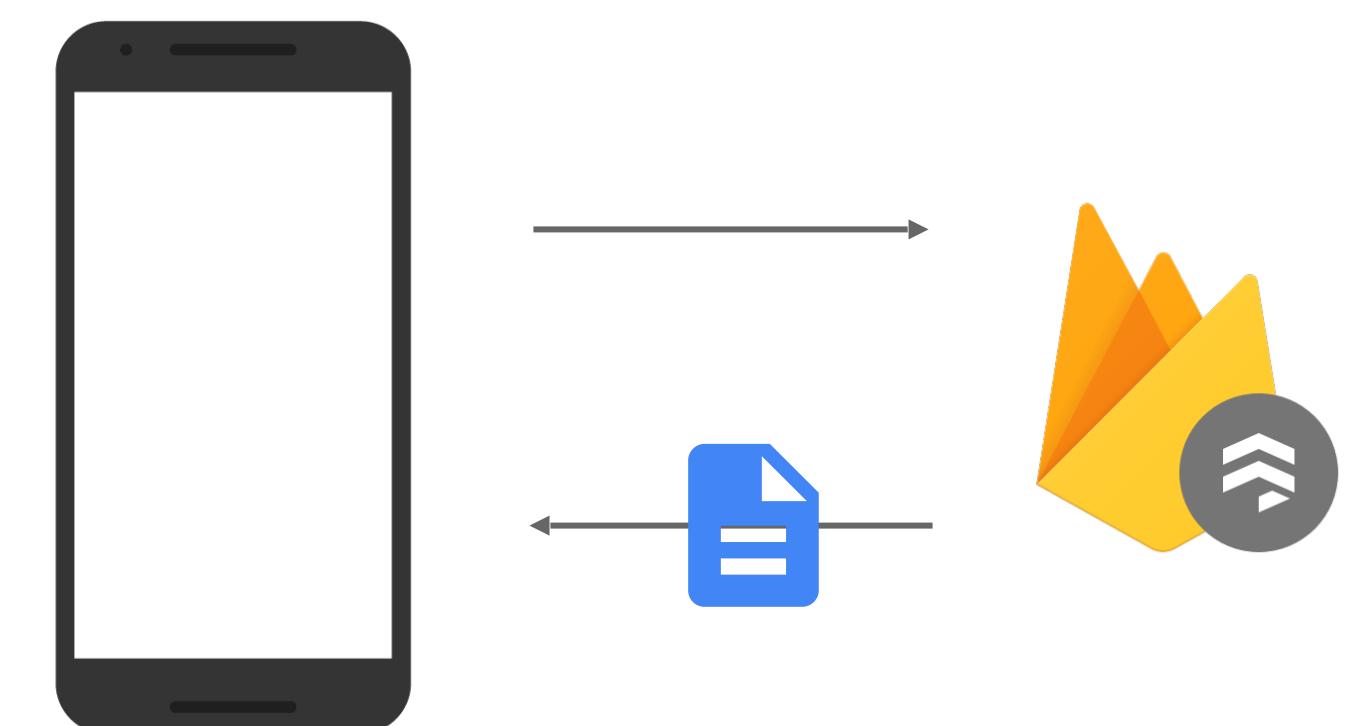
Cloud Firestore



Effortless Syncing



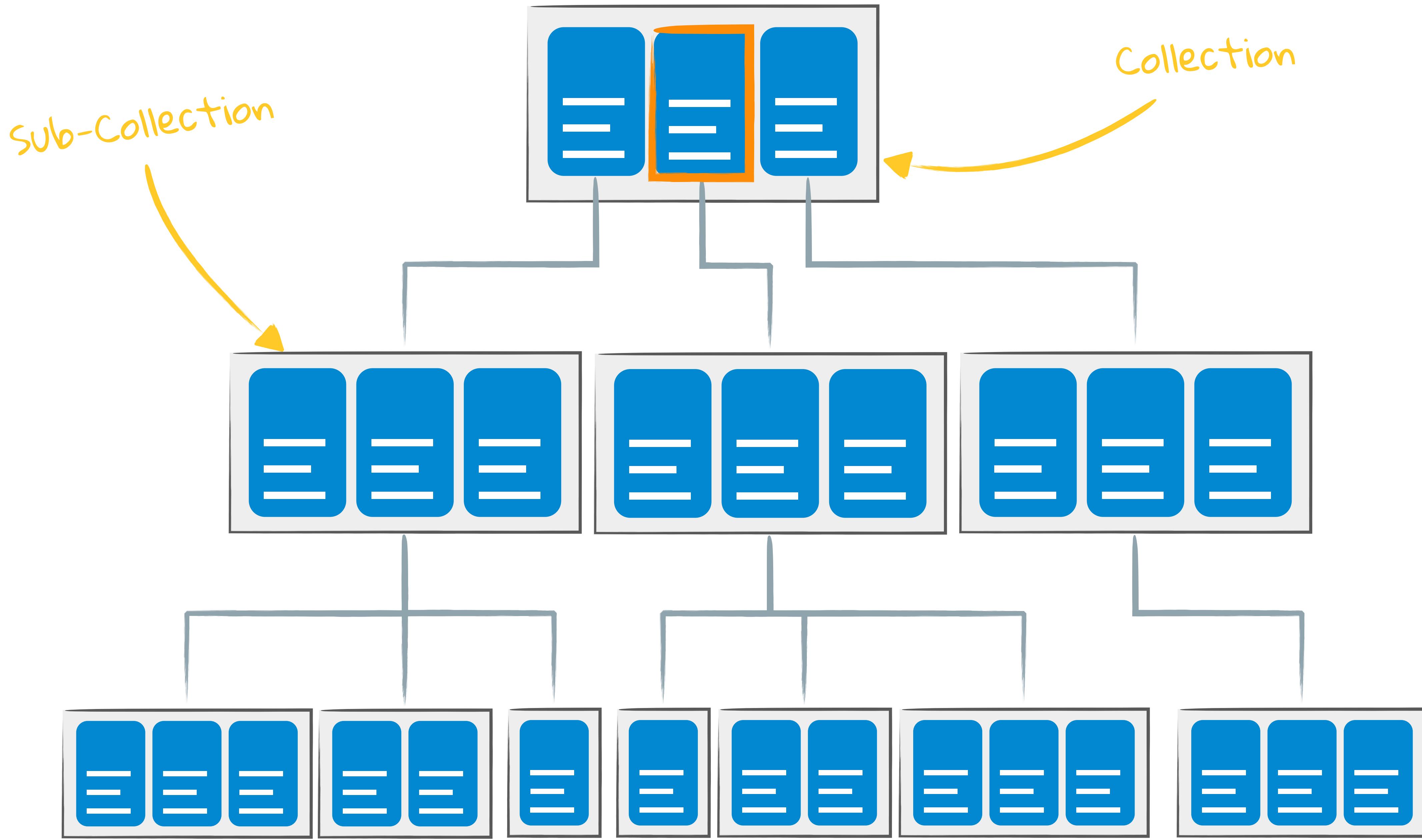
Offline Mode



One-Time Fetches

Document

```
bird_type: "swallow"
airspeed: 42.733
coconut_capacity: 0.62
isNative: false
icon: <binary data>
vector:
  { x: 36.4255,
    y: 25.1442,
    z: 18.8816 }
distances_traveled: [42, 39, 12, 42]
```



Top level
collections

"articles" collection

a single article document

The screenshot shows the Google Cloud Firestore interface. On the left, under the 'peterfriese-cf-workshop' database, there is a 'Start collection' button and a list of collections including 'articles'. The 'articles' collection is selected, highlighted with a yellow border. A yellow arrow points from the text 'Top level collections' to this selection. In the center, the 'articles' collection is shown with a list of document IDs: 3yHPKbG5ns32W8bVM5Jm, V3beVkB501W2aef5us5Y, VPDcwE4ivPeQa2aAU86T, cP8iYjBdCWAYnAihdBut, cqWRKDgRg35moyg6csWo, dz1Y05V8qj9qxJ9eXLnJ, jOUXvwENWab8vKjyeFQ1, nxezMSVARH31S5VLnEr1, ocxPG7ykbF7ryBX07DC5, and pKWus0VYlphaDSQzUf6i. The document 'pKWus0VYlphaDSQzUf6i' is selected and expanded, shown on the right. A yellow arrow points from the text '"articles" collection' to this expansion. The document details are as follows:

```
author: "Jamie Grierson"  
dateAdded: May 10, 2023 at 11:14:06 PM UTC+2  
datePublished: "2023-05-10T17:07:38.000Z" (string)  
excerpt: "Clock dials atop Elizabeth Tower in London briefly stopped at 12.55pm on Wednesday"  
imageUrl: "https://i.guim.co.uk/img/media/45256a86482d18545a97df31a9e450a  
width=1200&height=630&quality=85&auto=format&fit=crop&overlay-align  
base64=L2ltZy9zdGF0aWMvb3ZlcmxheXMvdGctZGVmYXVsdC5wbmc&  
isRead: false  
isStarred: false  
notes: ""  
