

# Make smarter decisions faster with Firebase

## Remote Config

Mangirdas Kazlauskas, Daria Orlova  
Average cat enjoyers  
@mkobuolys | @dariadroid

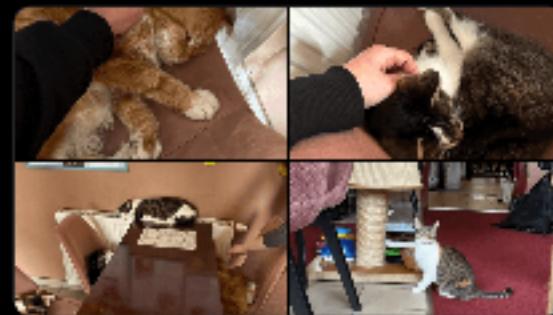
Daria ❤️ @dariadroid • Aug 14, 2022

We spent the weekend in a University. I hope I meet with you because in Latvia 😊. And there are many more friends with them! I've met at least 5 of them! And my version of "choose yourself with one photo" is here 😊



Daria ❤️ @dariadroid • Mar 10

Replying to @mkobuolis  
I was!! And I patted them all of course 😊😊



Daria ❤️ @dariadroid • Mar 10

Finally I had to go to eat cakes in several countries, but they were always full. Finally got into one 😊. Knows is the build 😊😊



Daria ❤️ @dariadroid • Jan 8

All you need to know about my love for cats 😊. A plushie that you can do just with experiments! I just couldn't walk by 😊. All the friends with Daria 😊😊





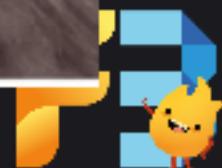
@mkobuolys | @dariadroid



@mkobuolys | @dariadroid



@mkobuolys | @dariadroid



@mkobuolys | @dariadroid



@mkobuolys | @dariadroid



@mkobuolys | @dariandroid



@mkobuolys | @dariadroid

X-SCIENCE ▾

## "Dog people" and "cat people" have different personality traits, study finds

BY RACHAEL RETTNER

MAY 29, 2016 / 4:40 PM / LIVESCIENCE.COM

[f](#) [t](#) [p](#)

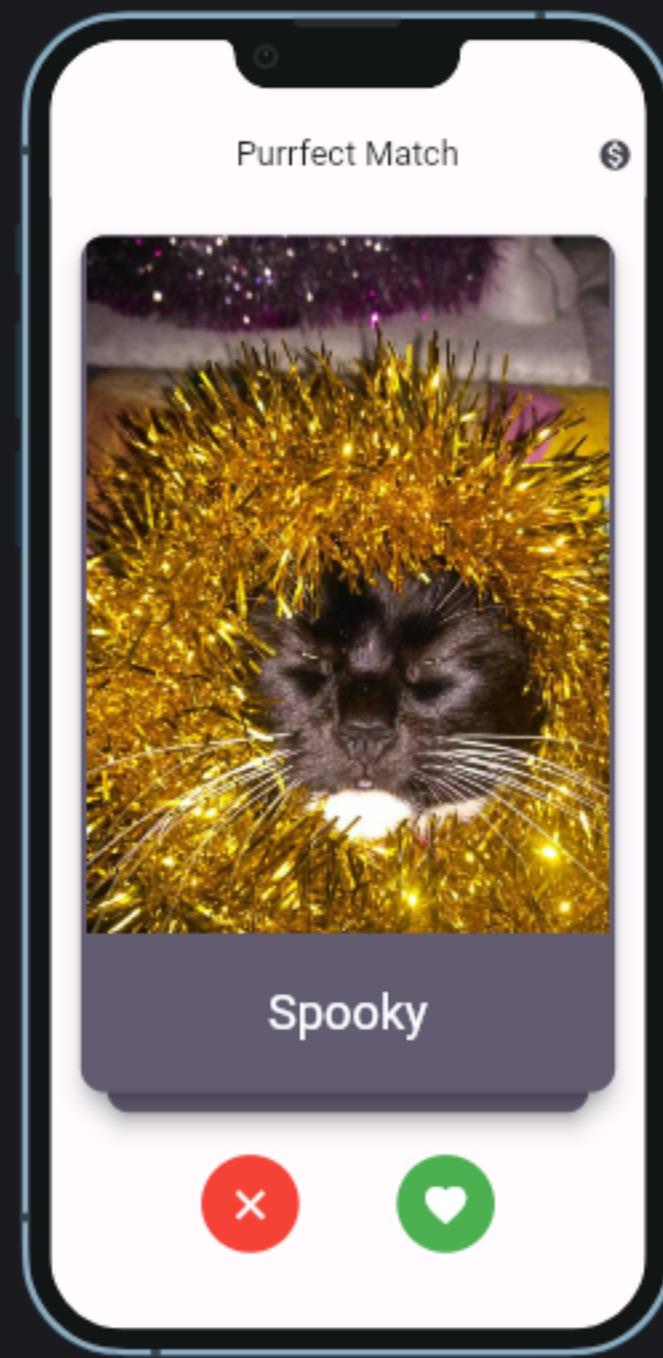
"Dog people" and "cat people" really do have different personalities, according to a new study.

People who said they were dog lovers in the study tended to be more lively -- meaning they were more energetic and outgoing -- and also tended to follow rules closely. Cat lovers, on the other hand, were more introverted, more open-minded and more sensitive than dog lovers. Cat people also tended to be non-conformists, preferring to be expedient rather than follow the rules.

# Purrfect match



@mkobuolys | @dariadroid



Lesson #1

# Making smart decisions with Analytics

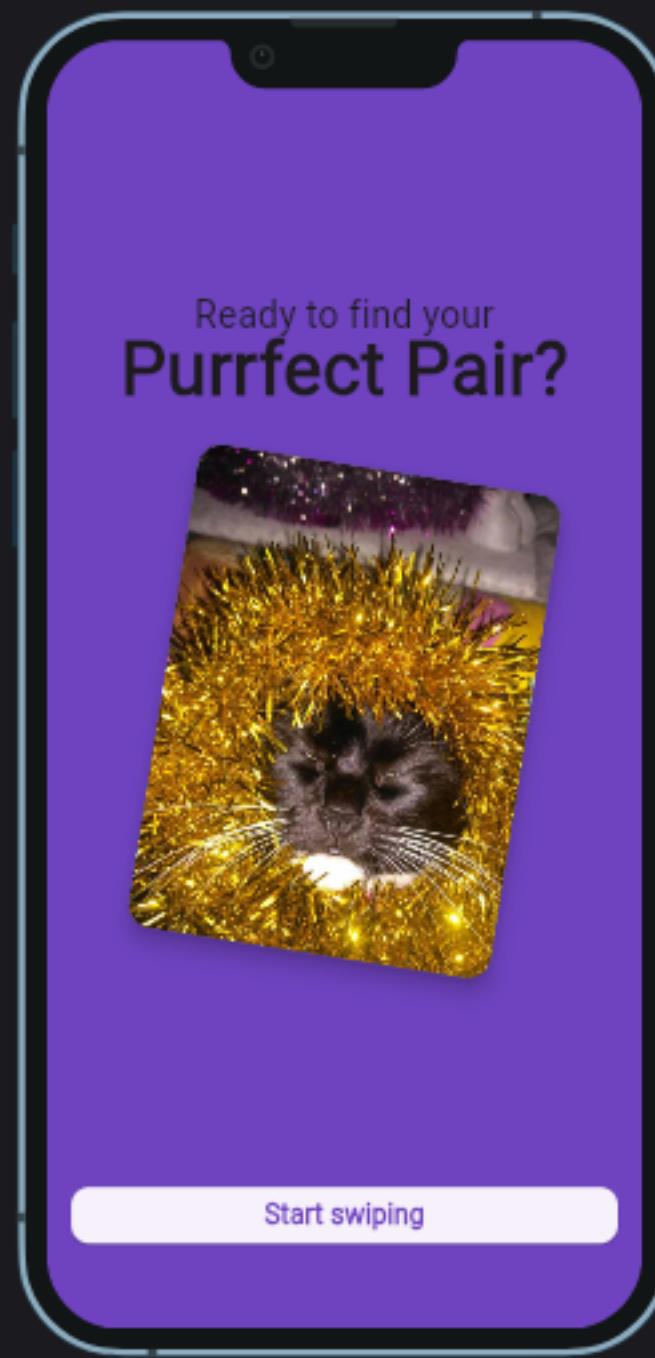


What is the first  
thing that the user  
sees?



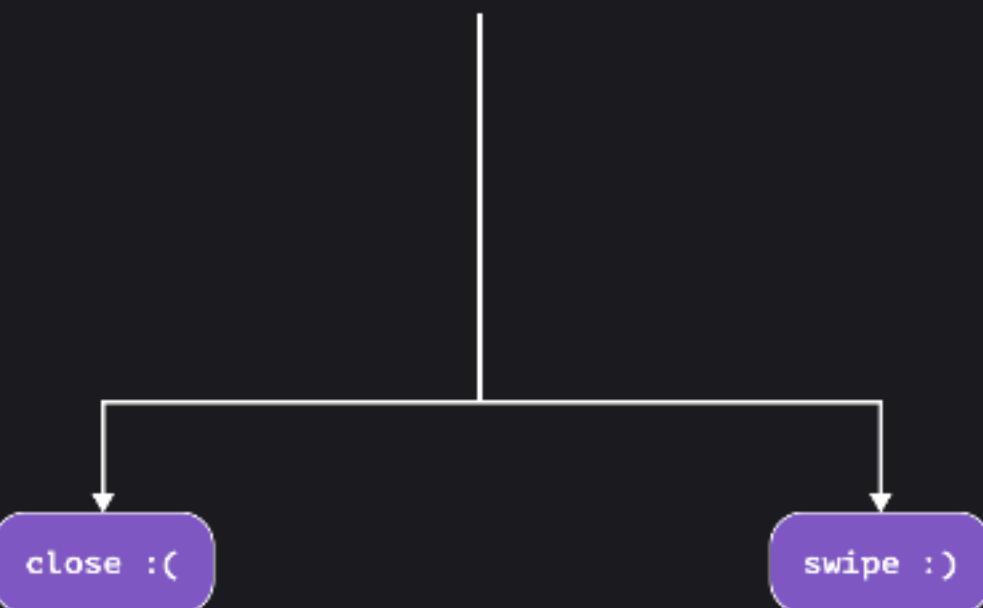
What is the first  
thing that the user  
sees?

# Onboarding

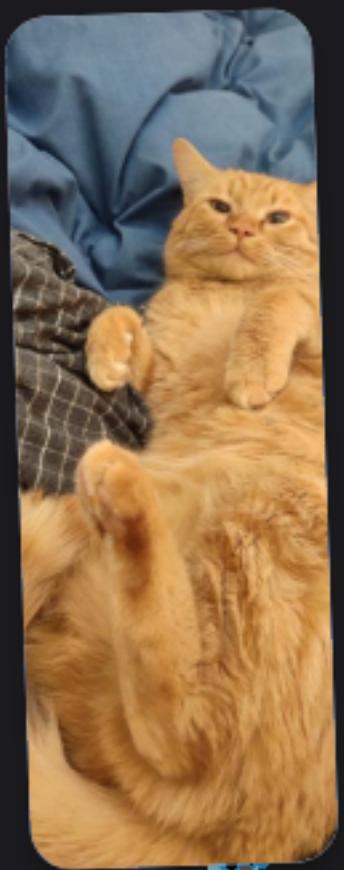
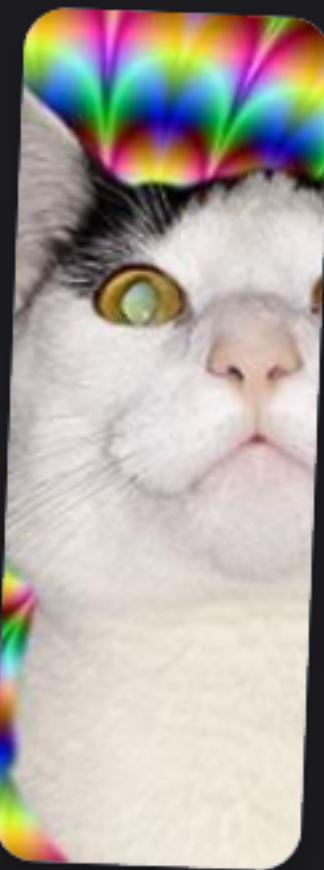
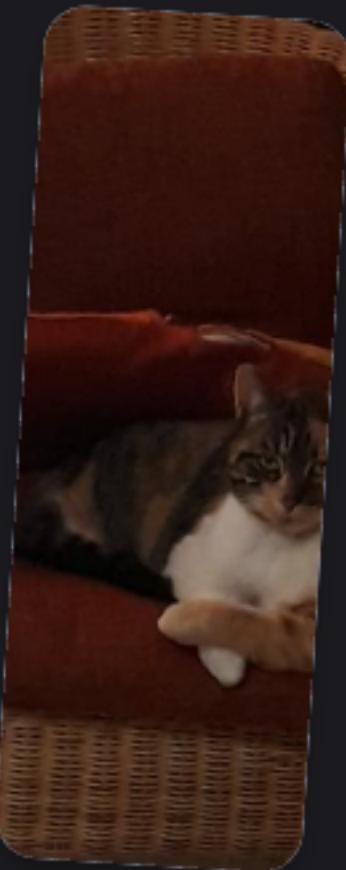


What is the first thing that the user sees?

