

# Make smarter decisions faster with Firebase Remote Config

Mangirdas Kazlauskas, Daria Orlova

Average cat enjoyers

@mkobuolys | @dariadroid

**Daria** @dariadroid - Aug 14, 2022  
 We spent this weekend in a lovely city I hope found you if you've ever in Latvia 🇱🇻 and there are so many friends who there I've gotten at least 5 of them! And my version of "Daria" by yourself with one photo" is here 📷



**Daria** @dariadroid - Mar 10  
 Replying to @vibe: [@vibe](#)  
 I wish! And I putted them all of course 🥰🥰



**Daria** @dariadroid - Mar 10  
 Finally! I tried to go to cat cafes in several countries, but they were always full! Finally got into one 🥰 Kawaii is the best! 🥰🥰



**Daria** @dariadroid - Jan 5  
 All you need to know about my love for cats 🥰 a photo of that you could get with support and I just couldn't work by 🥰 so I'll be in work with Daria 🥰🥰  
 @KishuCatHouse





@mkobuolys | @dariadroid





+



+



=







@mkobuolys | @dariadroid





@mkobuolys | @dariadroid







@mkobuolys | @dariadroid





@mkobuolys | @dariadroid







@mkobuolys | @dariadroid



X-SCITECH &gt;

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

BY RACHAEL RETTNER

MAY 29, 2016 | 4:10 PM | LIVESCIENCE.COM



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

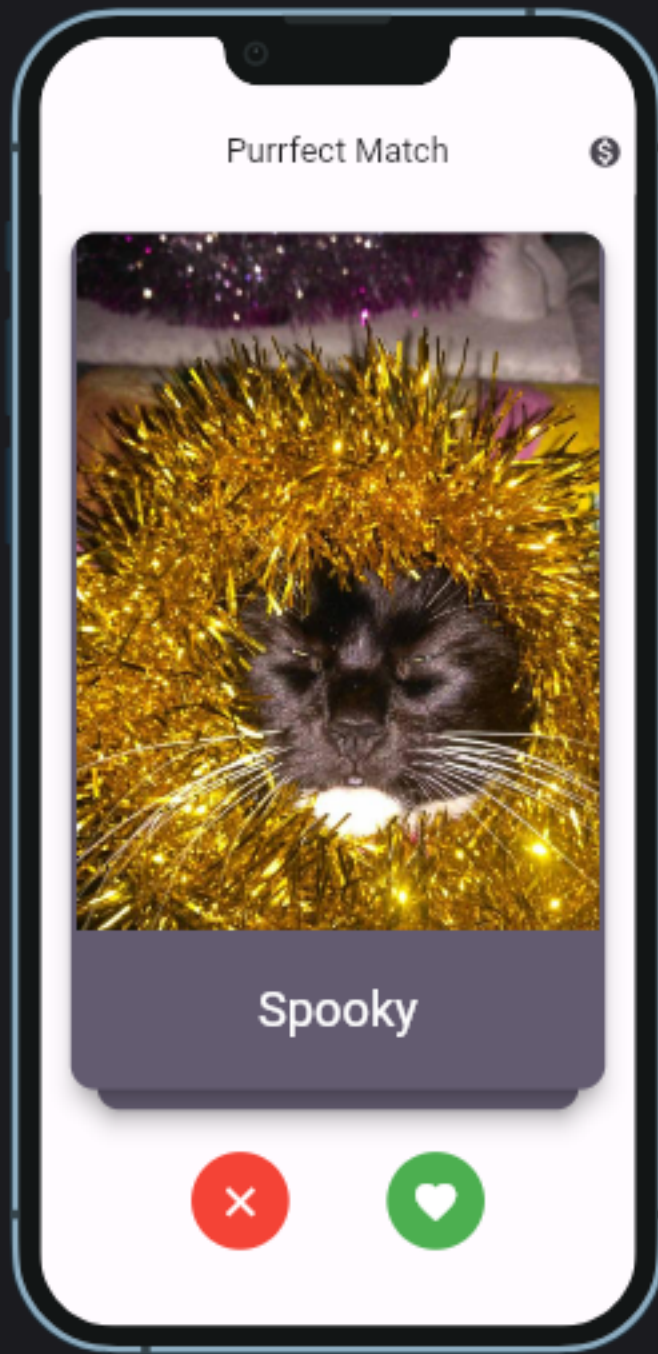
People who said they were dog lovers in the study tended to be more lively -- meaning they were more energetic and outgoing -- and also tended to follow rules closely. Cat lovers, on the other hand, were more introverted, more open-minded and more sensitive than dog lovers. Cat people also tended to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



# Purrfect match

@mkobuolys | @dariadroid







Lesson #1

# Making smart decisions with Analytics

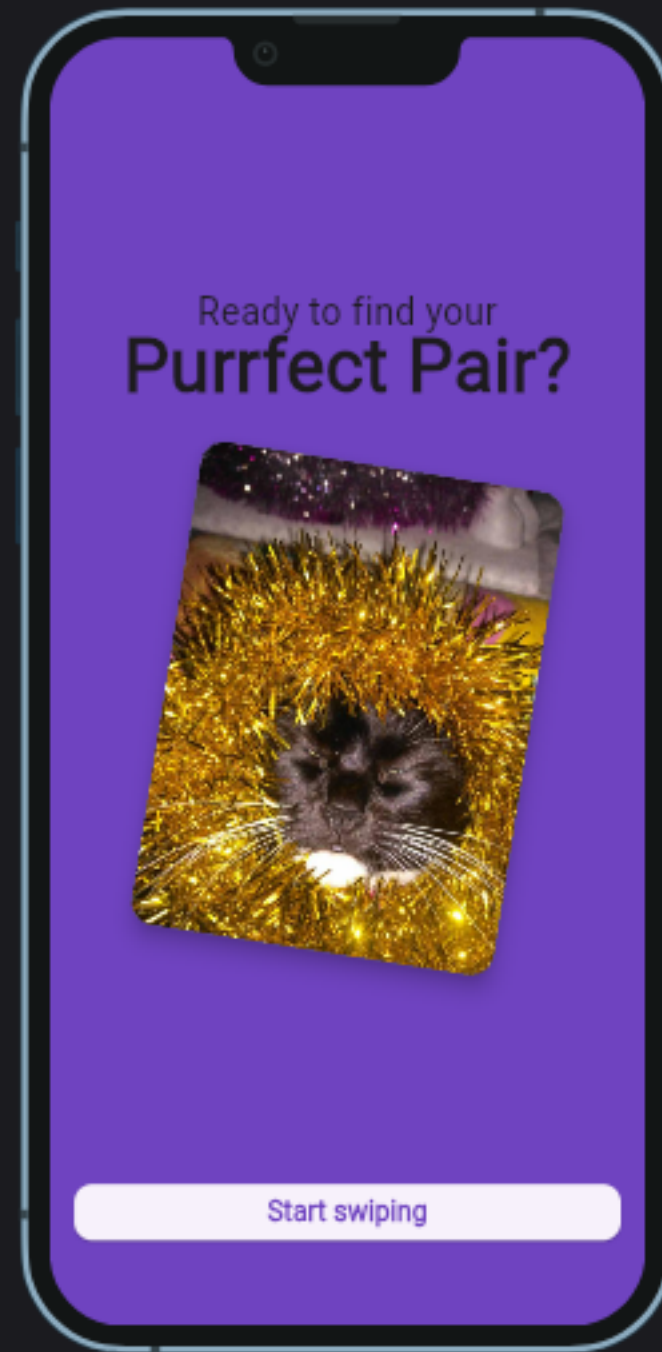


What is the first  
thing that the user  
sees?



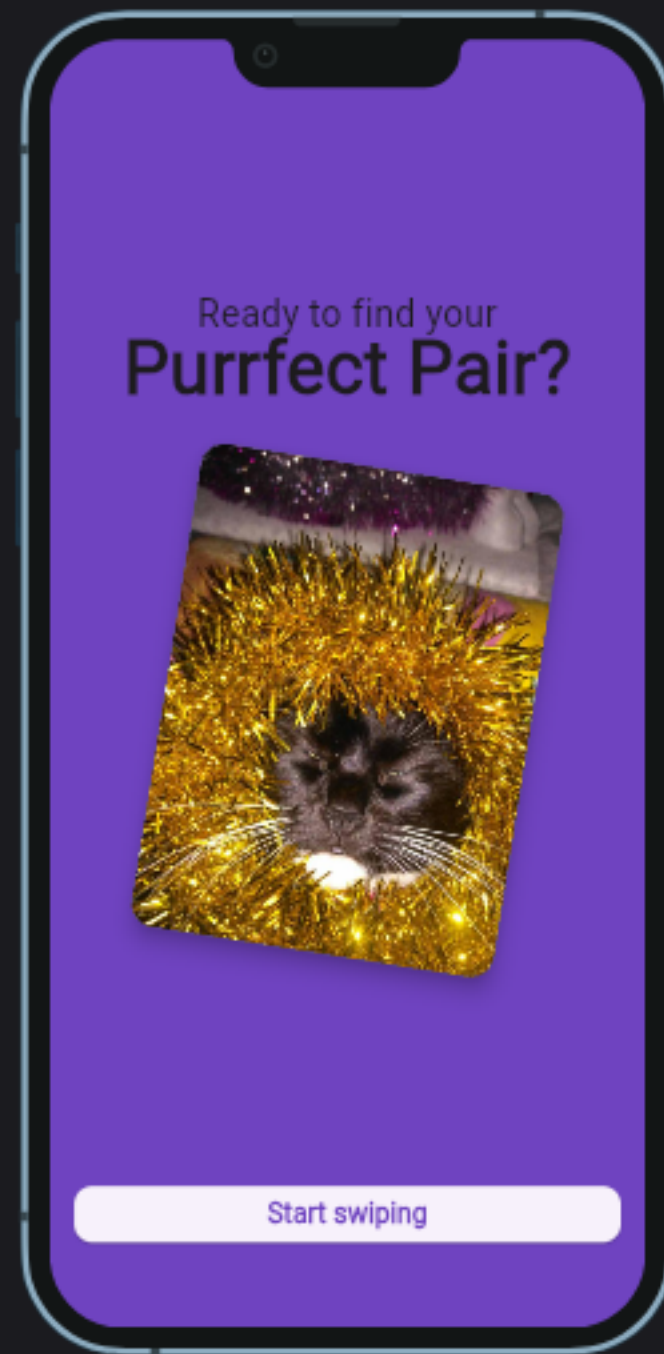
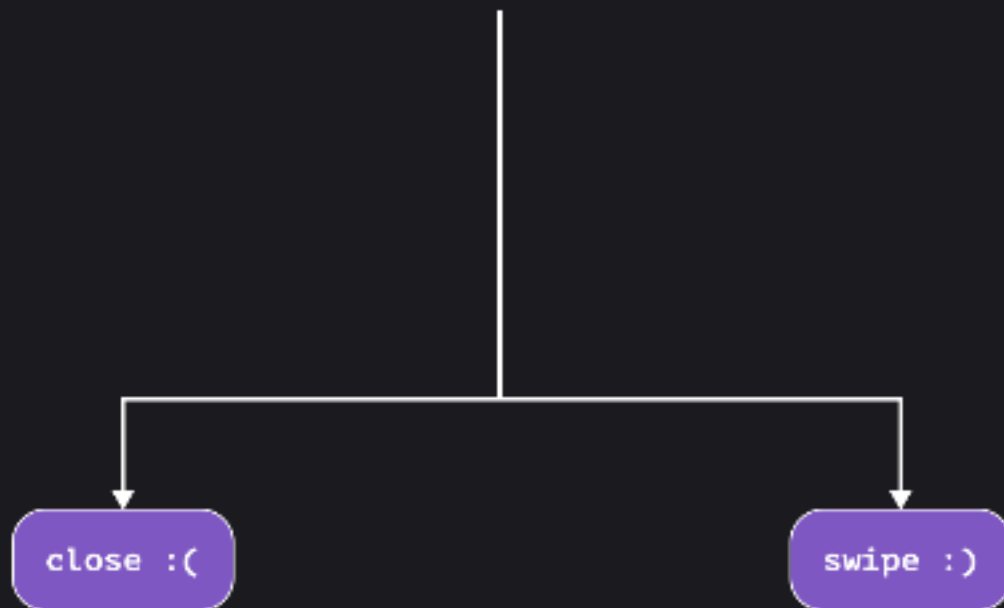
What is the first  
thing that the user  
sees?