readableHTML: "<div class="article-body-commercial-selector article-body-viewer-selector dcr-18i1c38"><p class="dcr-n6w1lc">Time stood still in <a href="https://www.theguardian.com/uk/london">London</a> on Wednesday, it has emerged.</p><p class="dcr-n6w1lc">Big Ben, the famous bell at the top of the 96-metre Elizabeth Tower in Westminster, failed to chime when the dials of the clock briefly stopped.</p><p class="dcr-n6w1lc">The four clock dials, one on each side of the tower in the <a href="https://www.theguardian.com/uk-news/palace-of-"
```

Data Model

```
public struct Article: Identifiable, Codable {  
    @DocumentID  
    public var id: String?  
  
    public var title: String  
    public var author: String?  
    public var readingTime: Int?  
    public var url: String  
    public var imageUrl: String?  
    public var siteName: String  
    public var dateAdded: Date  
    public var excerpt: String?  
    public var notes: String?  
    public var isRead: Bool? = false  
    public var isStarred: Bool? = false  
    public var userId: String?  
    public var readableHTML: String? = nil  
}
```

The screenshot shows a document detail screen with the following fields:

- Document ID:** pKWus0VYIphaDSQzUf6i
- Collection:** art collection
- Add field** button
- author:** Jamie Grierson
- dateAdded:** May 10, 2023 at 11:14:06 PM UTC+2
- datePublished:** "2023-05-10T17:07:38.000Z" (string) edit delete
- excerpt:** "Clock dials atop Elizabeth Tower in London briefly stopped at 12.55pm on Wednesday"
- imageUrl:** "https://i.guim.co.uk/img/media/45256a86482d18545a97df31a9e450a... width=1200&height=630&quality=85&auto=format&fit=crop&overlay-align=bottom-left&w=1200&h=630" base64=L2ltZy9zdGF0aWMvb3ZlcmxheXMvdGctZGVmYXVsdC5wbmc8...
- isRead:** false
- isStarred:** false
- notes:** ""
- readableHTML:** (Large block of HTML text describing the event)