# Onboarding



# Which cat performs best?



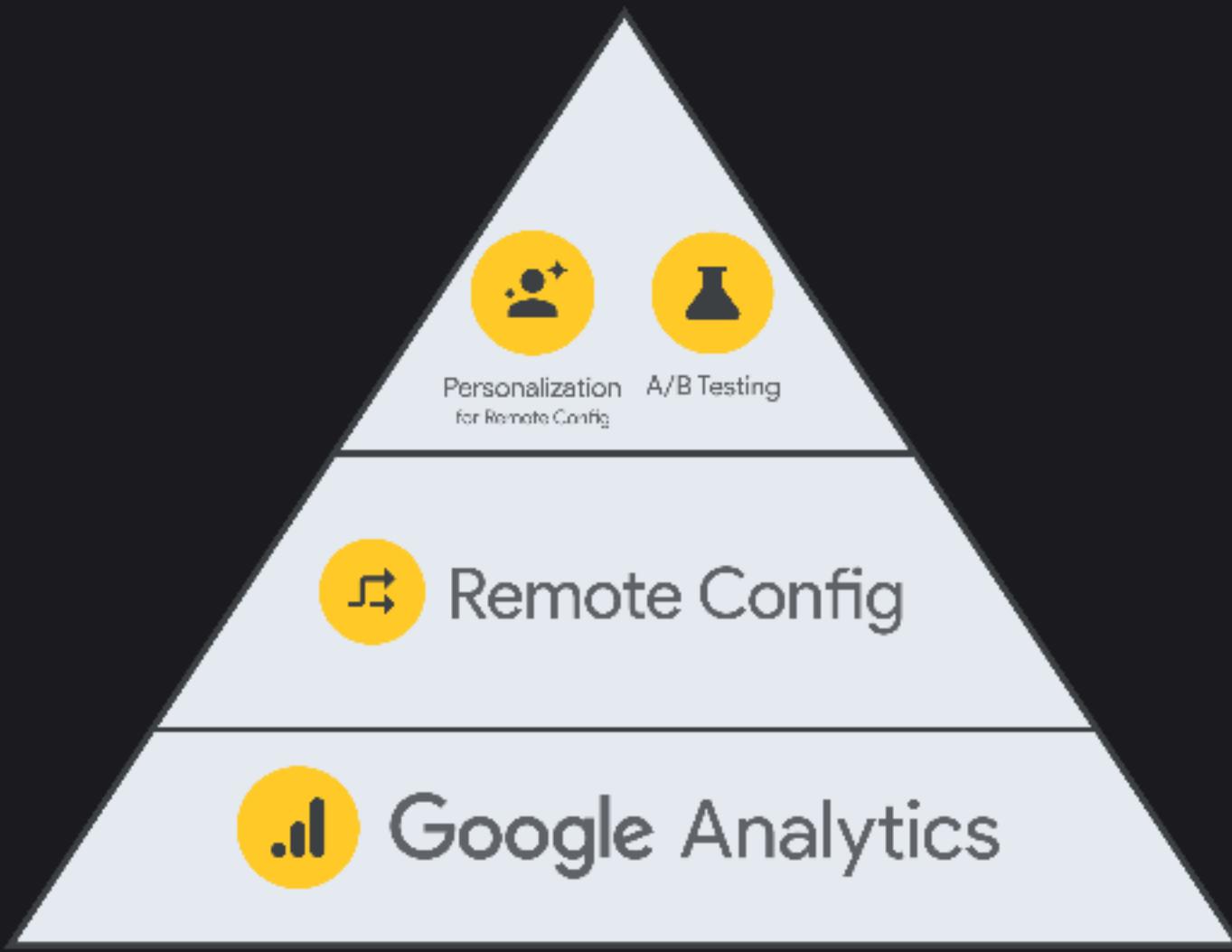
# How can we find out?



# How can we find out?

. Google Analytics





# Firebase Pyramid



The Foundation  
Google Analytics



X Create a project (Step 2 of 3)

## Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, In-app Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics modules

- Advertising
- Attribution
- User Acquisition & targeting services
- Firebase products
- Conversion reporting
- Enable Google Analytics for this project

Previous Continue



# Step 1: Enable Google Analytics for your project



```
flutter pub add firebase_analytics
```

Step 2: Add the dependency to your  
project



notification_foreground (app)	when a notification sent by FCM is received while the app is in the foreground	message_name, message_time, message_device_time, message_id, topic, label, message_channel, message_type
notification_open (app)	when a user opens a notification sent by FCM	message_name, message_time, message_device_time, message_id, topic, label, message_channel
notification_receive (app)	when a notification sent by FCM is received by a device when the app is in the background  Android only	message_name, message_time, message_device_time, message_id, topic, label, message_channel, message_type
os_update (app)	when the device operating system is updated to a new version. The previous operating system version id is passed as a parameter	previous_os_version
page_view (web)	each time the page loads or the browser history state is changed by the active site  Collected by default via <a href="#">enhanced measurement</a> .	page_location (page URL), page_referrer (previous page URL), engagement_time_msec
screen_view (app)	when a screen transition occurs and any of the following criteria are met: <ul style="list-style-type: none"> <li>• No screen was previously set</li> <li>• The new screen name differs from the previous screen name</li> <li>• The new screen-class name differs from the previous screen-class name</li> <li>• The new screen id differs from the previous screen id</li> </ul> ... and many more	firebase_screen, firebase_screen_class, firebase_screen_id, firebase_previous_screen, firebase_previous_class, firebase_previous_id, engagement_time_msec

Source: <https://support.google.com/analytics/answer/9234069>

Firebase automatically collects some events 

The screenshot shows the Firebase Analytics Events page. On the left, a sidebar lists various Firebase services: Remote Config, Analytics Dashboard, Events, Audience, DebugView, Conversations, and Accesses. Below these are sections for Build, Test Lab & Monitor, Analytics, and Realtime Database. Under Analytics, the 'Events' option is highlighted and circled in red. The main content area has a header 'Analytics | Events' with a search bar and date range filters ('Aug 24 - Sep 2019'). A blue banner at the top says 'Analyze user behavior across screens and measure in-app interactions'. Below this are two tabs: 'Events' (selected) and 'Custom Events'. The 'Recommended events' section contains three items: 'View' (Recommended event), 'Page' (Recommended event), and 'Status' (Recommended event). The 'Editing events' section lists several tracked events with their details and visibility toggles:

Event name	Event category	Event label	Event value	Visibility
onboarding	Onboarding	Onboarding	10	<input checked="" type="checkbox"/>
onboarding	Onboarding	Onboarding	5	<input type="checkbox"/>
onboarding	Onboarding	Onboarding	3	<input type="checkbox"/>
onboarded	Onboarding	Onboarded	10	<input type="checkbox"/>
onboarded	Onboarding	Onboarded	1	<input type="checkbox"/>
onboarded	Onboarding	Onboarded	5	<input type="checkbox"/>

# Step 3: Track events in the Firebase console



```
FirebaseAnalytics.instance.logEvent(  
    name: 'start_swiping',  
    parameters: {  
        'cat_index': index,  
    },  
);
```

## Step 4: Log custom events and parameters



Option 1: Release new app  
version every week with new  
index



# Option 1: Release new app version every week with new index

1. Change value



# Option 1: Release new app version every week with new index

1. Change value
2. Deploy to stores



# Option 1: Release new app version every week with new index

1. Change value
2. Deploy to stores



# Option 1: Release new app version every week with new index

1. Change value
2. Deploy to stores
3. Monitor metrics
4. Track analytics for one week



# Option 1: Release new app version every week with new index

1. Change value
2. Deploy to stores
3. Track analytics for one week
4. Repeat 5 times

Firebase Analytics helps us to make  
***smarter*** decisions...



Firebase Analytics helps us to make  
***smarter*** decisions...

but how to make them ***faster***?





# Firebase Remote Config

## Option 2

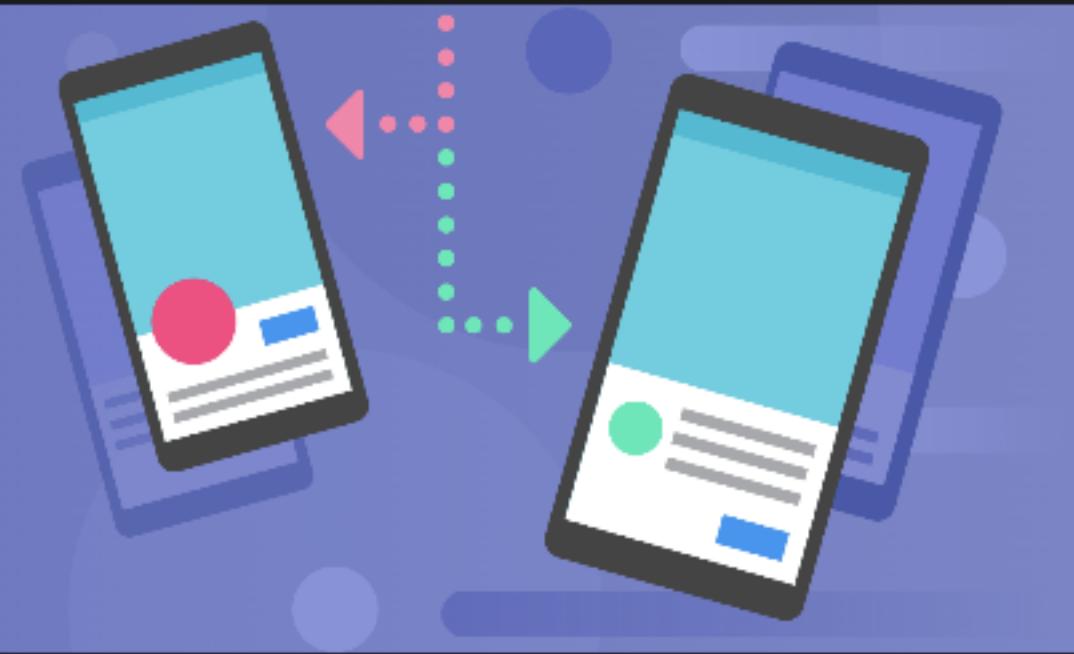


## Remote Config

Customize and experiment with app behavior using server-side configuration parameters & feature flags