# Onboarding



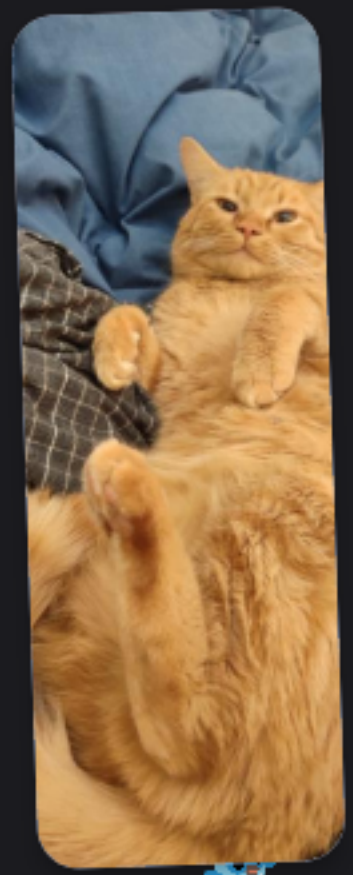
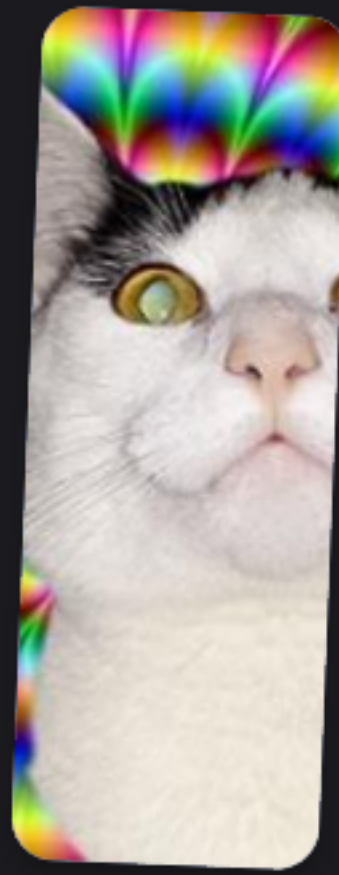
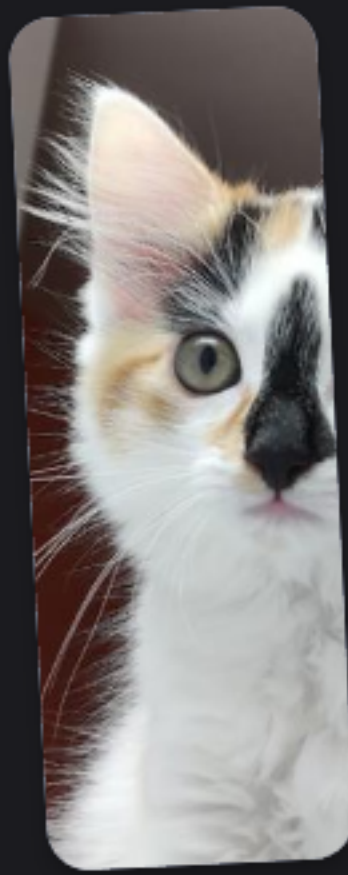
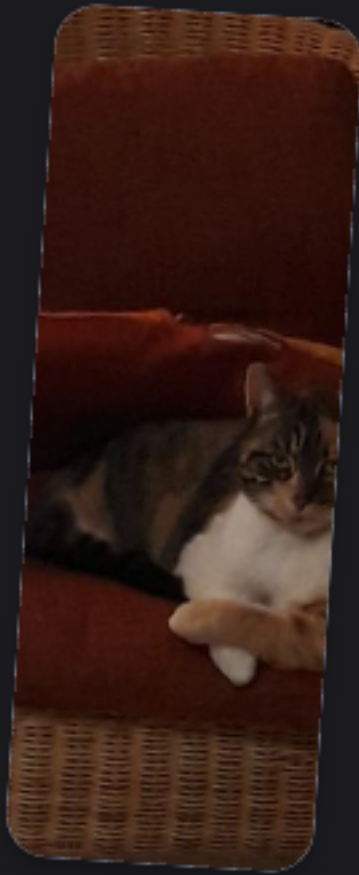
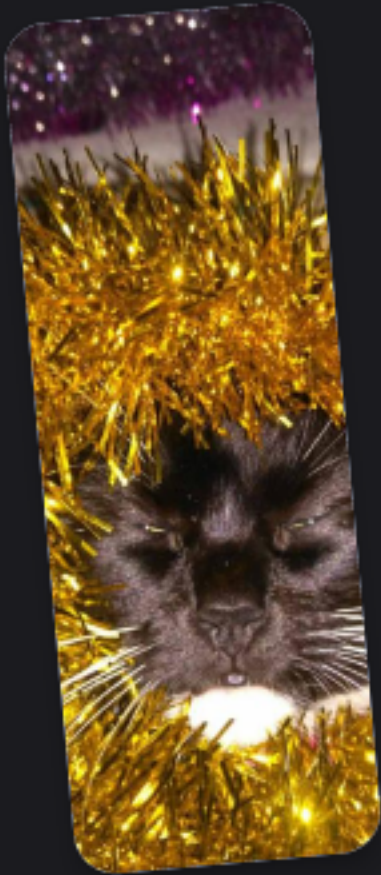
What is the first thing that the user sees?