Cloud Firestore Triggers



Execute a Cloud Function for the following events:

- onCreate(change, context)
- onUpdate(change, context)
- onDelete(change, context)
- onWrite(change, **context**)

```
const uid = context.auth.uid;
const name = context.auth.token.name || null;
const picture = context.auth.token.picture || null;
const email = context.auth.token.email || null;
```

Store readable HTML for link



10:48

THE NEW YORKER

Subscribe

NEW YORKER FAVORITES "Cat Person" Mae West, Queen

THE FRONT ROW

THE 2023 OSCAR NOMINATIONS, AND WHAT SHOULD HAVE MADE THE LIST

These selections highlight the gap between what's on the Academy's radar and what's going on in the world of movies at large.

By Richard Brody
January 24, 2023

Become a New Yorker subscriber.
12 weeks for \$29.99 \$6. [Subscribe now](#)

AA newyorker.com

< > ↗ ↘ ↙ ↖

10:47

Inbox



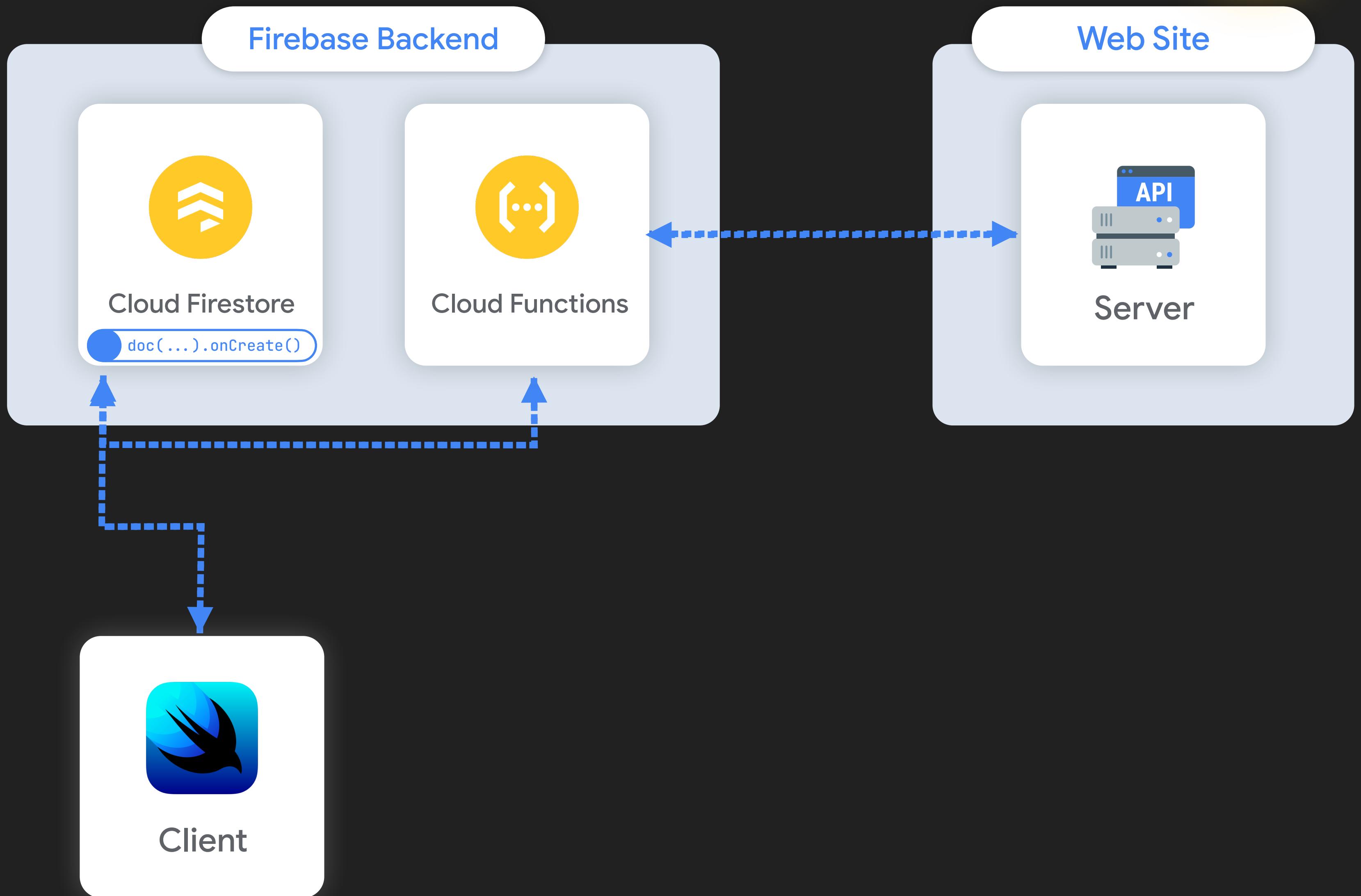
The 2023 Oscar Nominations, and What Should Have Made the List