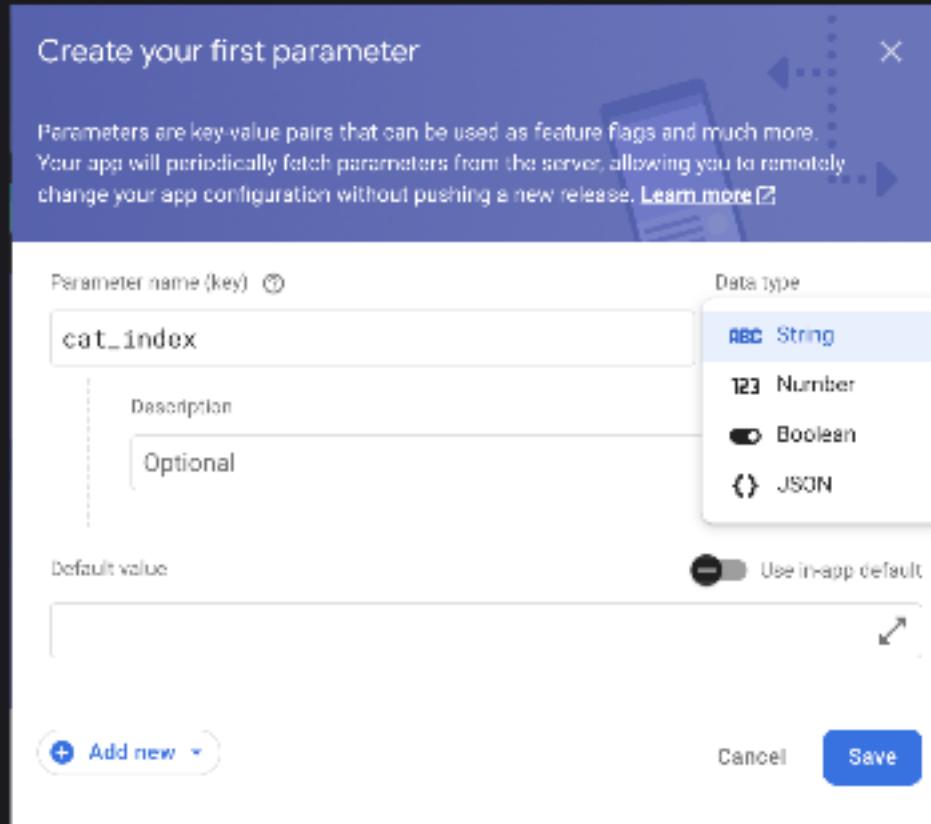
[Create configuration](#)

Or upload template



Key-Value pairs that live in the  
cloud





# Create a remote param: string, number, boolean or json



The screenshot shows the Firebase Remote Config interface. On the left, there's a sidebar with navigation links like Firebase, Remote Config, Analytics, and Crashlytics. The main area is titled 'Remote Config' and shows a table of parameters. One parameter, 'RNG AGCM20 logs', is expanded, revealing four variants: 'baseline', 'medium', 'high', and 'super'. Another parameter, 'video\_cmaf.enabled', is also expanded, showing its current value as 'true'. A third parameter, 'advertising\_opt\_in', is partially visible at the bottom.

Parameter	Variant	Value	Last Update
RNG AGCM20 logs	baseline	0%	Sept 16, 2020
	medium	0%	
	high	0%	
	super	0%	
video_cmaf.enabled	true	0%	Sept 16, 2020
advertising_opt_in	available	0%	Sept 16, 2020

# Review all params in the Firebase Console



```
"parameterGroups": {  
    "Premium": {  
        "parameters": {  
            "paywall_type": {  
                "defaultValue": {  
                    "value": "simple"  
                },  
                "description": "Possible values: \"simple\", \"detailed\"",  
                "valueType": "STRING"  
            },  
            "video_call_enabled": {  
                "defaultValue": {  
                    "value": "false"  
                },  
                "conditionalValues": {  
                    "20% of users": {  
                        "value": "true"  
                    }  
                },  
                "description": "Feature flag for the video call feature.",  
                "valueType": "BOOLEAN"  
            }  
        }  
    }  
}
```

# Simple JSON under the hood



```
flutter pub add firebase_remote_config
```

Step 1: Add dependency to your  
project



## Create parameter

Parameter name (key) [?](#)

Data type

123 Number

Description

Possible values are in the range 1..5

Default value



Use in-app default

[+ Add new](#)

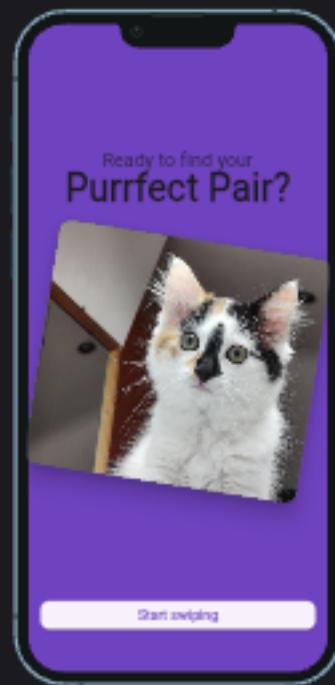
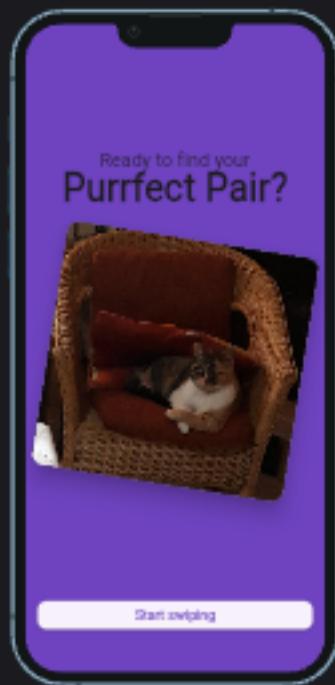
Cancel

Save

Step 2: Create param in Firebase  
console



```
final catIndex = FirebaseRemoteConfig.instance.getInt('onboarding_cat_index');
```



catIndex  
= 1

catIndex  
= 2

catIndex  
= 3

catIndex  
= 4

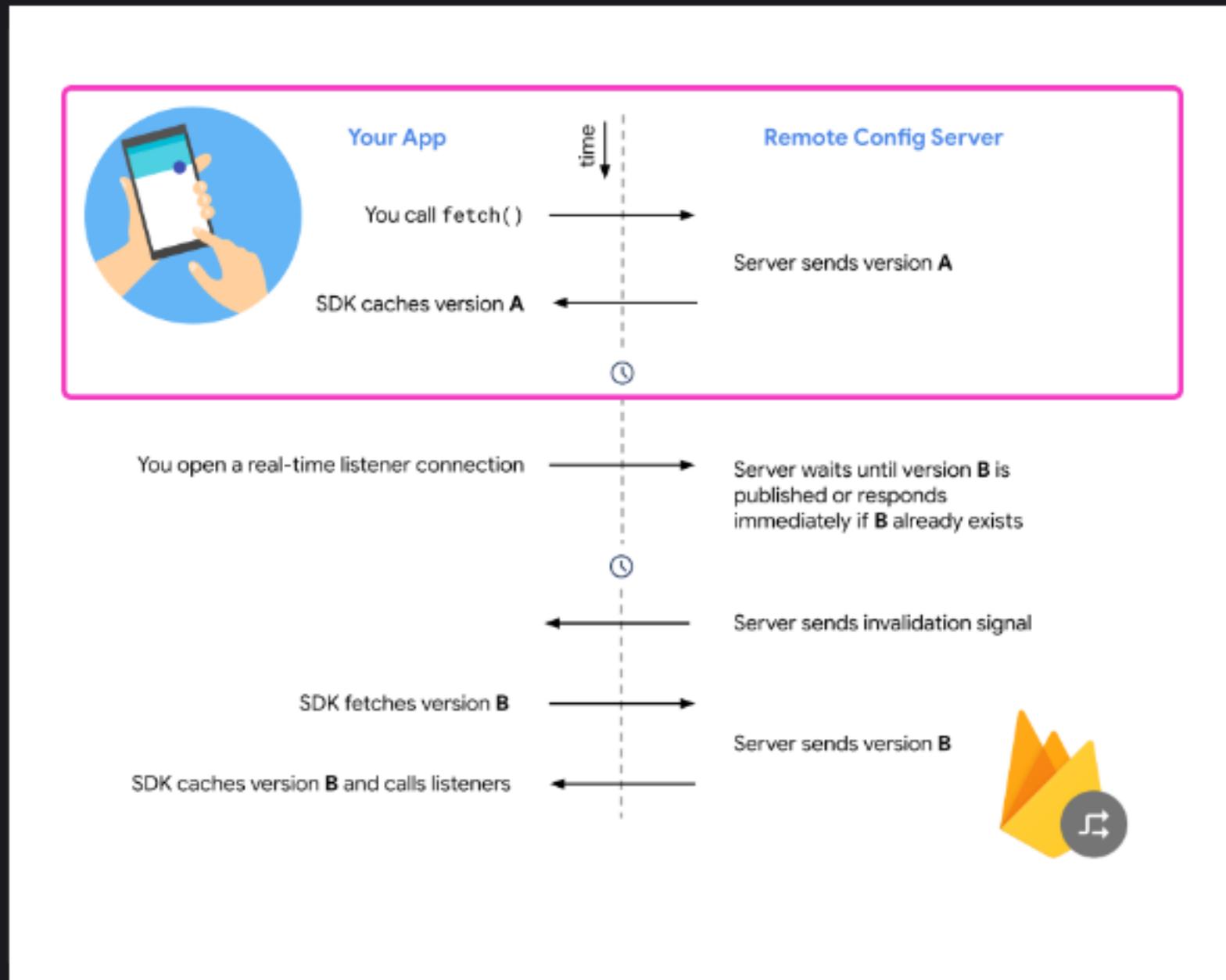
catIndex  
= 5



The screenshot shows the Firebase Remote Config interface. At the top, there are tabs for 'Permissions', 'Conditions', 'A/B Tests', and 'Personalizations'. Below these, a section titled 'Published changes' is shown. This section includes a 'Fetch + Load 2 hours' button, a 'No recent fetch data' message with a 'View log' link, and three cards: 'Add rules' (Customize your app for different users), 'Personalizations' (Personalize experiences to drive engagement), and 'Metrics' (Analyze user behavior). A 'Data entry' section at the bottom lists a single entry for 'Dark' with a 'Set index' checkbox and a note that values are in the range 1-2. The 'Publish changes' button at the top right is circled in red.

# Step 3: Publish changes on Firebase console





Source: <https://firebase.google.com/docs/remote-config/real-time>



```
Future<void> init() async {
    await firebaseRemoteConfig.ensureInitialized();
    await firebaseRemoteConfig.setConfigSettings(
        RemoteConfigSettings(
            fetchTimeout: const Duration(seconds: 10),
            minimumFetchInterval: const Duration(hours: 12),
        ),
    );
    await firebaseRemoteConfig.fetchAndActivate();
}
```

## Step 4: Fetch & activate initial values





Your App

You call fetch()

time

Remote Config Server

Server sends version A

SDK caches version A



You open a real-time listener connection



Server waits until version B is published or responds immediately if B already exists



Server sends invalidation signal

SDK fetches version B



Server sends version B



SDK caches version B and calls listeners



Source: <https://firebase.google.com/docs/remote-config/real-time>



```
FirebaseRemoteConfig.instance.onConfigUpdated.listen((event) async {  
    await FirebaseRemoteConfig.instance.activate();  
    // Do whatever to update your app  
});
```

## Step 5: Set listener to update values



## Edit parameter

Parameter name (key) ?

cat\_index

Data type

123 Number

Description

Possible values are in the range 1..5

Default value

1

Use in-app default

+ Add new ▾

Cancel

Save

Update cat index to 4

Ready to find your  
**Purrfect Pair?**



Start swiping



# But what if we want to limit the values



36

But what if we want to limit the values  
... to a specific *location* ?



But what if we want to limit the values  
... to a specific *location* ?  
... or *language* ?



But what if we want to limit the values  
... to a specific *location* ?  
... or *language* ?  
... or *date* ?



## Remote Config

Parameters    Conditions    A/B Tests    Personalizations

Add condition



No conditions have been created

Use conditions to change the value of parameters for defined sets of users

[View the docs](#)

Step 1: Open Conditions tab in

Remote Config



## Edit condition

Use conditions to provide different parameter values if a condition is met. Any changes made to this condition will apply to **all parameters** using this condition.

Name

Valentines Day Promo

Color

Pink

Applies if...

Date/time

Before

2/15/2023

00:00 ⓘ

Devic...

Date/time

After

2/13/2023

00:00 ⓘ

Devic...

and

Targeting estimations are not available for the selected condition types



Cancel

Save condition

## Step 2: Set date & time conditions



## Create parameter

X

Parameter name (key) ②

Data type

show\_valentines\_promo

Boolean

Description

Optional

Default value

Use in-app default

false

+ Add new ▾

Cancel

Save

# Step 3: Create a new FRC param



## Create parameter

X

Parameter name (key) ⓘ

show\_valentines\_promo

Data type

Boolean

Description

Optional

Default value

false

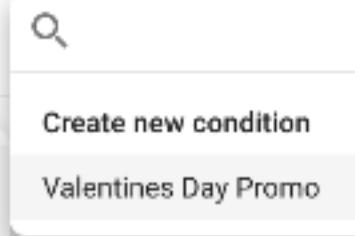
Use in-app default

↳ Conditional value

Send a value to a targeted group of app instances.