# Onboarding





# Which cat performs best?



# How can we find out?

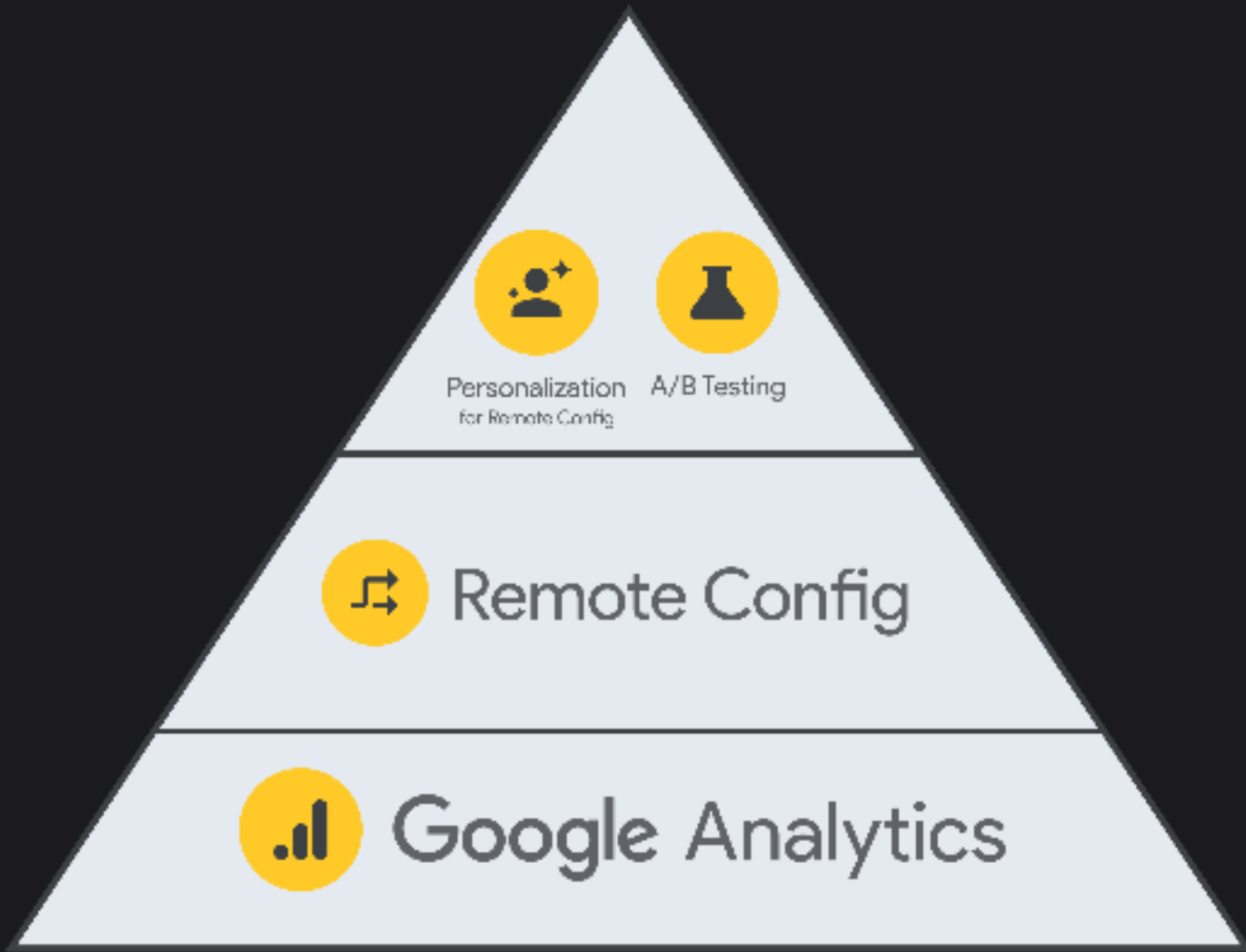


# How can we find out?



# Google Analytics





# Firebase Pyramid





The Foundation

# Google Analytics





```
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>	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 console interface. On the left sidebar, the 'Events' option is highlighted with a red circle. The main content area displays the 'Analytics | Events' page, which includes a search bar, a 'New event' button, and a table of existing events.

Event name	Total	Scamp	View	Scamp	Measurement ID
event_1	10	10	X	10	<input checked="" type="checkbox"/>
event_2	5	5		5	<input type="checkbox"/>
event_3	5	5	X	5	<input type="checkbox"/>
event_4	10	10		10	<input type="checkbox"/>
event_5	1	1		1	<input type="checkbox"/>
event_6	10	10	X	10	<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. Track analytics for one week
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. Track analytics for one week
5. 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 your first parameter