NEWYORKER.COM

If Oscar nominations bore a movie's title, this year's rundown would borrow one from Stanley Kubrick's first feature, "Fear and Desire." The movie-industry news of the past year has been the collapse of the box-office for almost everything but blockbusters, and the Academy's response has been to put its mouth where its money

✉️ 📄 📁 🎫 📱 📲

Store readable HTML for link



Store readable HTML for link



```
export const storeReadableLink =  
functions.firebaseio  
  .document("artifacts/{documentId}")  
  .onCreate(async (documentSnapshot) => {  
  
  const url = documentSnapshot.data().url;  
  const metadata = await parser.parse(url);  
  
  const artifactDocument = documentSnapshot.data() as ArtifactDocument;  
  
  if (!artifactDocument.readableHTML && metadata.content) {  
    artifactDocument.readableHTML = metadata.content;  
  }  
  
  // update the document in Firestore  
  return documentSnapshot.ref.update(artifactDocument);  
});
```

Chapter 3

Callable Functions



Callable Functions

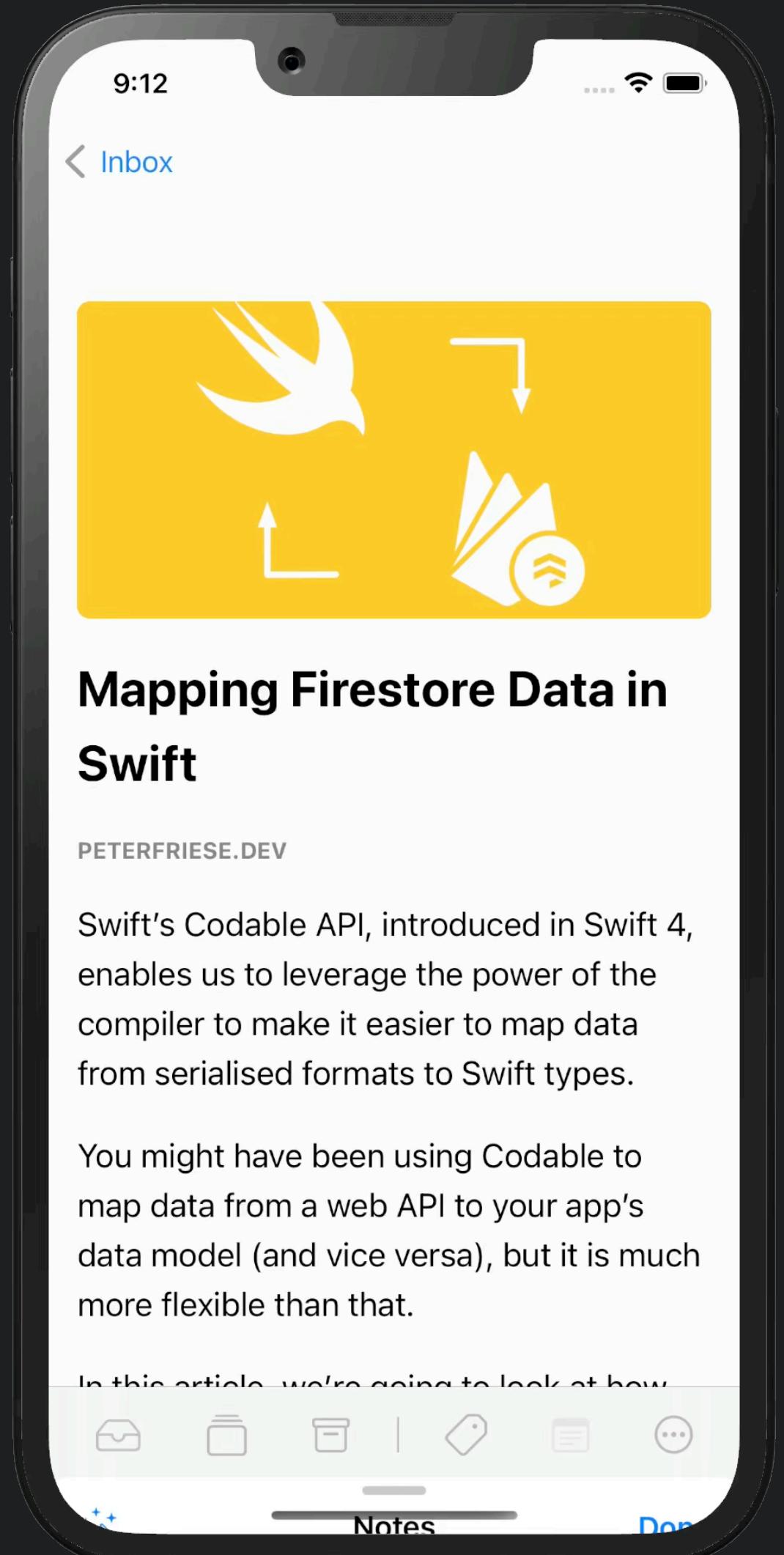


Call a Cloud Function directly from your app

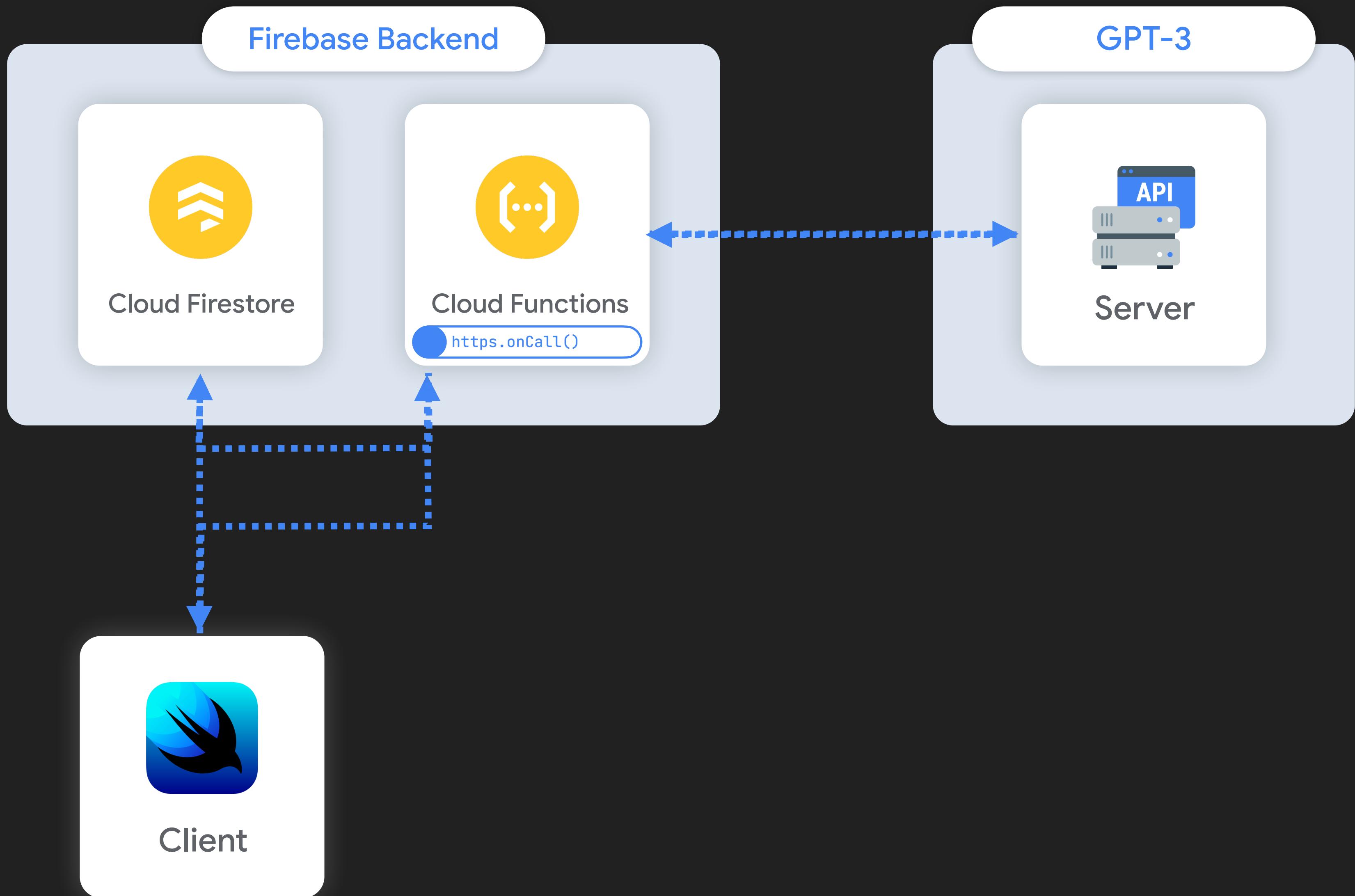
- onRequest(request, response)
- onCall(data, context)

```
const uid = context.auth.uid;
const name = context.auth.token.name || null;
const picture = context.auth.token.picture || null;
const email = context.auth.token.email || null;
```