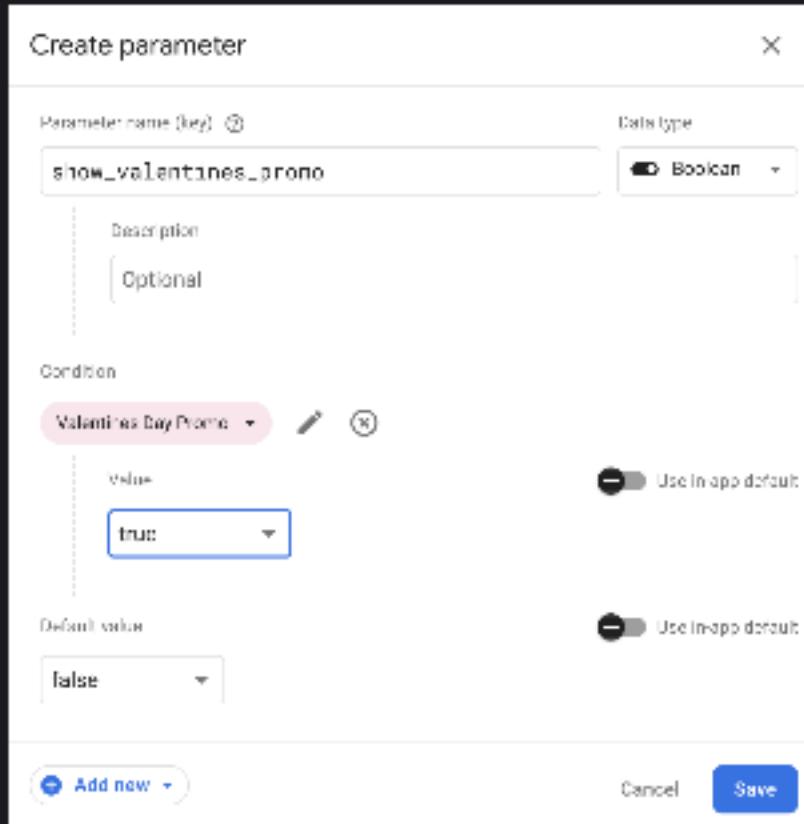
👤 Personalization

Let machine learning determine what value is best for what user.



# Step 3: Apply condition





# Step 3: Select value for this condition



## Remote Config

Parameters    Conditions    A/B Tests    Personalizations

Add condition



Search

### Valentines Day Promo

Date/Time before Feb 15, 2023, 12:00:00 AM User time zone

Date/Time after Feb 13, 2023, 12:00:00 AM User time zone

Last published Darja Orlova, 14 minutes ago

Affects parameter show\_valentines\_promo

The order of conditions determines their evaluation order. Parameter values are determined by the first true condition.

# Review conditions in the conditions tab



A screenshot of a software interface for configuring FRC parameters. At the top, there's a header with a dropdown, a checkbox, and a switch icon followed by the text "show\_valentines\_promo". Below this, a pink rounded rectangle contains the text "Valentines Day Promo". To the right of the rectangle, a slider is set to "true". Further to the right, the status "0%" is shown, along with the name "Daria Orlava" and the timestamp "Just now". On the far right are edit and info icons. Below the main row, there's another row labeled "Default value" with a slider set to "false" and a status "0%".

Review what conditions are applied  
to FRC params



## Define a new condition

Use conditions to provide different parameter values if a condition is met. Any changes made to this condition will apply to all parameters using this condition.

Name

Color

Exclusive promo for new iOS users in Latvia

Deep Orange

Applies if...

App	iOS - purfect_match (ios)
Country/Region	is in Latvia
First open	After 9/22/2023 2:00 PM Latvia Ti... and

Targeting estimations are not available for the selected condition types



Targeting first open time requires iOS SDK 9.0.0 and Android SDK 21.1.1 (Firebase BoM 30.3.0) or later

[Learn more](#) [Dismiss](#)

Cancel

Create condition

# Combine various conditions



## ⌘ Exclusive promo for new iOS users in Latvia

App: **iOS+ purrfect\_match (ios)**  
Country/Region: Latvia  
First open: after 10/1/23, 2:00 PM Europe/Riga

Last published: Darja Orlova, Just now



**App**

**Version**  
Select an app to target by version

**Build number**  
Select an app to target by build number

**First open**  
Select an app to target by first open time.

**Platform**

**Languages**  
The inferred language for the user's device

**Country/Region**  
The inferred region or country for the user's device

**User audience(s)**   
Audience to which the user belongs

**User property**   
Match against the value of an Analytics User Property

**User in random percentage**  
A fractional percentage randomly and persistently assigned to each user on a per-project basis.

**Date/time**  
The date/time of the request

**Installation ID**  
The Firebase installation ID of the instance

**User exists**  
Targets all users within this project

# Possible conditions params



Lesson #2

# Safer decisions with Crashlytics & feature flags



12:58



X

## Why go Premium?

- Unlock unlimited matches
- Video call & super like features
- Advanced filtering options
- Ad-free experience
- Priority customer support



### Choose Your Plan:

Weekly \$2.99

Monthly \$9.99

Yearly \$89.99

First 7 days for free!

Go Premium





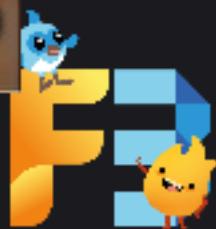
@mkobuolys | @dariandroid

**YOU GET A FEATURE,  
AND YOU GET A FEATURE**



**EVERYONE GETS A NEW FEATURE**

imgflip.com



Purrfect Match



Spooky



@mkobuolys | @dariandroid

## Ratings and Reviews

[See All](#)

# 1.5

out of 5

69 Ratings



Mr. Not Hater

IS THIS A JOKE?

Your app is bad, you should feel bad



Tinder owner

Video call crash

Video call feature crashes the app



Purrfect Stalker

Video call sucks

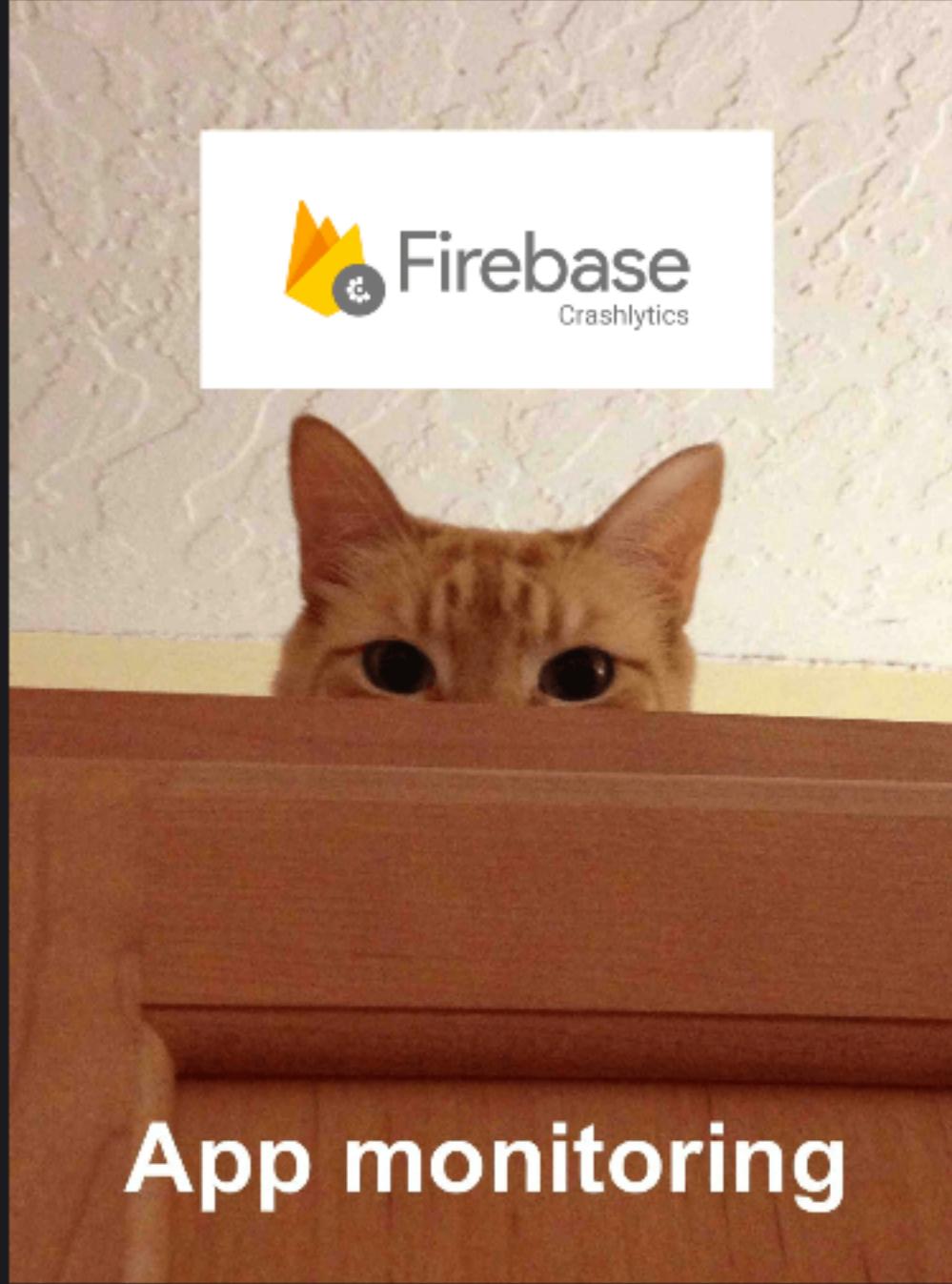
Paid for premium, got an app crash

# 84%

of users abandon an application after  
seeing two crashes

Source: <https://www.bugsnag.com/research/are-your-apps-healthy>





```
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);

  // Pass all uncaught "fatal" errors from the framework to Crashlytics
  FlutterError.onError = FirebaseCrashlytics.instance.recordFlutterFatalError;

  // Pass all uncaught asynchronous errors that aren't handled by the Flutter framework to Crashlytics
  PlatformDispatcher.instance.onError = (error, stack) {
    FirebaseCrashlytics.instance.recordError(error, stack, fatal: true);
    return true;
  };

  runApp(const App());
}
```



# Crashlytics

purfect\_match (android) ▾

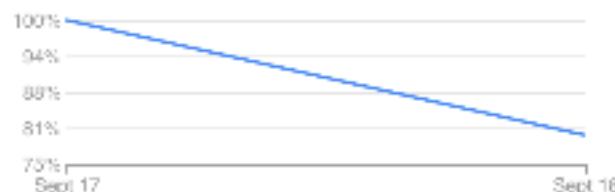
Custom  
17 Sept – 19 Sept

Filter

## Crash-free statistics

Crash-free users ⓘ

83.33%



## Trends

Crashes ⓘ

1

Users

1

Non-fatal ⓘ

7

Users

1

ANRs ⓘ

—

Users

—



## Issues

Search by user ID

Filter Issues

Issue state = "Open" X



Search issue title, subtitle or keys



Issues

Versions

Events ↓

Users

● Non-fatal |  package:purfect\_match/video\_call/widgets/video\_call\_page.dart:36

\_VideoCallPageState.initState

io.flutter.plugins.firebaseio.crashlytics.FlutterError - CameraException(Camera not available, Camera is not available on this device.)