Parameters are key value pairs that can be used as feature flags and much more. Your app will periodically fetch parameters from the server, allowing you to remotely change your app configuration without pushing a new release. [Learn more](#)

Parameter name (key) 

cat\_index

Description

Optional

Data type

- String
- Number
- Boolean
- JSON

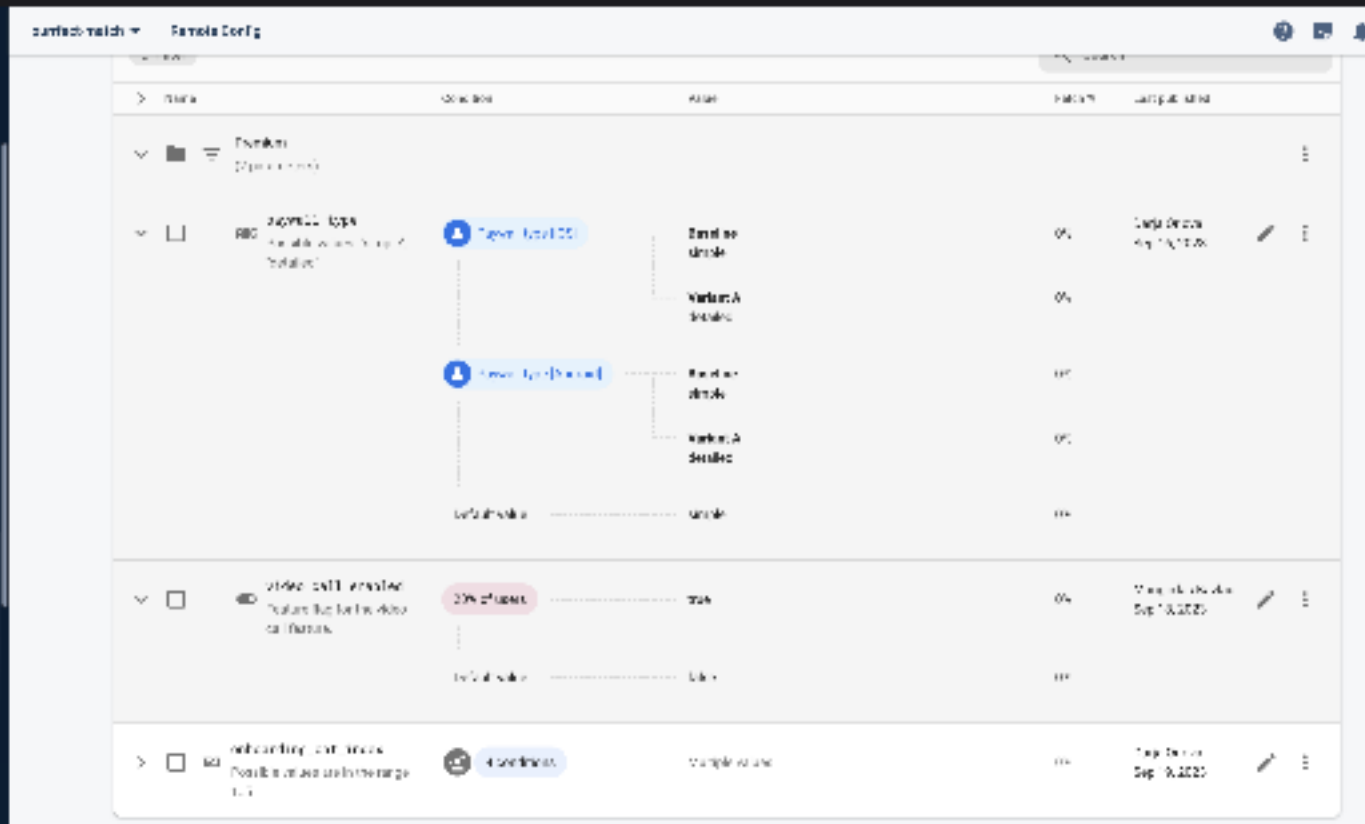
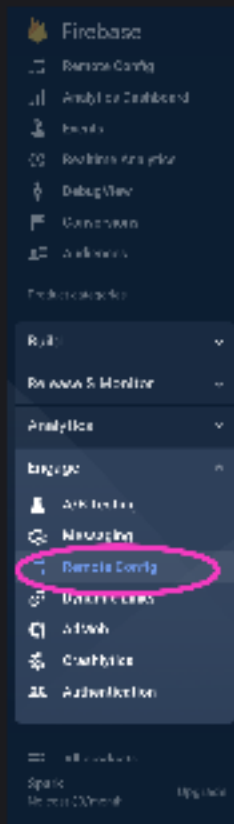
Default value

Use in-app default

[+ Add new](#) [Cancel](#) [Save](#)

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





# 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

Description

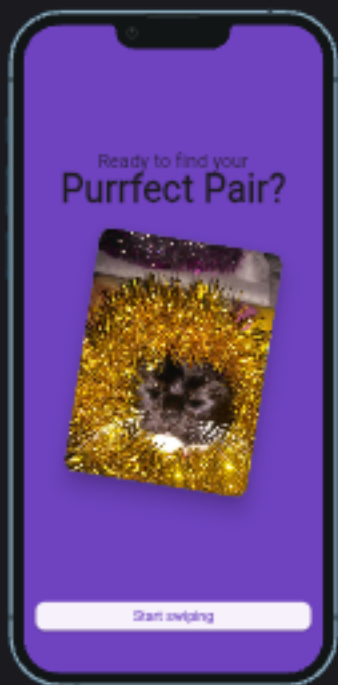
Default value   Use in-app default

[+ Add new](#) [Cancel](#) [Save](#)

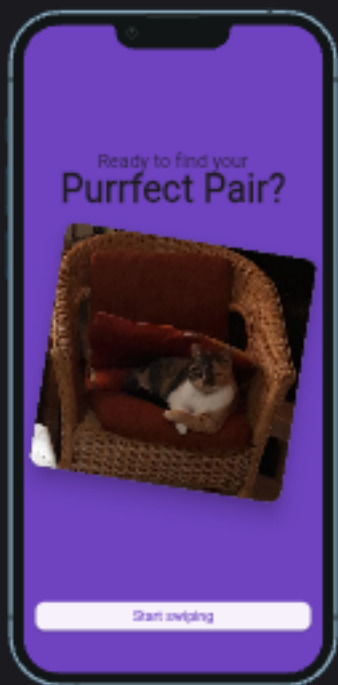
# Step 2: Create param in Firebase console



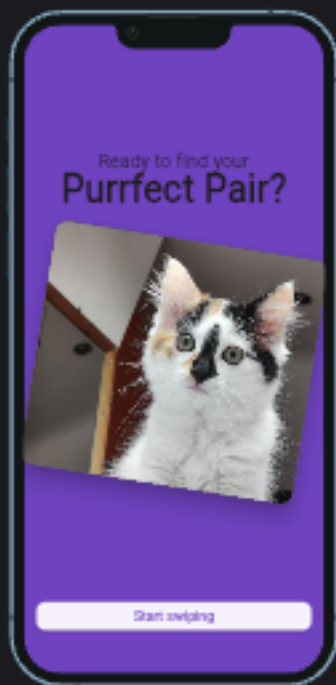
```
final catIndex = FirebaseRemoteConfig.instance.getInt('onboarding_cat_index');
```



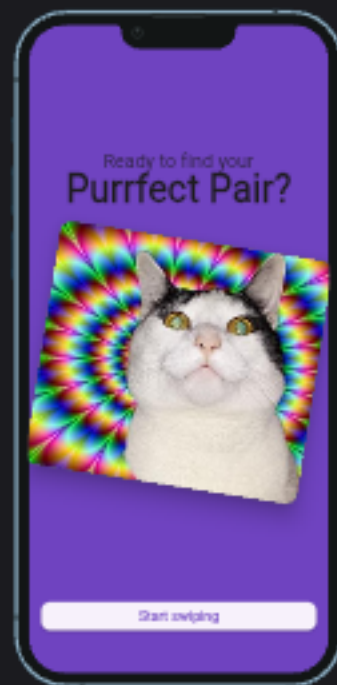
catIndex  
= 1



catIndex  
= 2



catIndex  
= 3

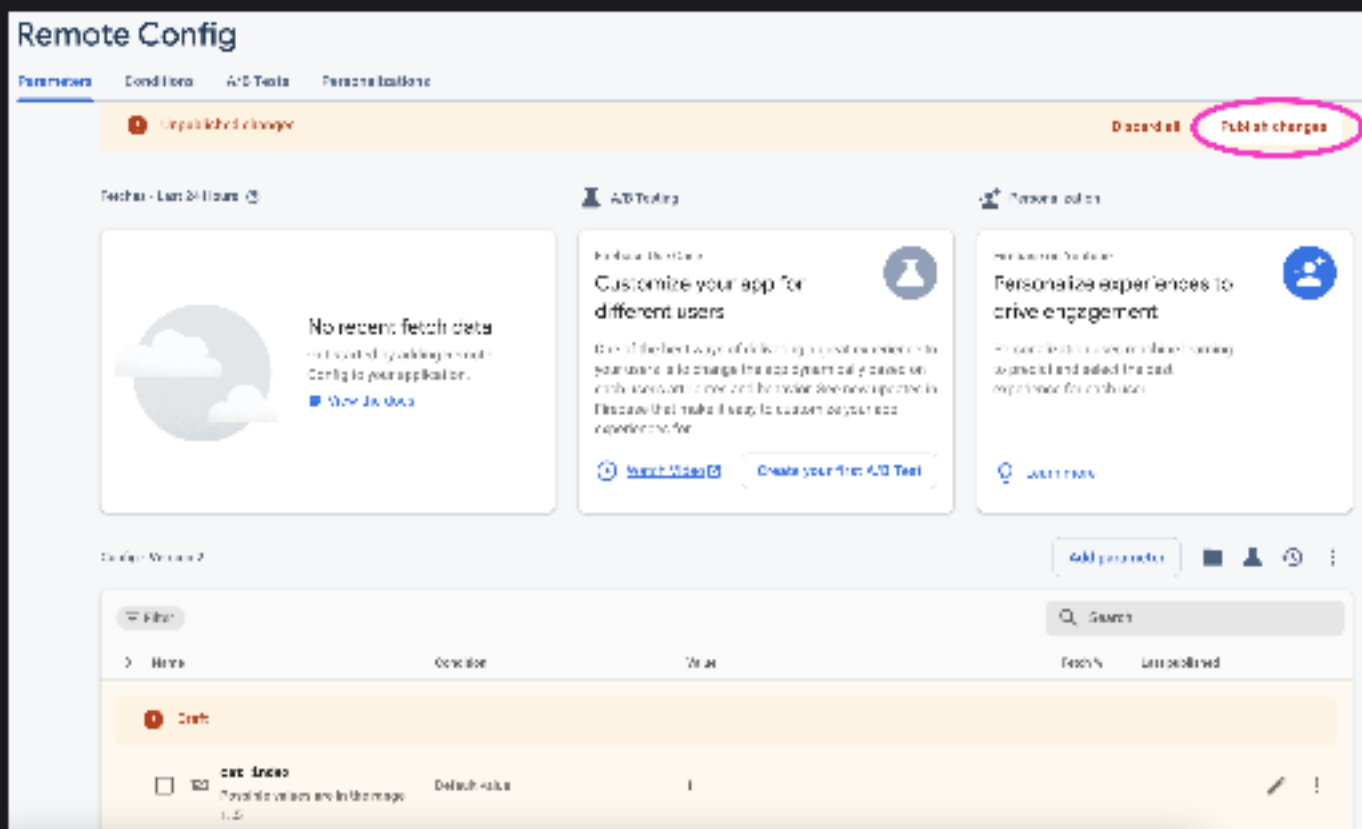


catIndex  
= 4



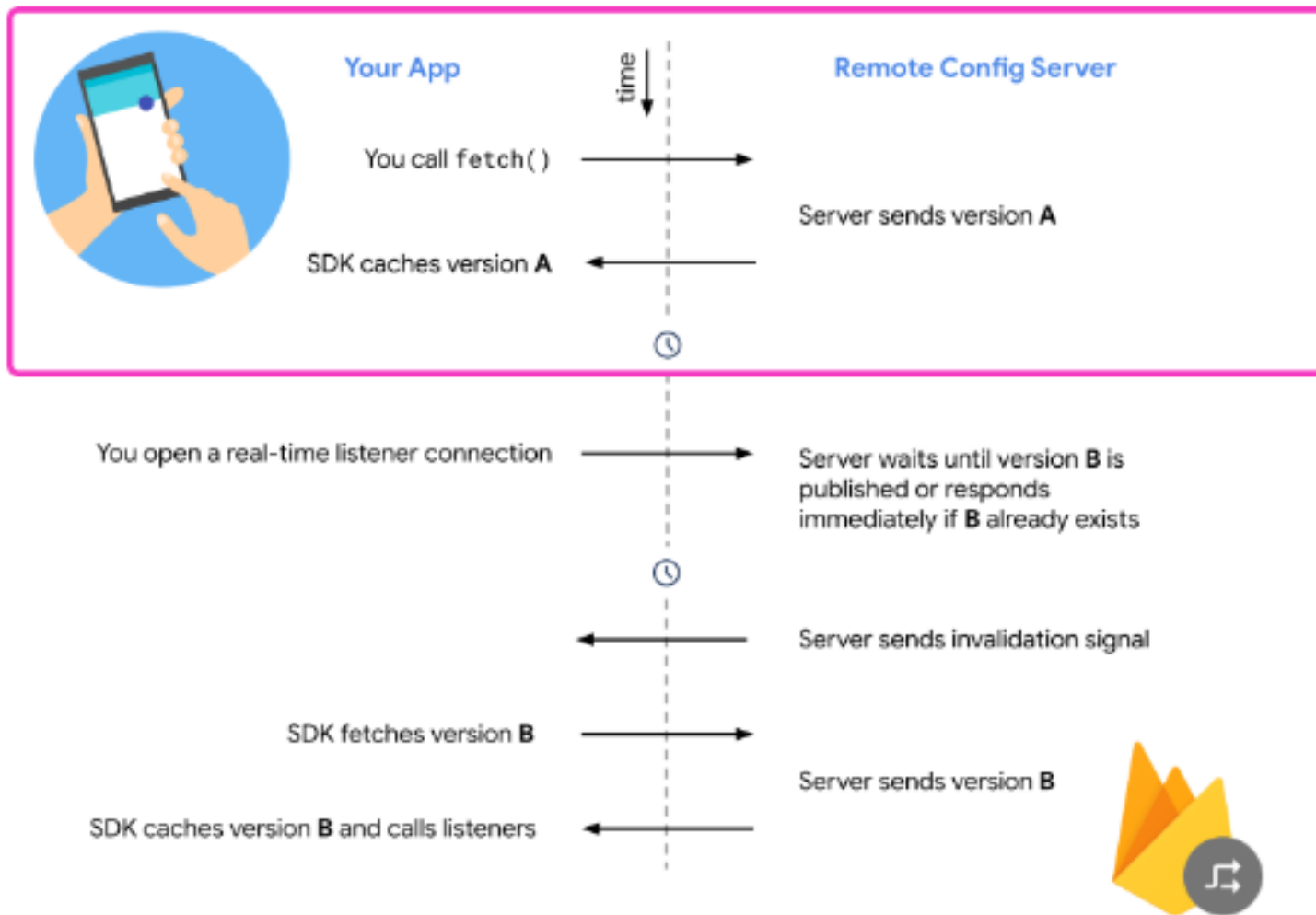
catIndex  
= 5





# Step 3: Publish changes on Firebase console





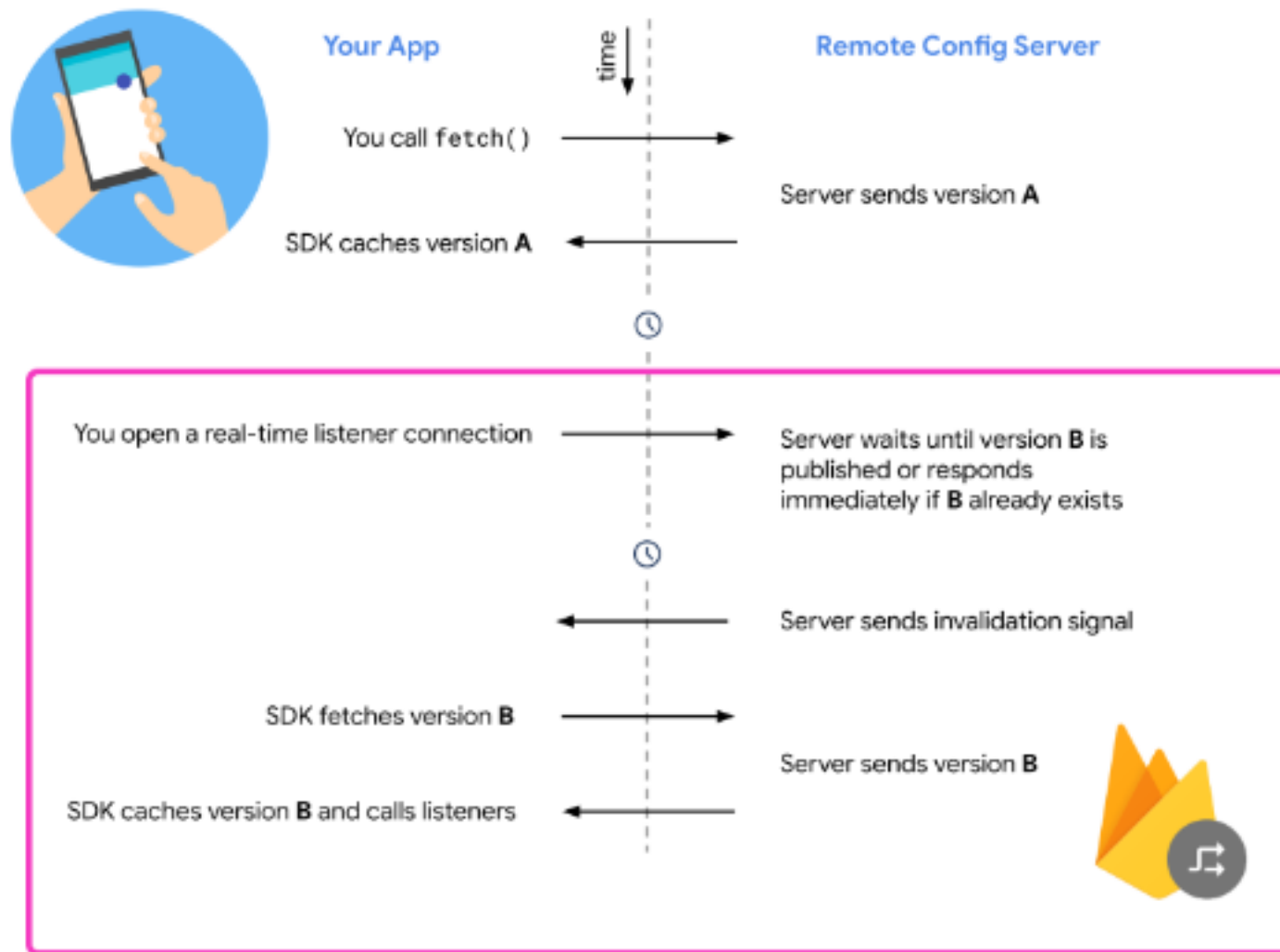
Source: <https://firebase.google.com/docs/remote-config/real-time>