Summarise article



Summarise article



Summarise article



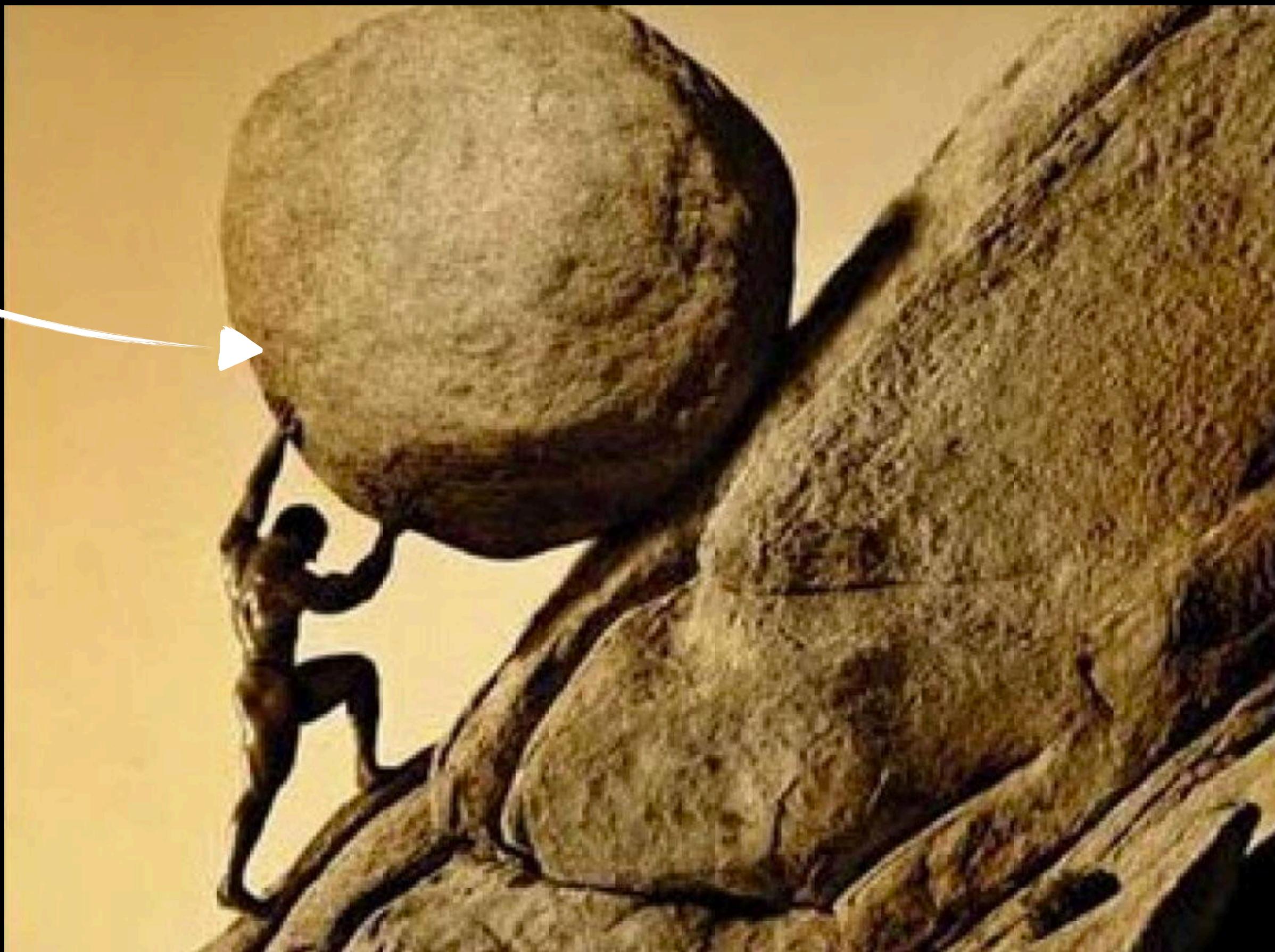
```
export const summarise =  
functions.https  
  .onCall(async (data) => {  
    const url = data;  
  
    const completion = await openai.createCompletion({  
      model: "text-davinci-001",  
      prompt: `Summarize this article: ${url}`,  
    });  
  
    const summary = completion.data.choices[0].text;  
  
    return summary;  
  });
```

Chapter 4

Firebase Extensions



*Do we need to implement
everything ourselves?*





Firebase Extensions





Firebase Extensions

- Pre-built packages of functionality
- Based on Cloud Functions
- One-click install to Firebase project
- Configurable

Search extensions

Categories

- Marketing
- Messaging
- Payments
- Search
- Shipping
- Social
- Utilities

Providers

- Algolia
- Elastic
- Firebase
- Google Pay
- Invertase
- Jigsaw
- Mailchimp
- Meilisearch
- MessageBird
- Purchasely
- RevenueCat
- ShipEngine

Search with Algolia

Made by [Algolia](#)

Enables full text search of your Cloud Firestore data with Algolia.

Search

Install

View details

Stream Collections to BigQuery

Made by [Firebase](#)

Sends realtime, incremental updates from a specified Cloud Firestore collection to BigQuery.

Utilities

Install

View details

Shorten URLs

Made by [Firebase](#)

Shortens URLs written to a specified Cloud Firestore collection (uses Bitly).

View details



Search App Sync

Made by [Elastic](#)

Syncs documents collection to Elastic enable full-text sea

Search

Install

View details



Distribute

Made by [Elastic](#)

Records event count accommodate high Cloud Firestore.

Utilities

Install

View details

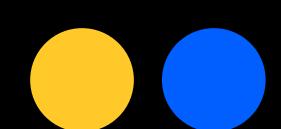


Translate

Made by [Elastic](#)

Translates strings Firestore collection languages (uses C

Wrap-up



Find extensions to build an app quickly and easily

[Explore extensions](#)

[View docs !\[\]\(04174670108811f2fbba5bcdabcf46f4_img.jpg\)](#)

Popular extensions



Stream Collections to BigQuery

Made by [Firebase !\[\]\(27ef29dc8b4e394bb982c545d26a8abb_img.jpg\)](#)

Sends realtime, incremental updates from a specified Cloud Firestore collection to BigQuery.

[Install](#)

[View details](#)



Search with Algolia

Made by [Algolia !\[\]\(8b80316e8e650f6fed9b8156545797c1_img.jpg\)](#)

Enables full text search of your Cloud Firestore data with Algolia.

[Install](#)

[View details](#)



Run Payments with Stripe

Made by [Stripe !\[\]\(dcbcccd837f83908a989a862923635c7_img.jpg\)](#)

Controls access to paid content by syncing your one-time and recurring payments with Firebase Authentication.



Trigger Email

Made by [Firebase !\[\]\(9abfec5139e94a78ef306d7e4e437e55_img.jpg\)](#)

Composes and sends an email based on the contents of a document written to a specified Cloud Firestore...