1.0.0 – 1.0.0

7

1





## Crashlytics

Alert

In Firebase

Email

A new fatal issue is reported

 ▾

A new non-fatal issue is reported

 ▾

A new ANR issue is reported

 ▾

Trending issues are detected

 ▾

A regression is reported

 ▾  ▾

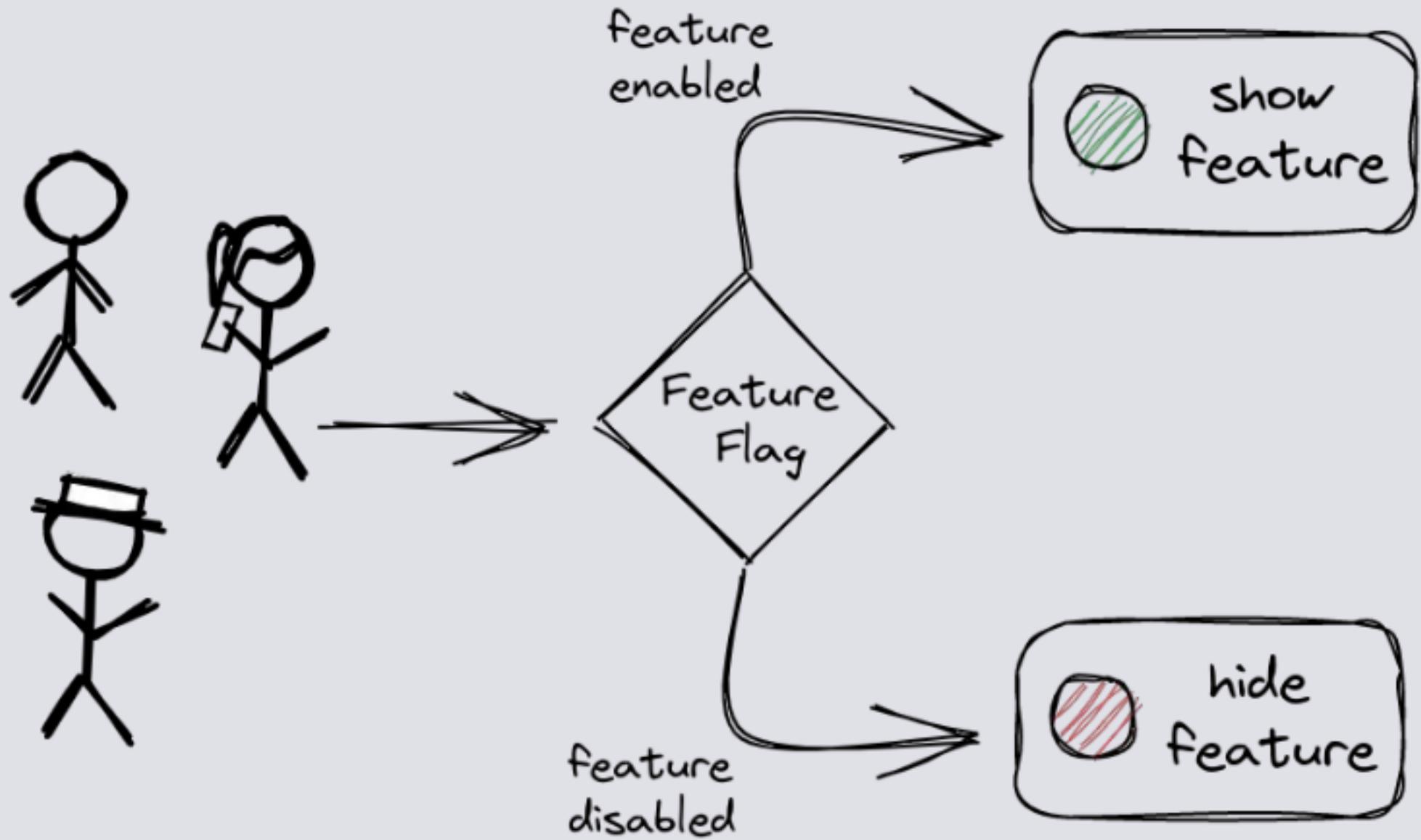
A velocity alert is reported

 ▾  ▾

A missing Apple dSYM has been detected

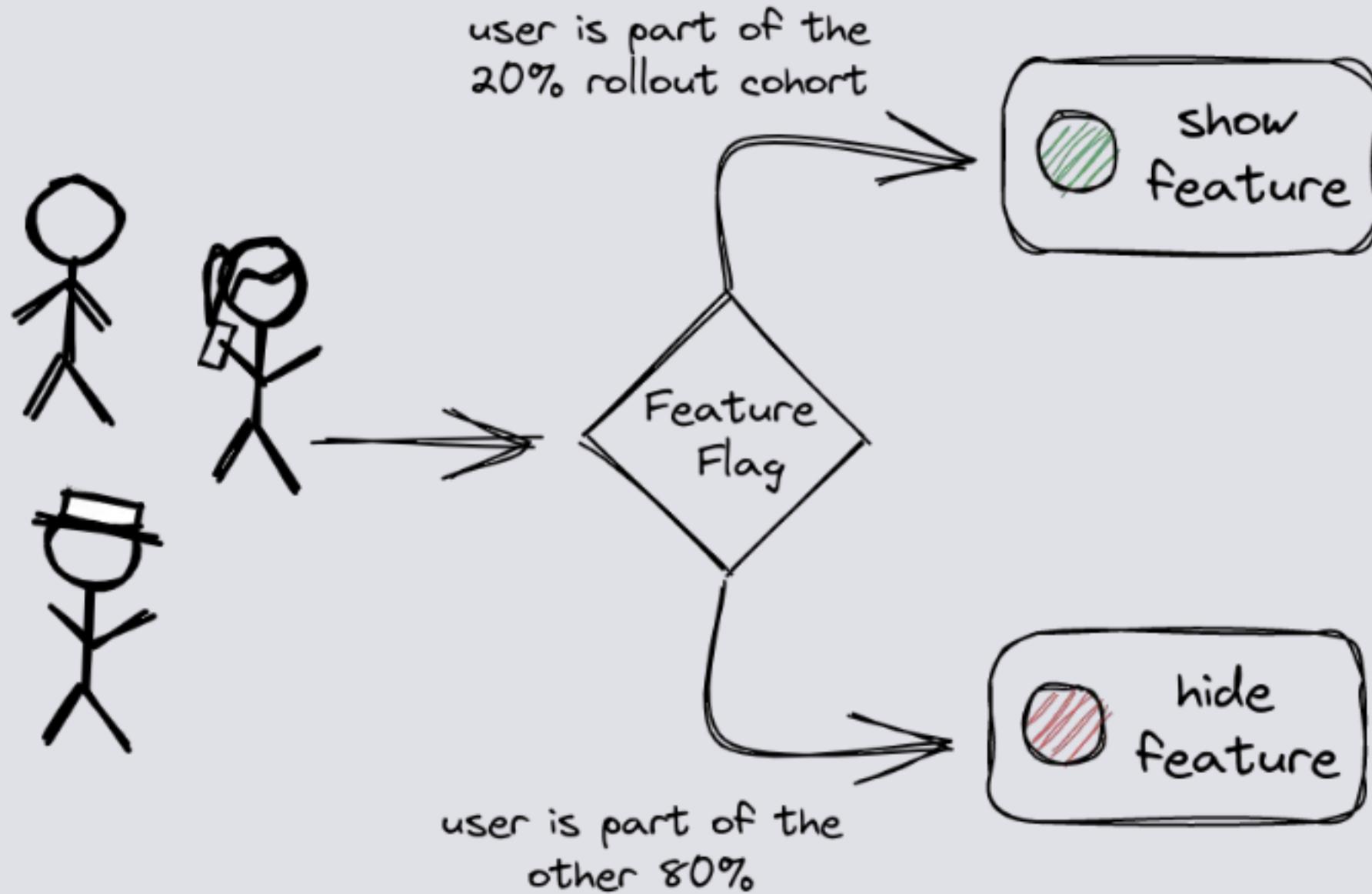
 ▾





Source: <https://reflectoring.io/java-feature-flags/>





Source: <https://reflectoring.io/java-feature-flags/>



## Define a new condition

Use conditions to provide different parameter values if a condition is met. Any changes made to this condition will apply to **all parameters** using this condition.

Name

Colour

20% of users

Cyan



Applies if...

User in random percentage



0



20

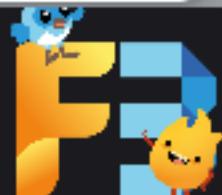
Edit seed

and

Fewer than 100 users match this condition

Cancel

Create condition



## Edit parameter

X

Parameter name (key) ?

Data type

video\_call\_enabled

Boolean

Description

Feature flag for the video call feature.

Condition

20% of users



Value

true



Use in-app default

Default value

false



Use in-app default

+ Add new

Cancel

Save



```
class CardSwiperActions extends StatelessWidget {  
  const CardSwiperActions({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    final videoCallEnabled = FirebaseRemoteConfig.instance.getBool('video_call_enabled');  
  
    return Row(  
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
      children: [  
        const RejectButton(),  
        if (videoCallEnabled) const VideoCallButton(),  
        const ApproveButton(),  
      ],  
    );  
  }  
}
```



## Edit parameter

Parameter name (key) [?](#)

video\_call\_enabled

Data type

Boolean

Description

Feature flag for the video call feature.

Condition

20% of users



Value

true

Use in-app default

Default value

false

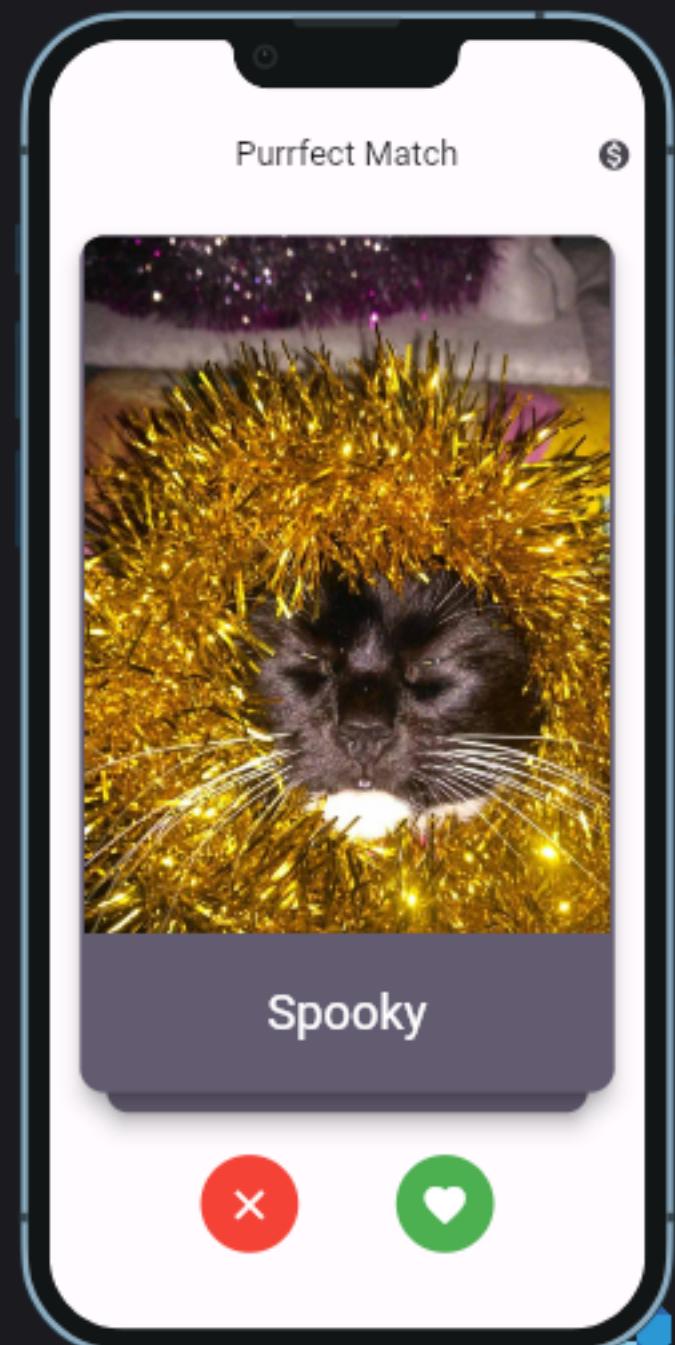
Use in-app default

[+ Add new](#)

Cancel

Save

Video call enabled:



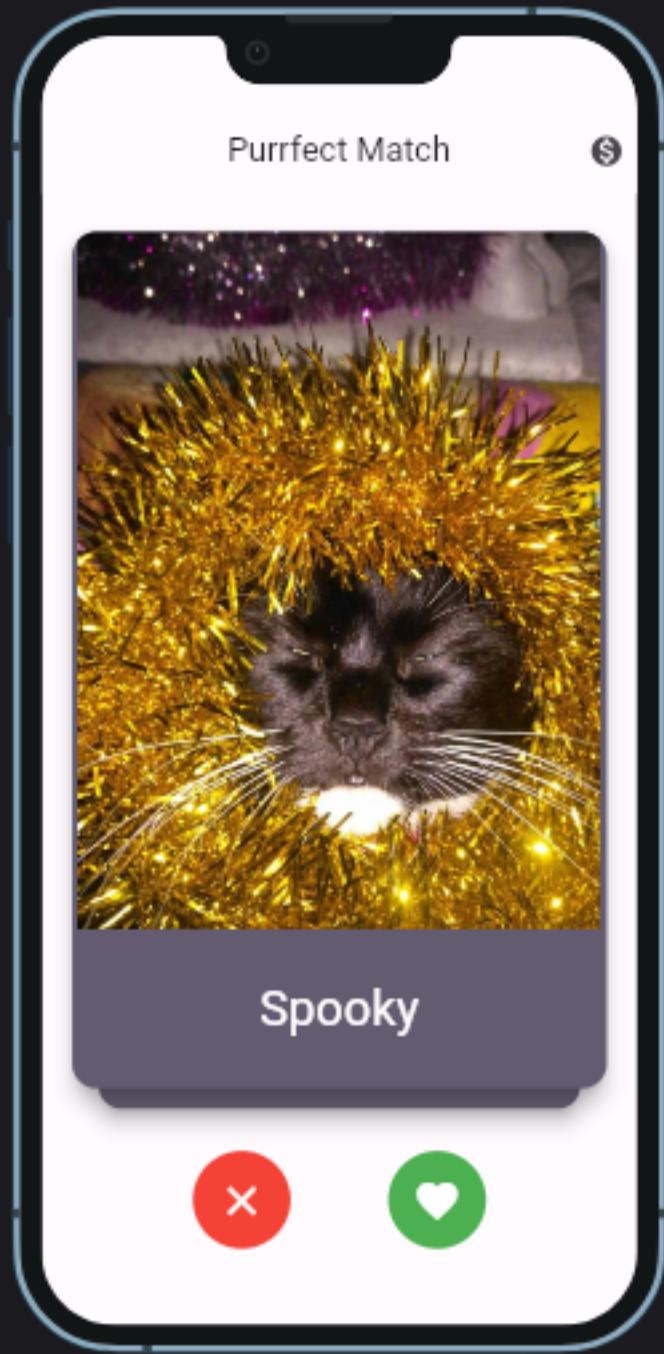
# Turning Users into Subscribers



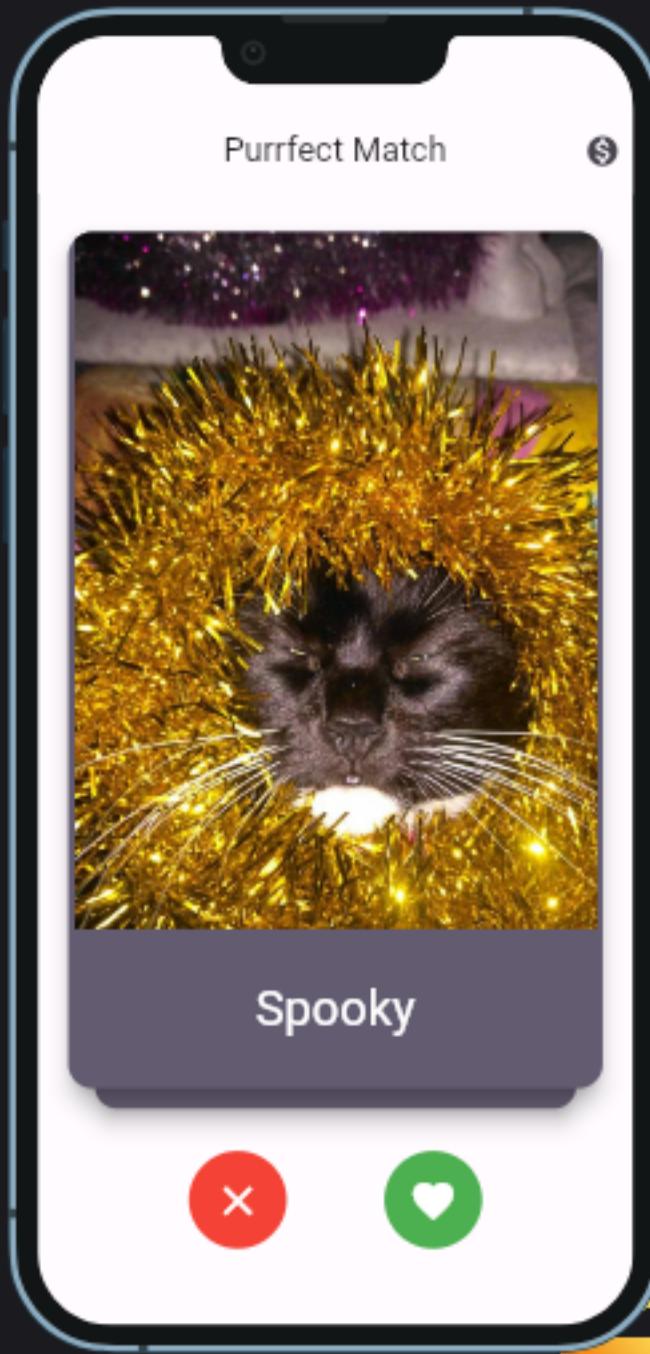
Lesson #3

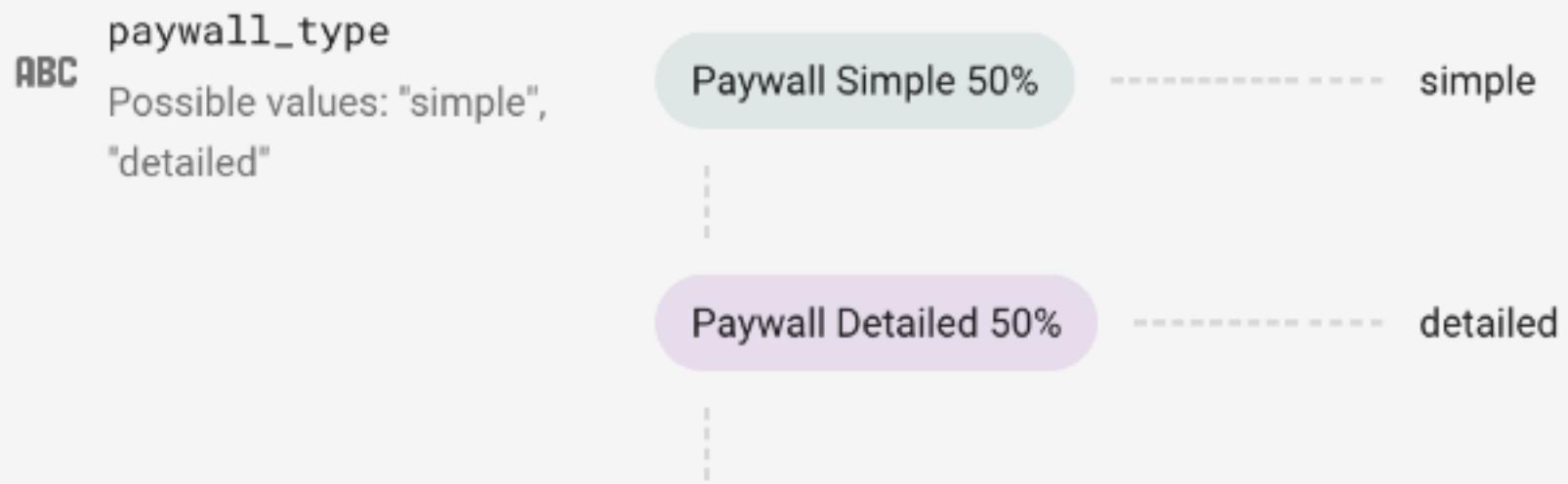
A/B testing – even smarter  
decisions, even faster





VS





# Option 1: Distribute via Conditions & analyze with Analytics





## Firebase A/B Testing

Option 2: A/B testing



```
final paywallType = FirebaseRemoteConfig.instance.getString('paywall_type');
```

# Step 1: Use as a regular FRC param



```
void logShowPaywall() {  
    analytics.logEvent(name: 'show_screen_paywall');  
}  
  
void logHidePaywall() {  
    analytics.logEvent(name: 'hide_screen_paywall');  
}  
  
void logActivateTrial() {  
    analytics.logEvent(name: 'activate_trial');  
}
```

## Step 2: Log analytics events



The screenshot shows the Firebase console interface. On the left, there's a sidebar with various project management and developer tools: Project Overview, Project shortcuts, Realtime Database, Remote Config, Product categories, Build, Release & Monitor, Analytics, Engage, A/B Testing (which is highlighted with a pink oval), and Messaging. The main content area is titled "A/B Testing" and includes a sub-header "Beta" and a sub-sub-header "Run experiments to make key engagement flows more effective". There's a prominent "Create experiment" button. To the right of the text is a cartoon illustration of a scientist wearing a lab coat and safety goggles, holding a flask with red liquid. The background of the main area is a light blue gradient.

# Step 3: Firebase Console -> Engage

## -> A/B testing



A screenshot of the Figma A/B Testing interface. At the top left is the project name "purrfect-match". On the right are three small icons: a question mark, a message bubble, and a bell. Below the project name is the title "A/B Testing" followed by a "Beta" badge. At the bottom right of the interface is a blue button labeled "Create experiment".

# Step 4: Create Experiment



Which service do you want to experiment with?

X

 Notifications

Experiment with sending messages to engage the right users at the right moment

 Remote Config

Experiment with app behavior using server-side configuration parameters

 In-App Messaging

Experiment with sending different In-App Messages to drive maximum engagement

# Step 5: Select Firebase Remote Config



1 Basics

This experiment will run for up to 90 days

Experiment name: Paywall type (Android)

Experiment description (optional):

Next

2 Targeting

3 Goals

Additional metrics to track: Revenue (4-7 days) and Gross Revenue

4 Variants

Cancel Review

# Step 6: Set experiment name & description



Basics  
name: Paywall type (Android)

Targeting

Mixed users

App

pmflect\_match (android) (com.ecomplex.pmflect\_match)  
ios+ pmflect\_match (ios) (com.ecomplex.pmflect\_match)

100% of eligible users will be exposed to this experiment: ~10

Exposure 100 %

Action on event (optional): Select Activation Event

Next

Goals

Add goal metrics to track (start on 4-7 days) and Create first user

Variants

# Step 7: Select app & user percentage



Basics  
name: Paywall type (Android)

2 Targeting

Target users

App: purrfect\_match (android) (com.example.purrfect\_match) and  
iOS: purrfect\_match (ios) (com.example.purrfectMatch)

100% of eligible users will be exposed to this experiment: ~10 ⓘ Exposure: 100 %

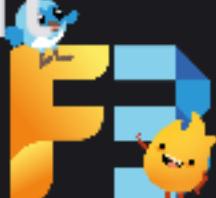
Activation event (optional) ⓘ  
Select Activation Event ▾

Next

3 Goals  
Additional metrics to track: Retention (4-7 days) and Crash-free users

4 Variants

# Step 8: Select activation event



The screenshot shows the Figma experiment editor interface. On the left, there's a sidebar with a search bar at the top labeled "Search for or add new event". Below it is a list of "Other analytics events" with one item, "show\_screen\_paywall", circled in red. To the right of the sidebar, there's a main panel titled "perfect match (android)" with a "this experiment" dropdown set to "10" and an "Exposure" slider set to 100%. At the bottom of the sidebar, there are two buttons: "Next" and "Goals".