```
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





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) ⓘ  Data type

Description

Default value   Use in-app default

[+ Add new](#) Cancel Save

Update cat index to 4



But what if we want to limit the values



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



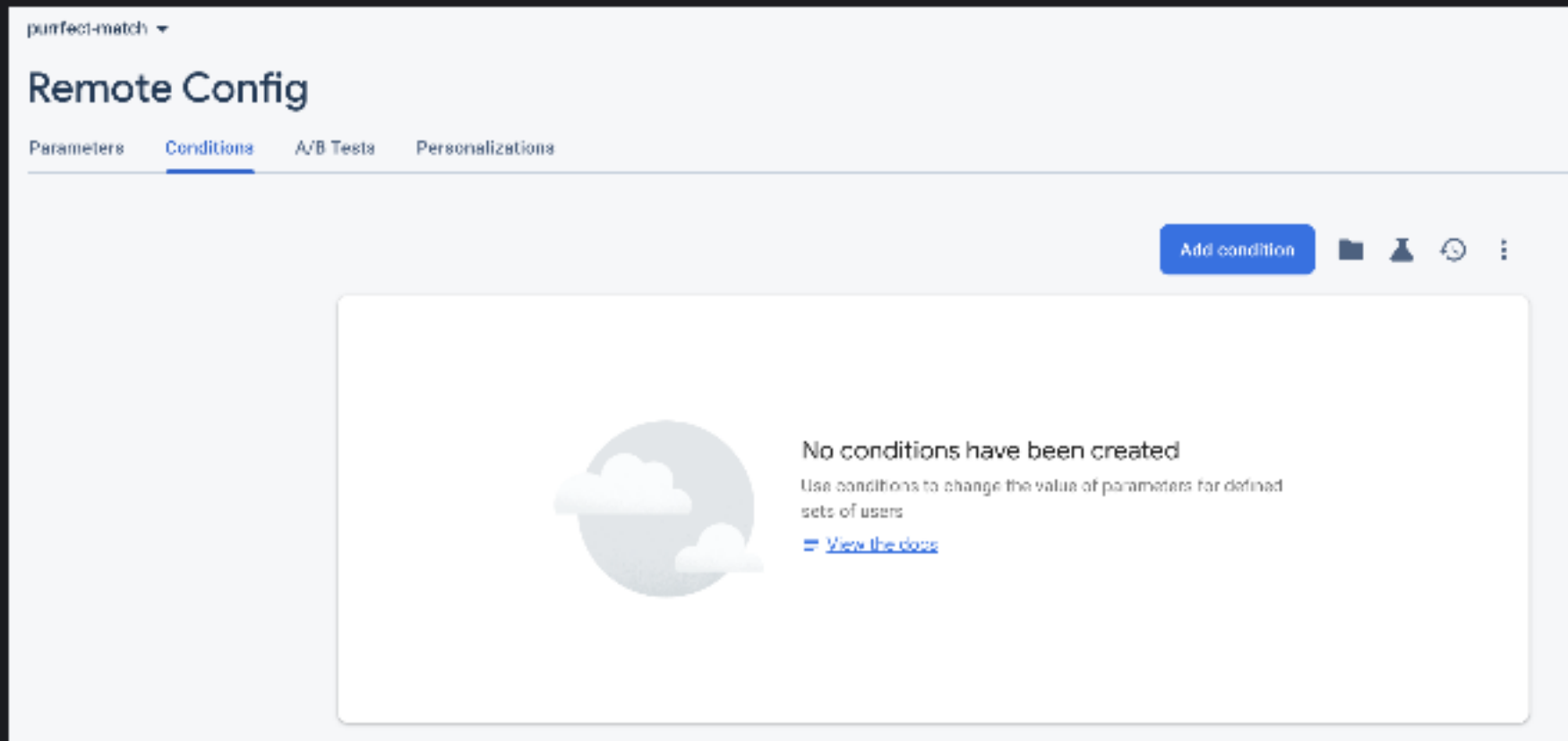
But what if we want to limit the values  
... to a specific *location* ?  
... or *language* ?



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







# 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

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 ✕

Parameter name (key) ? Data type

Boolean ▾

Description

Default value  Use in-app default

▾

+ Add new ▾ Cancel Save

# Step 3: Create a new FRC param



### Create parameter ✕

Parameter name (key) ?  Data type  Boolean ▾

Description

Default value   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



Create parameter ✕

Parameter name (key) ⓘ  Data type  ▼

Description

Condition  ✎ ⓘ

value   Use in-app default

Default value   Use in-app default

← Add new ▼ Cancel Save

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

@mkobuolys | @dariadroid



▼	<input type="checkbox"/>	<input checked="" type="checkbox"/> show_valentines_promo	-----	Valentines Day Promo	-----	true	0%	Darja Orlova	
				⋮				Just now	✎ ⋮
				Default value	-----	false	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

Deep Orange

Applies If...

App	▼	iOS- perfect_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



<b>App</b>
<b>Version</b> Select an app to target by version
<b>Build number</b> Select an app to target by build number
<b>First open</b> Select an app to target by first open time
<b>Platform</b>
<b>Languages</b> The inferred language for the user's device
<b>Country/Region</b> The inferred region or country for the user's device
<b>User audience(s)</b>  Audience to which the user belongs
<b>User property</b>  Match against the value of an Analytics User Property
<b>User in random percentage</b> A fractional percentage randomly and persistently assigned to each user on a per-project basis.
<b>Date/time</b> The date/time of the request
<b>Installation ID</b> The Firebase installation ID of the instance
<b>User exists</b> Targets all users within this project

# Possible conditions params



Lesson #2

# Safer decisions with Crashlytics & feature flags



12:58



## 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](#)





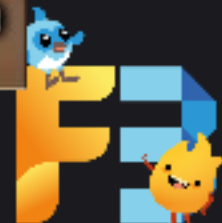
@mkobuolys | @dariadroid



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

**EVERYONE GETS A NEW FEATURE**

imgflip.com





Purrfect Match



Spooky



## 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>





# App monitoring



```
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());  
}
```



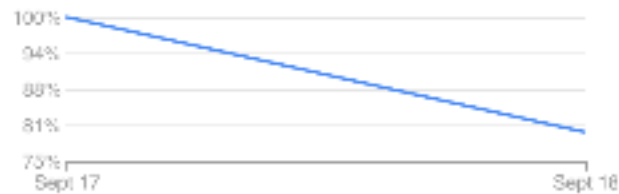
Filter

Custom  
17 Sept - 19 Sept

## Crash-free statistics

Crash-free users

83.33%



## Trends

[Latest Release](#)

Crashes	Users	Non-fatal	Users	ANRs	Users
1	1	7	1	---	---



## Issues

[Search by user ID](#)

Filter Issues

Issue state - \*Open

Search issue title, subtitle or keys

Issues	Versions	Events ↓	Users
<p><b>Non-fatal</b>   package:perfect_match/video_call/widgets/video_call_page.dart:35</p> <p><input type="checkbox"/> <b>_VideoCallPageState.initState</b></p> <p>io.flutter.plugins.firebase.crashlytics.FlutterError - CameraException(Camera not available, Camera is not available on this device.)</p>	1.0.0 - 1.0.0	7	1





## Crashlytics

Alert	In Firebase	Email
A new fatal issue is reported		<input type="checkbox"/> ▼
A new non-fatal issue is reported		<input type="checkbox"/> ▼
A new ANR issue is reported		<input type="checkbox"/> ▼
Trending issues are detected		<input checked="" type="checkbox"/> ▼
A regression is reported	<input checked="" type="checkbox"/> ▼	<input checked="" type="checkbox"/> ▼
A velocity alert is reported	<input checked="" type="checkbox"/> ▼	<input checked="" type="checkbox"/> ▼
A missing Apple dSYM has been detected		<input checked="" type="checkbox"/> ▼