Why extensions



Ease of use

Install easily and quickly with no maintenance required

[View docs](#)

Productivity

Extend your app's functionality without additional research

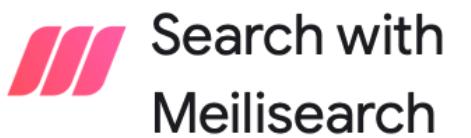
[View all extensions](#)

Showcase

Discover Kara's Coffee, an e-commerce app built using Firebase Extensions

[View on GitHub](#)

Extension spotlight ⓘ | [View all extensions](#)



Search with Meilisearch

Made by [Meilisearch](#) ✅

Full-text Search on Firebase with Meilisearch

[Install](#)[View details](#)

Firestore Bundle Builder

Made by [Firebase](#) ✅

Caches publicly available documents in Cloud Firestore data bundles that are served from either Firebase Hosting...

[Install](#)[View details](#)

Run In-App Purchases &...

Made by [Purchasely](#) ✅

Quickly implement & seamlessly manage your iOS & Android In-App Purchases & In-App Subscriptions with...

[Install](#)[View details](#)

Create Multiparty Video Calls

Made by [Vonage](#) ✅

Quickly build sophisticated live video apps with many participants.

[Install](#)[View details](#)

Image Processing API

Made by [Invertase](#) ✅

Use this extension to optimize & transform images via a powerful API with over 30 different image...

[Install](#)[View details](#)

Why extensions



One more thing...

Search with Meilisearch

Made by [Meilisearch](#)

Full-text Search on Firebase with Meilisearch

[Install](#)[View details](#)

Firestore Bundle Builder

Made by [Firebase](#)

Caches publicly available documents in Cloud Firestore data bundles that are served from either Firebase Hosting...

[Install](#)[View details](#)

Run In-App Purchases &...

Made by [Purchasely](#)

Quickly implement & seamlessly manage your iOS & Android In-App Purchases & In-App Subscriptions with...

[Install](#)[View details](#)

Create Multiparty Video Calls

Made by [Vonage](#)

Quickly build sophisticated live video apps with many participants.

[Install](#)[View details](#)

Image Processing API

Made by [Invertase](#)

Use this extension to optimize & transform images via a powerful API with over 30 different image...

[Install](#)[View details](#)

Search with Meilisearch

Made by [Meilisearch](#) 

Full-text Search on Firebase with Meilisearch

[Install](#)

[View details](#)

Firebase Bundles Builder

Made by [Firebase](#) 

Caches publicly available documents in Cloud Firestore data bundles that are served from either Firebase Hosting...

[Install](#)

[View details](#)

In-App Purchases &...

Made by [Purchasely](#) 

Quickly implement & seamlessly manage your iOS & Android In-App Purchases & In-App Subscriptions with...

[Install](#)

[View details](#)

Create Multi-party Video Calls

Made by [Vonage](#) 

Quickly build sophisticated live video apps with many participants.

[Install](#)

[View details](#)

Image Processing API

Made by [Invertase](#) 

Use this extension to optimize & transform images via a powerful API with over 30 different image...

[Install](#)

[View details](#)

Build your own extension

Join the Extensions provider Alpha Group to build, publish, and share with the Firebase Community

Step 1

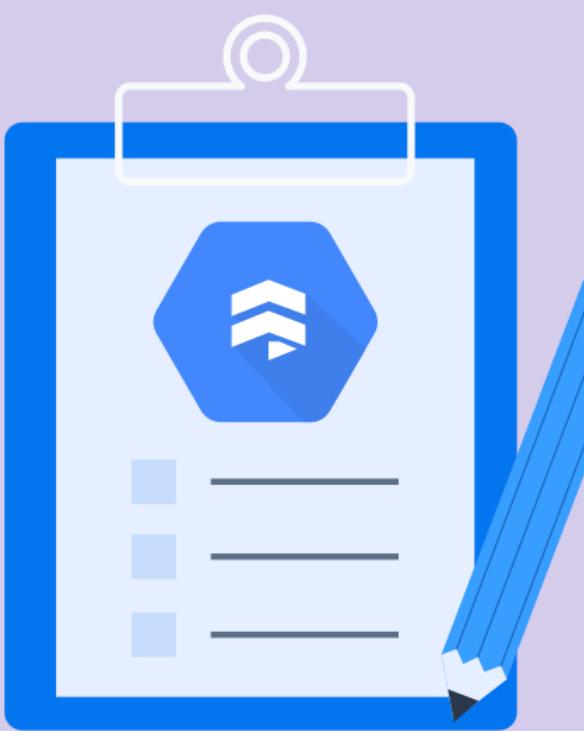
Join the [Alpha Group](#) 

Step 2

Build an extension

Step 3

Publish your extension



Try Firebase today

Integrating it into your app is easy.

Search with
Meilisearch

Made by [Meilisearch](#) 

Full-text Search on Firebase with
Meilisearch

[Install](#)

[View details](#)

Firebase Database
Builder

Made by [Firebase](#) 

Caches publicly available documents in
Cloud Firestore data bundles that are
served from either Firebase Hosting...

[Install](#)

[View details](#)

In-App App
Purchases &...

Made by [Purchasely](#) 

Quickly implement & seamlessly
manage your iOS & Android In-App
Purchases & In-App Subscriptions wit...

[Install](#)

[View details](#)

Create Multi-party
Video Calls

Made by [Vonage](#) 

Quickly build sophisticated live video
apps with many participants.

[Install](#)

[View details](#)

Image Processing API
Made by [Invertase](#) 

Use this extension to optimize >
transform images via a powerful
API with over 30 different image...