Goals  
Additional metrics: total sessions (4+7 days) and push frequency

1 Variants

## Step 9: Select

"show\_screen\_paywall"



**Basics**  
name: Paywall type (Android)

**Targeting**  
Targeting 1 criteria  
100% of target users  
Activation event: show\_screen\_paywall

**3 Goals**

Primary metric to track (determines leader) ⓘ  
 activate\_trial ▾

Additional metrics to track ⓘ  
Add metric + Retention (4–7 days) ✕ Crash-free users ✕

**Next**

**4 Variants**

Cancel Review

# Step 10: Set goal as "activate\_trial"



Basics  
name: Paywall type (Android)

Targeting  
Targeting 1 criteria  
100% of target users  
Activation event: show\_screen\_paywall

3 Goals

Primary metric to track (determines leader) ⓘ  
activate\_trial

Additional metrics to track ⓘ  
Add metric + Retention (4–7 days) ✕ Crash-free users ✕

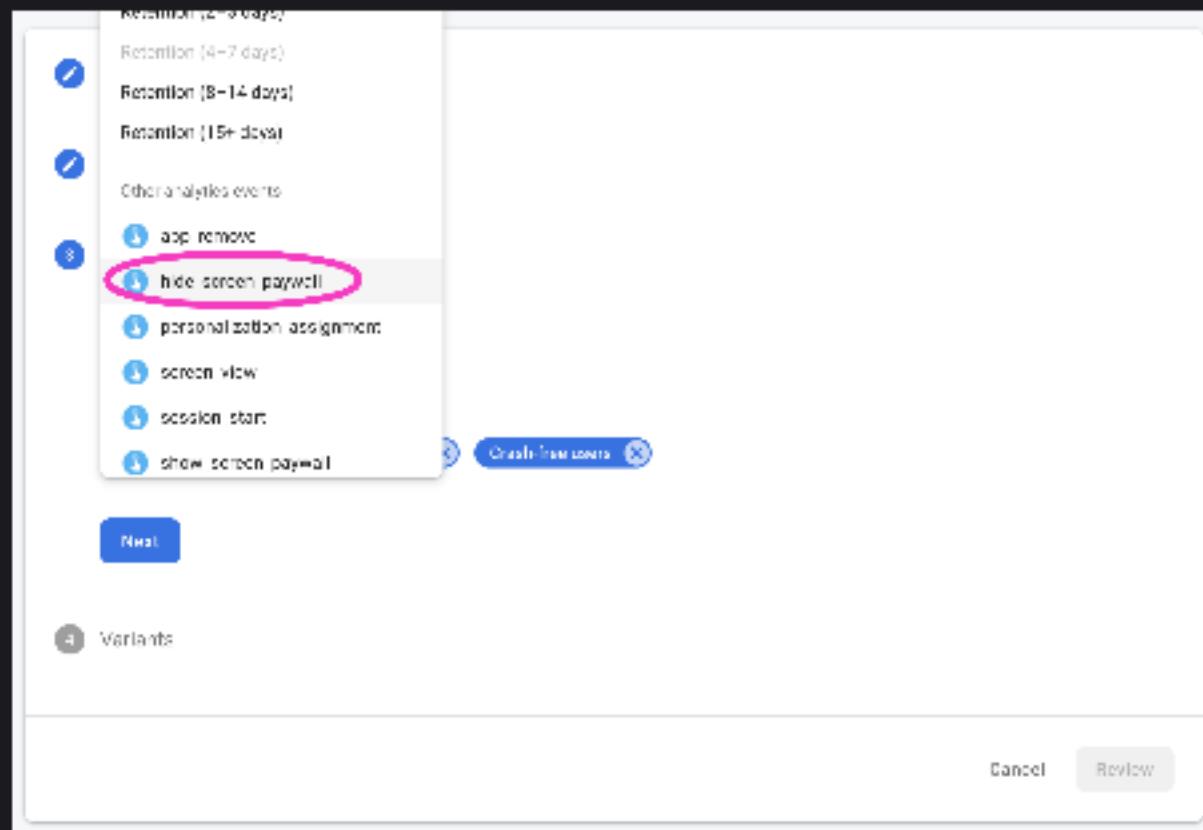
Next

4 Variants

Cancel Review

# Step 11: Add another metric





## Step 12: Select

"hide\_screen\_paywall"



4 Variants

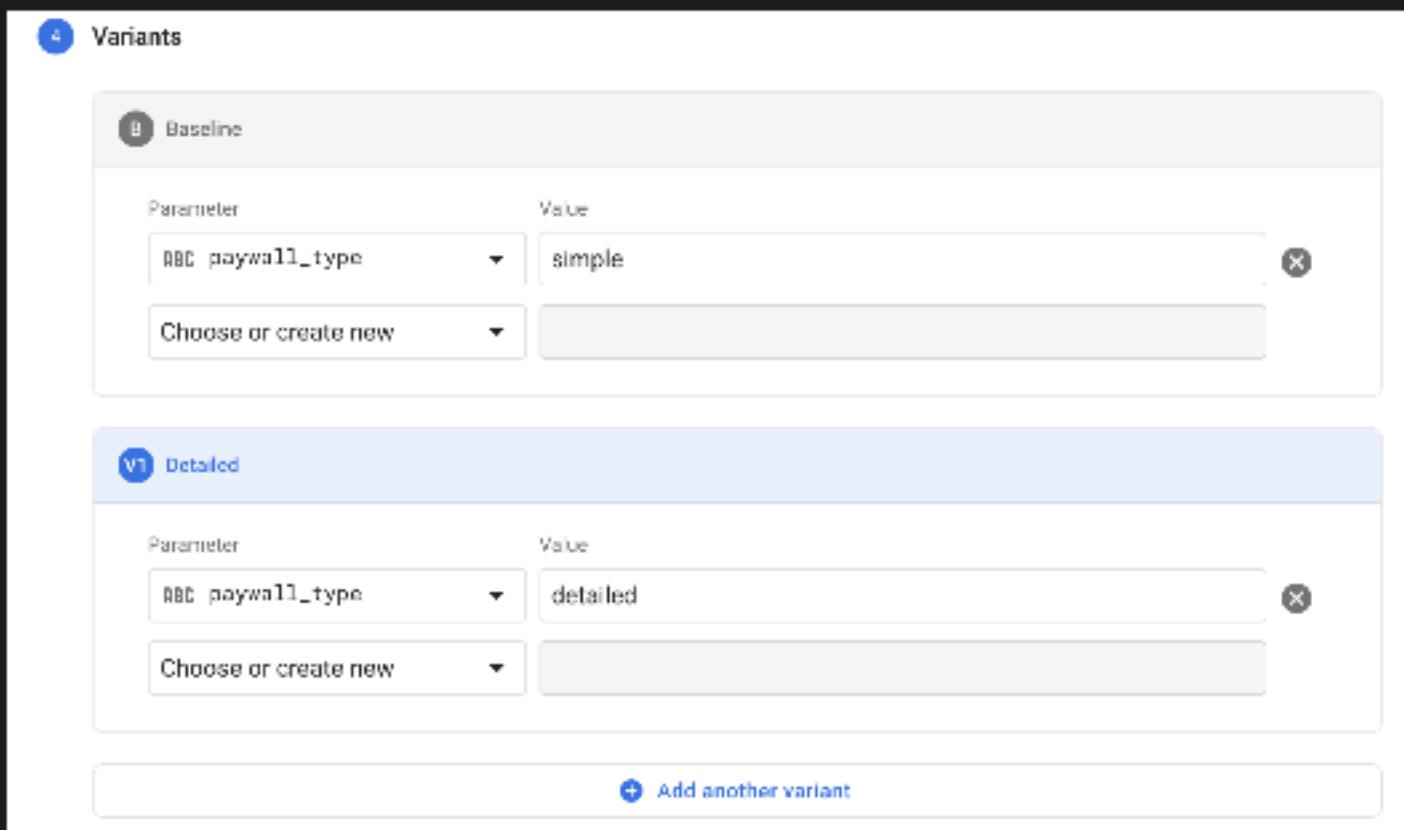
B Baseline

Parameter	Value
RBC_paywall_type	simple
Choose or create new	

V1 Detailed

Parameter	Value
RBC_paywall_type	detailed
Choose or create new	

+ Add another variant



# Step 13: Set values to experiment with: "detailed" and "simple"



Paywall type (Android) Beta

Experiment overview

Draft

Your experiment is ready to start

[Edit](#)[Start experiment](#)

Total users

0

 Remote Config Created just now show\_screen\_paywall activate\_tier 100% of users matching 1 criteria 2 variants 0 test devices

⋮

Start experiment?

X

- Once you start, experiment parameters are available to your apps and users

[Cancel](#)[Start](#)

Experiment results

How to read this chart

[show\\_screen\\_paywall](#) 