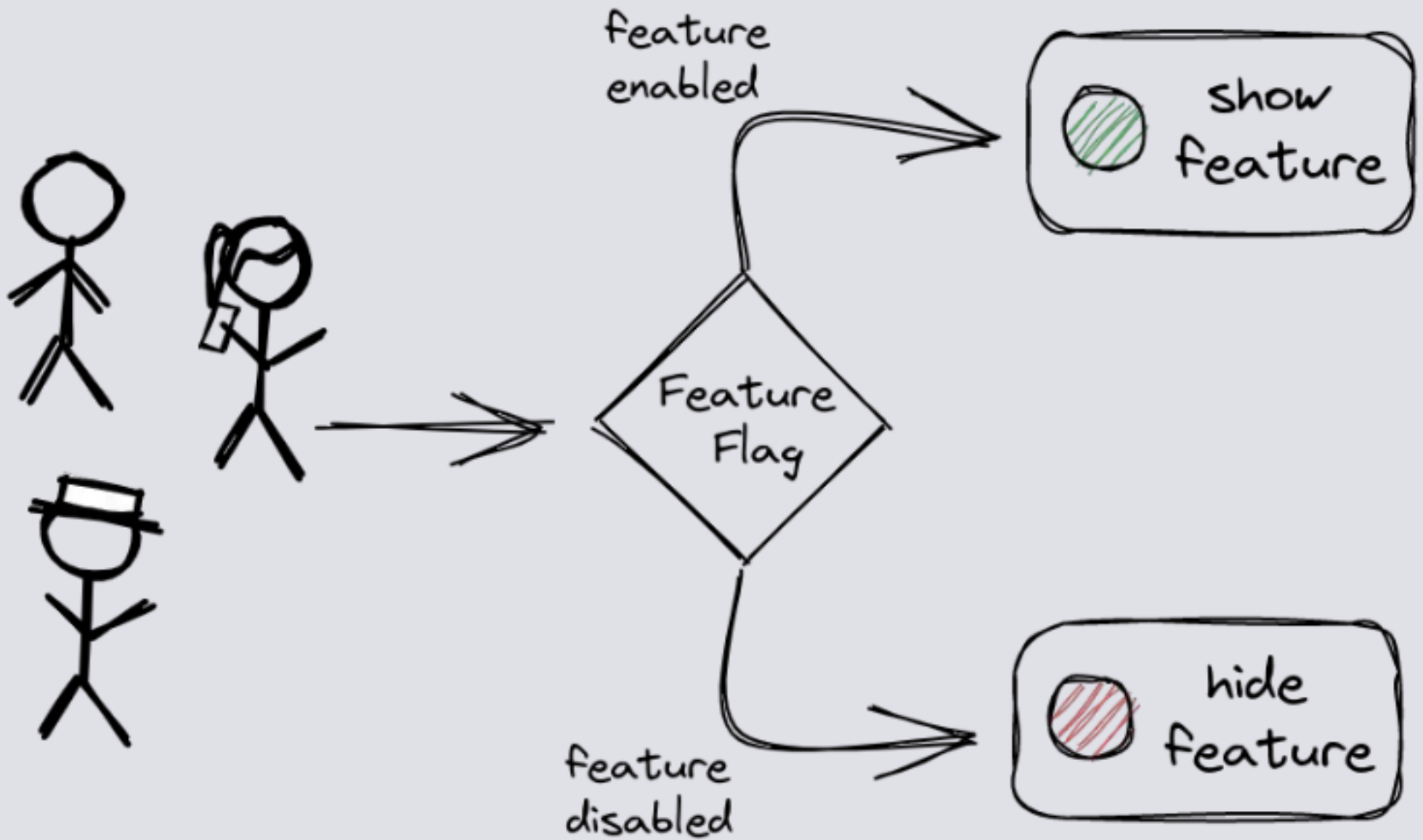


**BUGS**

**APP**

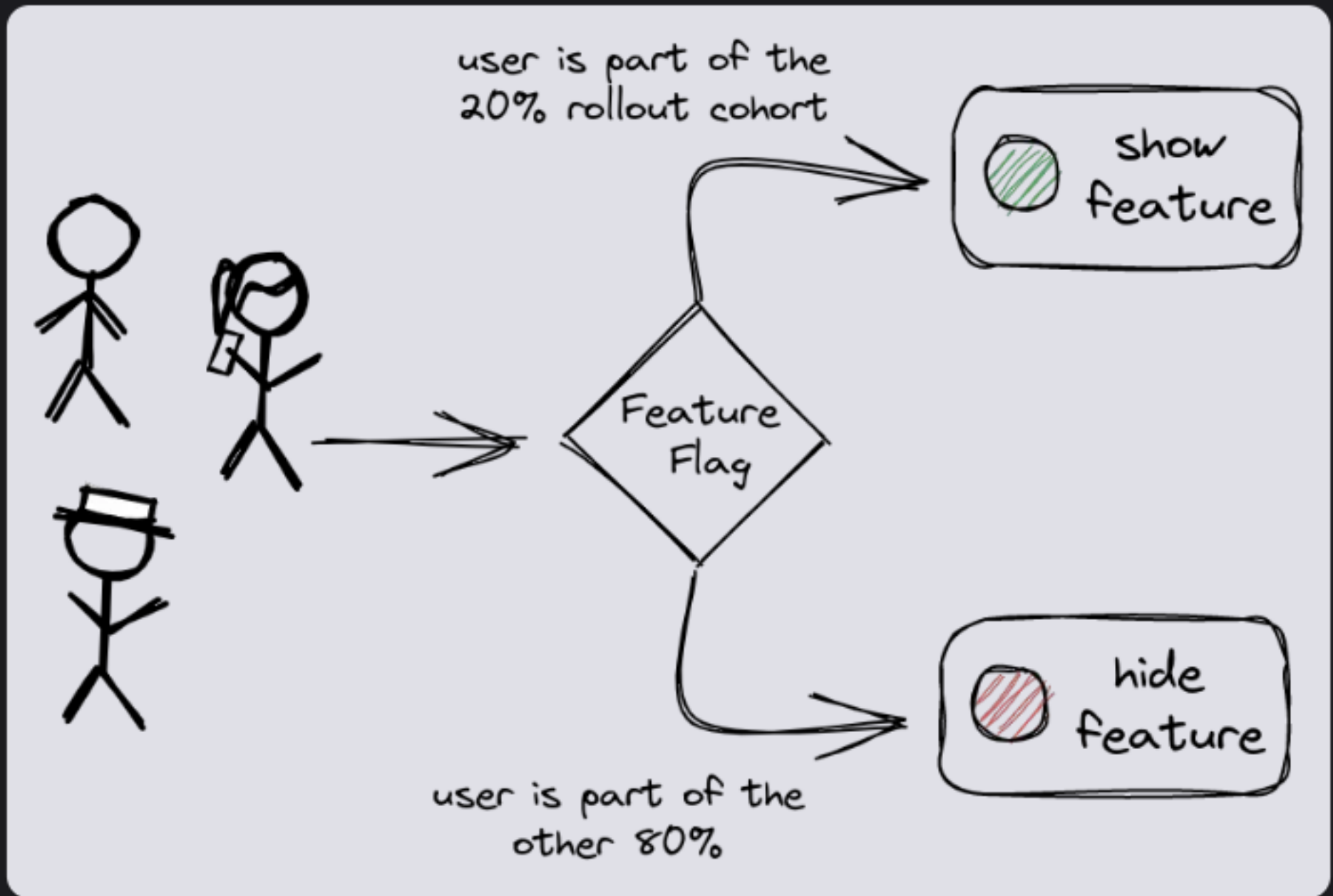






Source: <https://reflectoring.io/java-feature-flags/>





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

Cyan

Applies if...

User in random percentage

0



20

Edit seed

and

Fewer than 100 users match this condition ?

Cancel

Create condition



## 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



```
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) 🔗  Data type  Boolean ▼

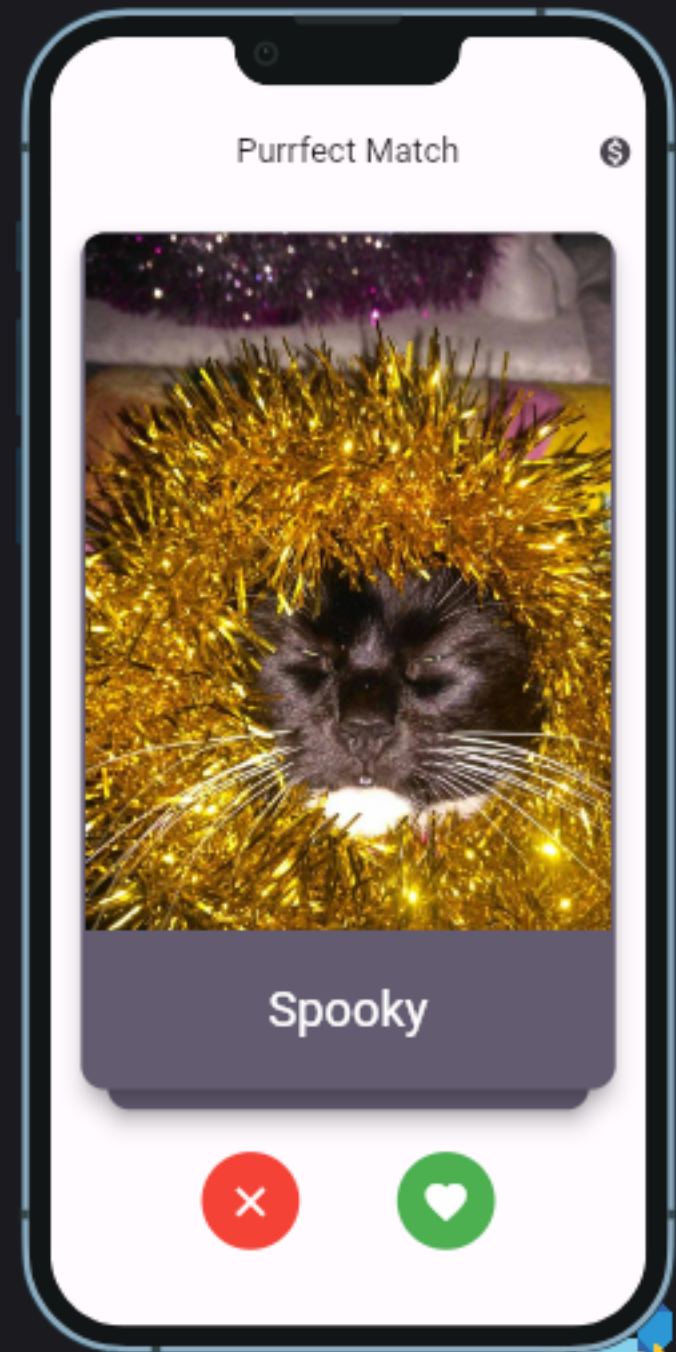
Description

Condition  ✎ ✕

Value   Use in-app default

Default value   Use in-app default

+ Add new Cancel Save



Video call enabled:



# Turning Users into Subscribers

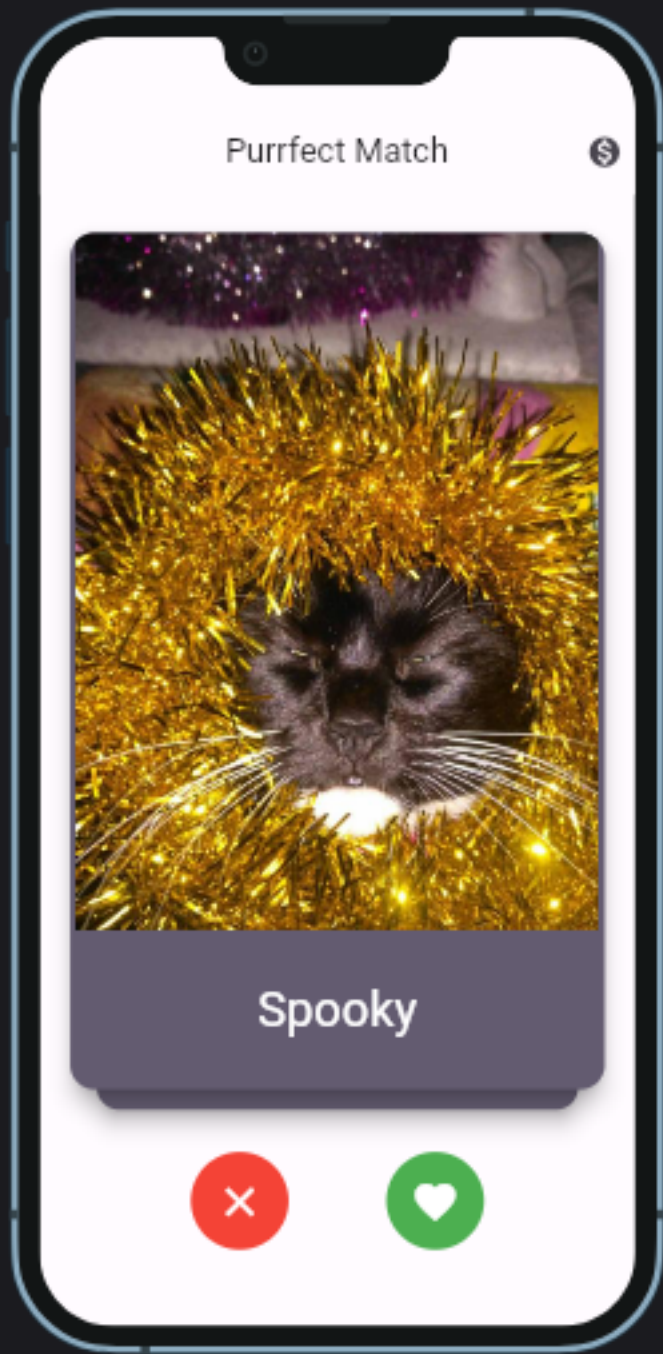


Lesson #3

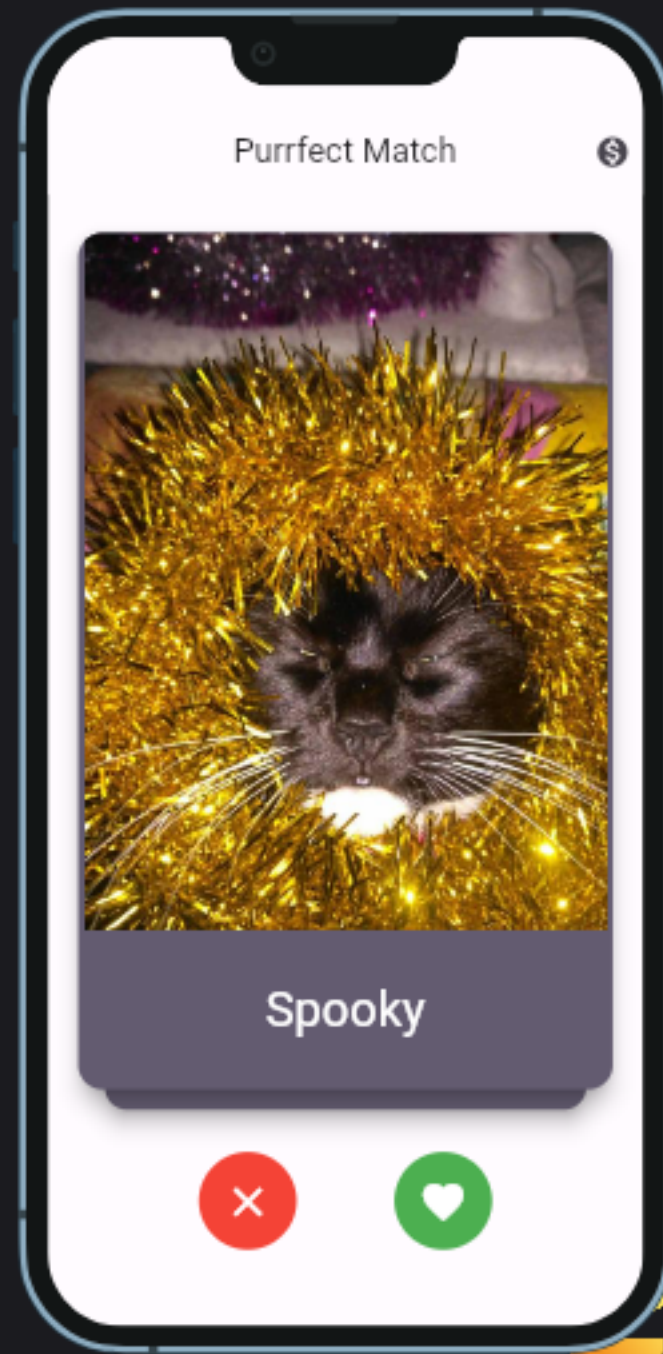
A/B testing – even smarter  
decisions, even faster







VS





ABC

paywall\_type

Possible values: "simple",  
"detailed"

Paywall Simple 50%

simple

Paywall Detailed 50%

detailed

# Option 1: Distribute via Conditions & analyze with Analytics





Firestore A/B Testing

Option 2: A/B testing



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

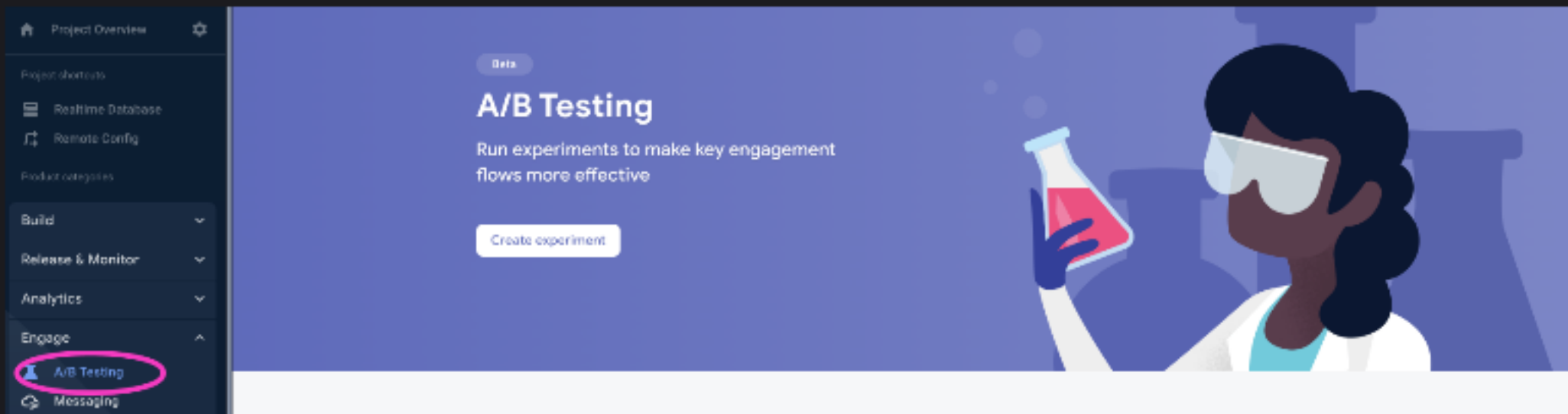
# Step 1: Use as a regular FRC param