[Install](#)

[View details](#)

Build your own extension

Join the Extensions provider Alpha Group to build, publish, and share with the Firebase Community

Step 1

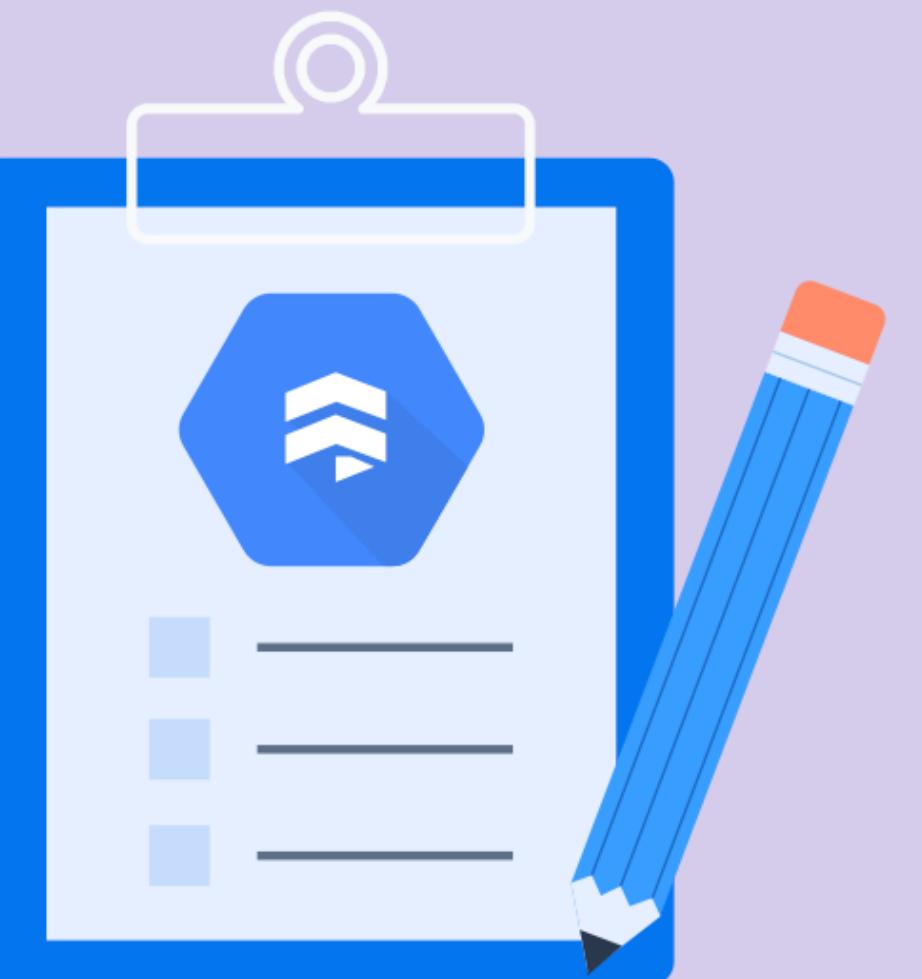
Join the [Alpha Group](#) 

Step 2

Build an extension

Step 3

Publish your extension



Try Firebase today

Integrating it into your app is easy.

Building an Extension

- Install Firebase tools (CLI)
- Write Cloud Functions (triggers)
- Package Extension

extension.yaml

```
name: auth-chat
version: 0.2.0
specVersion: v1beta
displayName: Authenticate with Stream Chat
description: Synchronizes Firebase Authentication users with Stream, and creates and revoke Stream Chat authentication tokens.
license: Apache-2.0
author:
  authorName: Stream
  url: https://getstream.io/
sourceUrl: https://github.com/GetStream/stream-firebase-extensions/tree/main/auth-chat
billingRequired: true
params:
  - param: LOCATION
    label: Cloud Functions location
    description: >
      Where do you want to deploy the functions created for this extension?
type: select
options:
```



index.ts

```
import * as admin from "firebase-admin";
import * as functions from "firebase-functions";
import { StreamChat } from "stream-chat";

admin.initializeApp();

const serverClient = StreamChat.getInstance(
  process.env.STREAM_API_KEY!,
  process.env.STREAM_API_SECRET!,
);

// When a user is created in Firebase an associated Stream account is also created.
export const createStreamUser = functions.auth.user().onCreate(async (user) => {
  functions.logger.log("Firebase user created", user);

  // Create user using the serverClient.
  const response = await serverClient.upsertUser({
    id: user.uid,
    name: user.displayName,
    email: user.email,
    image: user.photoURL,
  });
}
```

index.ts

```
const serverClient = StreamChat.getInstance(  
  process.env.STREAM_API_KEY!,  
  process.env.STREAM_API_SECRET!,  
);  
  
// When a user is created in Firebase an associated Stream account is also created.  
export const createStreamUser = functions.auth.user().onCreate(async (user) => {  
  functions.logger.log("Firebase user created", user);  
  
  // Create user using the serverClient.  
  const response = await serverClient.upsertUser({  
    id: user.uid,  
    name: user.displayName,  
    email: user.email,  
    image: user.photoURL,  
  });  
  
  functions.logger.log("Stream user created", response);  
  return response;  
});
```

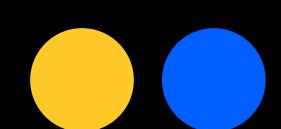
Thanks



@zamzampoooya



@peterfriese



Feedback



Cloud Functions ?!