7 days

30 days

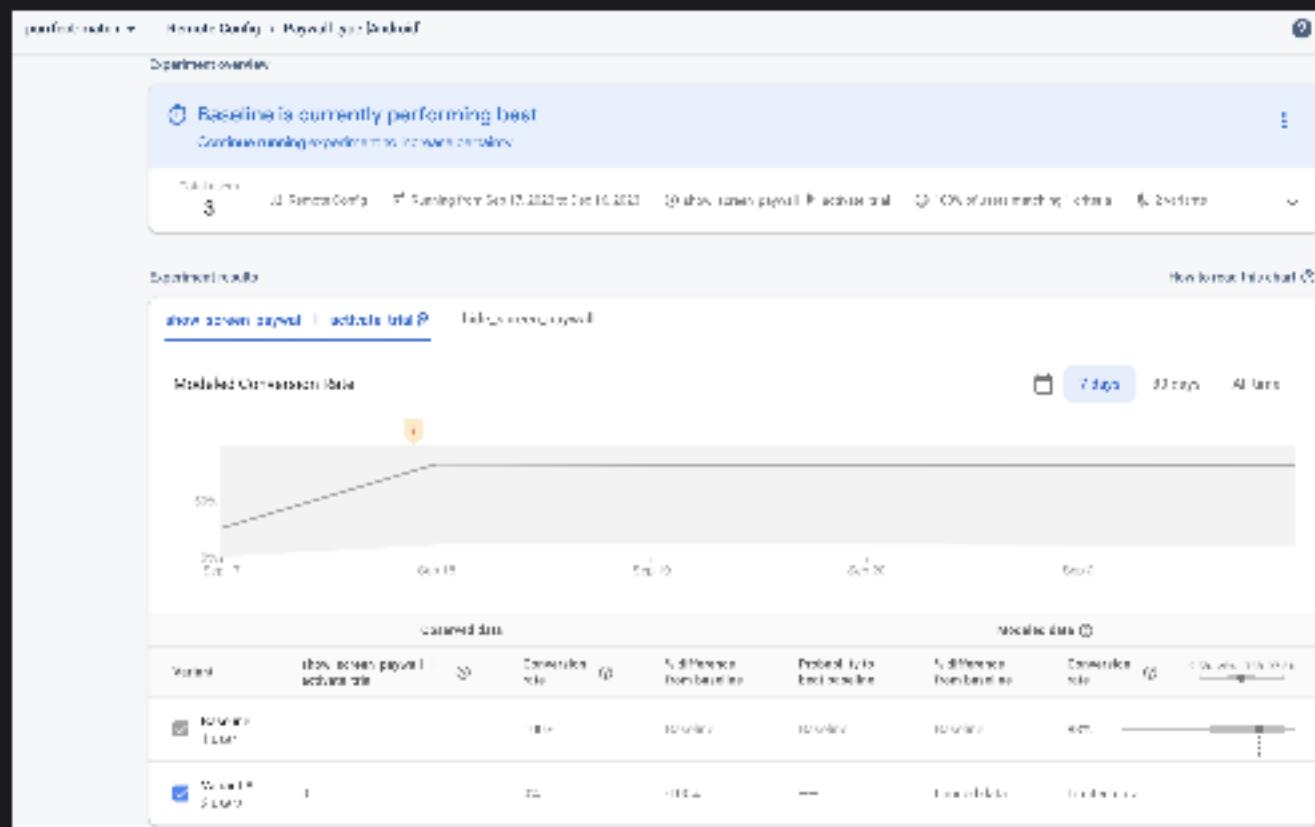
All time

Modeled Conversion Rate

No data

# Step 14: Start the experiment!





# Step 15: Some time later... analyse the results

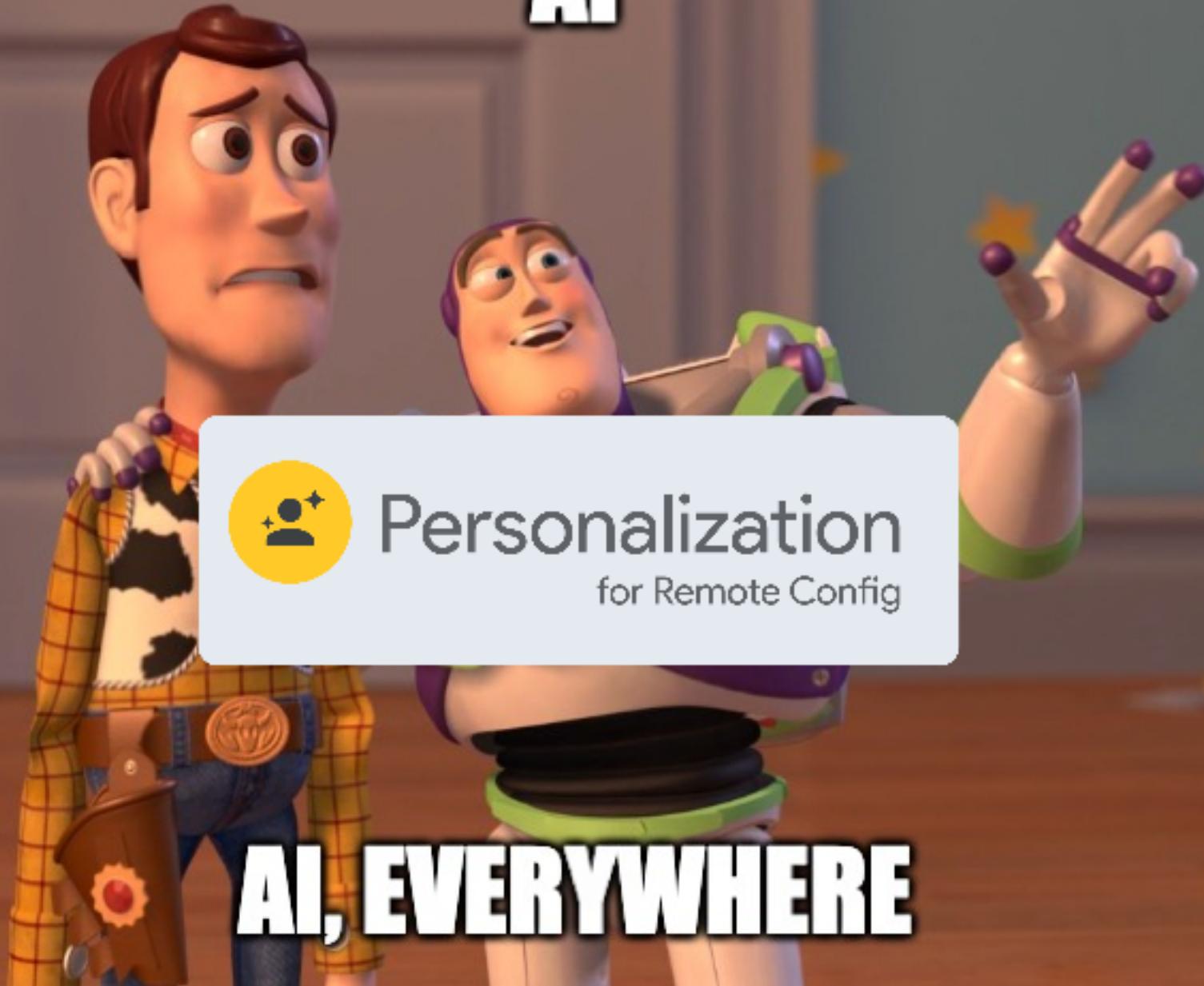


Smarter than the smartest, faster than the fastest

# Remote Config Personalization



AI

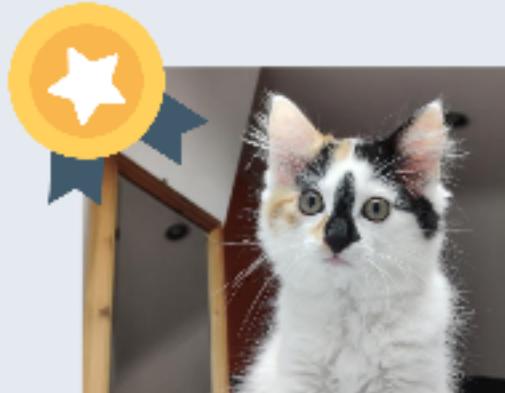


imgflip.com





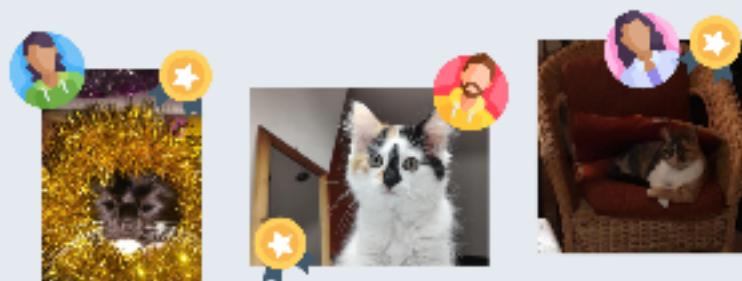
=



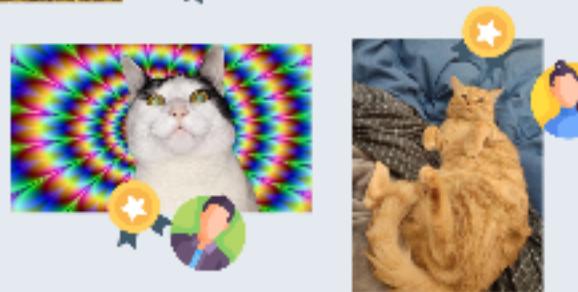
A/B Testing

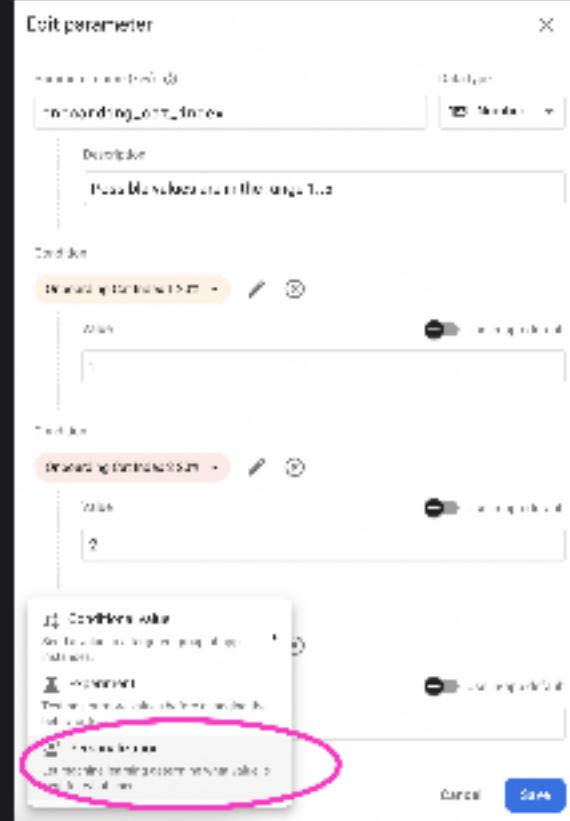


=



Personalization  
for Remote Config





1. Edit param in FRC -> Add new ->

## Personalization



← Create personalization

Personalization uses ML with Remote Config to deliver custom experiences to each of your end users based on their individual needs and device. [Learn more](#)

1 Enter alternative values

Alternative values are a collection of member values that may be selected and returned to an end user by the personalization algorithm.

1 Alternative 1

Value  
1

2 Alternative 2

Value  
2

+ Add alternative value (5 max)

Next

2 Choose objectives

Cancel Save

## 2. Select values to personalize



← Create personalization

Personalization uses ML with Remote Config to deliver custom experiences to each of your users, based on their individual needs and desires. [Learn more](#)

5 steps

2 Choose objective 

The algorithm finds which item or value is best to serve to each user to maximize your selected objective. E.g. The algorithm finds which ad frequency is best for user xyz.

Objective:  start swiping

Optimizer:  

Additional metrics: Additional metrics to track (2 max). These have no effect on the algorithm.

Add metric: 



3 Target condition

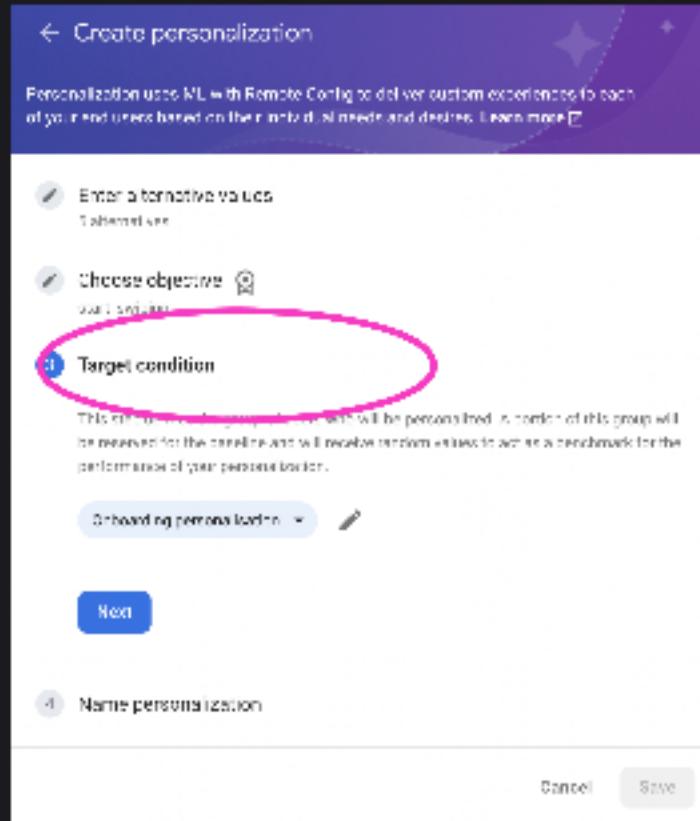
4 Name personalization

Cancel 



### 3. Select an objective to track





## 4. Select target user group with FRC conditions



← Create personalization

Personalization uses ML with Remote Config to deliver custom experiences to each of your end users based on their individual needs and desires. [Learn more](#)

- Enter alternative values  
5 alternatives
- Choose objective   
start\_swiping
- Target condition  
Onboarding\_personalisation
- Name personalization

Name

Onboarding personalisation

Description (optional)

Which cat performs best?

Cancel Save



## 5. Set name and description





See applied personalization in the  
console



The screenshot shows a video player interface with a dark theme. At the top left is the '#GoogleIO' logo. On the left side, there's a vertical sidebar with navigation links: 'Project Overview', 'EdenUI', 'Data', 'Release & Monitor', 'Analytics', 'Engage', 'Billing', and 'Help'. The main content area displays a 'Personalization' dashboard from the Firebase console. The dashboard has a header stating 'Your Personalization is outperforming the baseline by 6.7%' with a date range from 'Nov 30, 2021' to 'Dec 10, 2021'. It includes a summary card with metrics: 'Users in last 24 hours' (1204 Personalized vs 1065 Baseline), 'Ad click' (1 engagement), 'User engagement time' (1 hour), and 'Retention (30 day)' (1.6 app). Below this is a 'Results summary' section with a bar chart titled 'Engagement time in minutes' comparing 'Personalization' (blue bar) and 'Baseline (scaled up)' (grey bar). The chart shows a total of 15 + 24.6% engagement time. The video player controls at the bottom include play/pause, volume, and a progress bar showing '12:22 / 13:13'. To the right of the dashboard are two video thumbnails: one featuring a woman in a black sweater and another in a blue plaid shirt, both set against a wooden wall background.

Create better user experiences with Remote Config for advanced app personalization

Source: <https://www.youtube.com/watch?v=nRQwJV1iMKs>



Select app

ⓘ Your Personalization is underperforming the baseline by 25%

7  
Users in last  
28 days

5 Personalized  
2 Baseline

Started: Sep 24, 2023  
1 targeting criteria

start\_swiping  
No additional metrics

5 values  
Assignment window: 24 hours

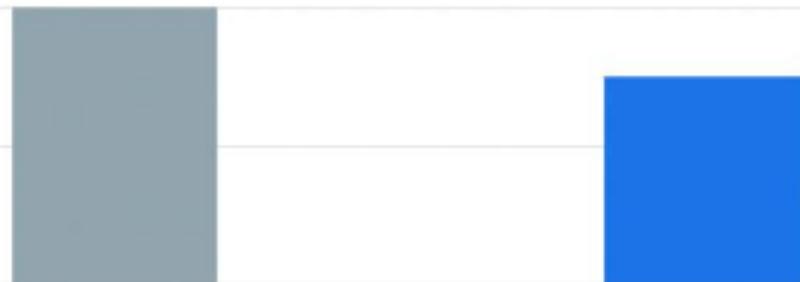
Results summary

Last 30 days  
Aug 26 – Sep 25

Count of start\_swiping

Daily Total

Total lift  
**-1 -25%**



Personalization  Baseline (scaled up) ?

Show baseline  
performance details ?



# Summary

- Smarter - Analytics and Firebase Remote Config
- Safer - Crashlytics and feature flags
- Faster - A/B testing and personalization



# Thank you! 😺



Mangirdas Kazlauskas, Daria Orlova  
Average cat enjoyers  
@mkobuolys | @dariadroid