```
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





# Step 3: Firebase Console -> Engage

## -> A/B testing





# Step 4: Create Experiment



Which service do you want to experiment with?



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

**i** This experiment will run for up to 90 days

Experiment name: Paywall type (Android)      Experiment description (optional):

Next

2 Targeting

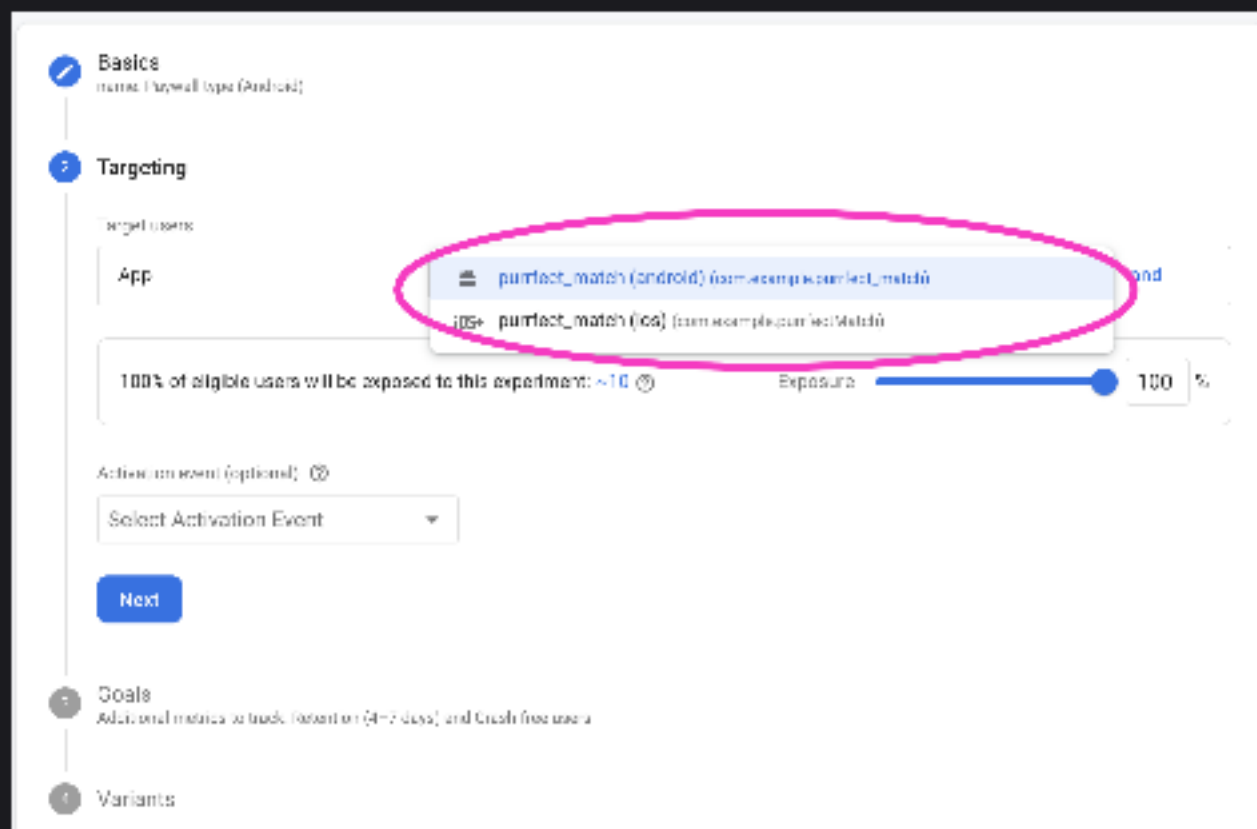
3 Goals  
Additional metrics to track: Revenue (4-7 days) and Churn (30 days)

4 Variants

Cancel    Review

# Step 6: Set experiment name & description





# Step 7: Select app & user percentage





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

**2 Targeting**

Target users

App

-  purrfect\_match (android) (com.example.purrfect\_match)
-  purrfect\_match (ios) (com.example.purrfectMatch)

and

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

Activation event (optional) ⓘ

Select Activation Event

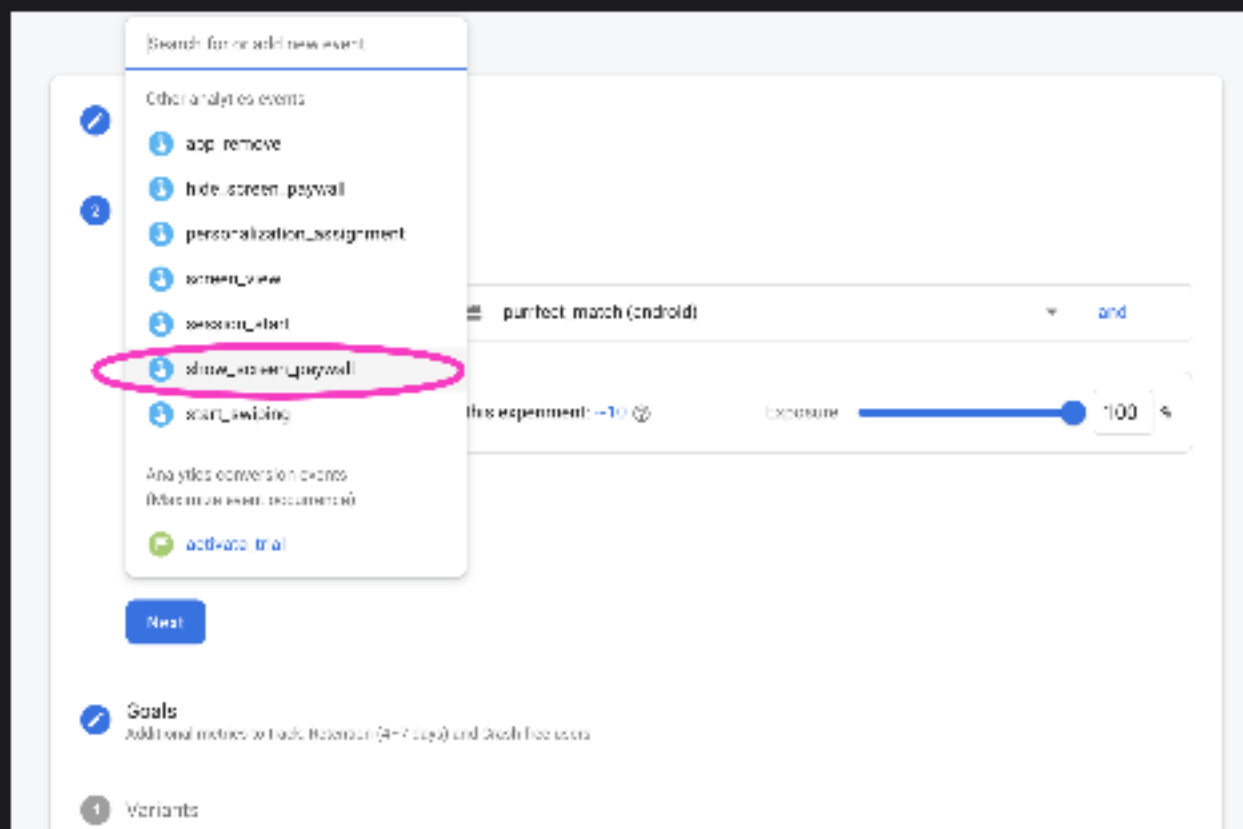
Next

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

**4 Variants**

# Step 8: Select activation event





# Step 9: Select

# "show\_screen\_paywall"



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

**2 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"



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

**2 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 ×

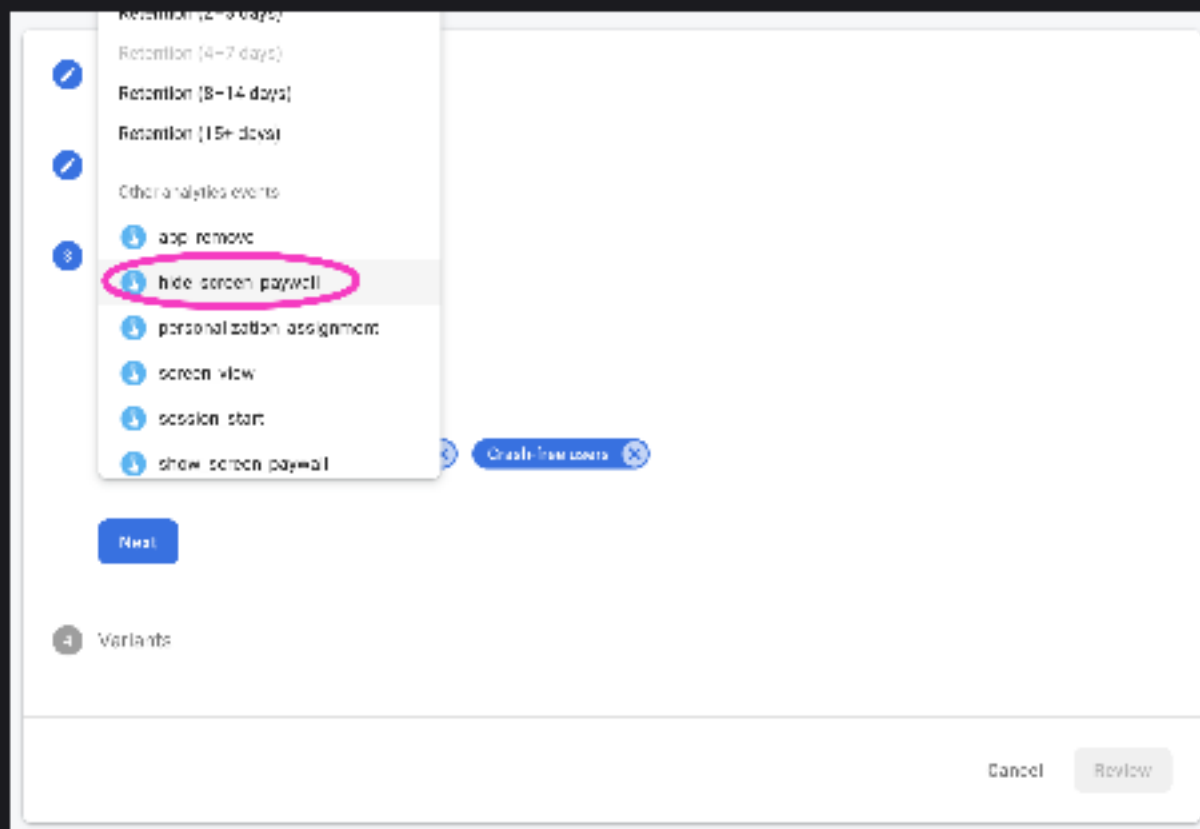
Next

**4 Variants**

Cancel Review

# Step 11: Add another metric





Step 12: Select

"hide\_screen\_paywall"



4 Variants

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"





## Experiment overview

Draft

Your experiment is ready to start

Edit

Start experiment

Total users

0

Remote Config

Created Just now

show\_screen\_paywall

active\_title

100% of users matching 1 criteria

2 variants

0 test devices

## Start experiment?



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

Cancel

Start

## Experiment results

How to read this chart

show\_screen\_paywall

Modeled Conversion Rate



7 days

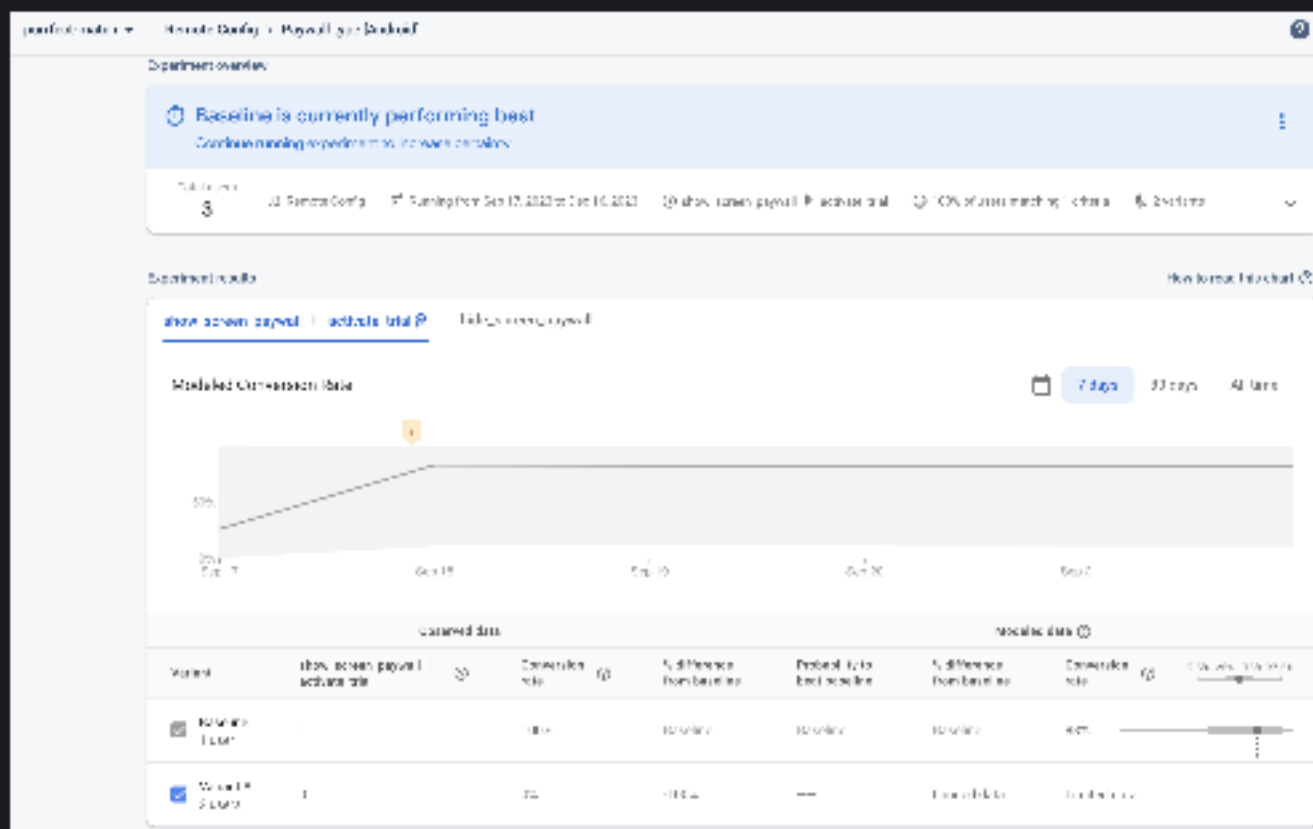
30 days

All time

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



Personalization  
for Remote Config

AI, EVERYWHERE

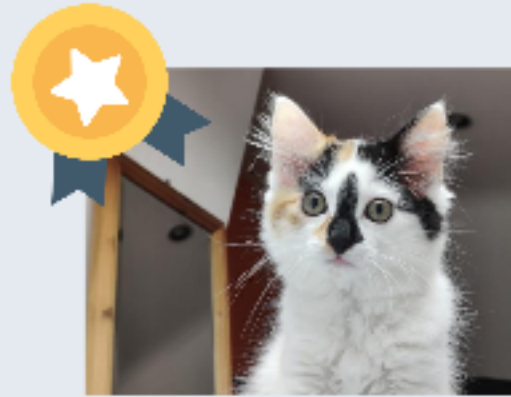
imgflip.com

@mkobuolys | @dariadroid

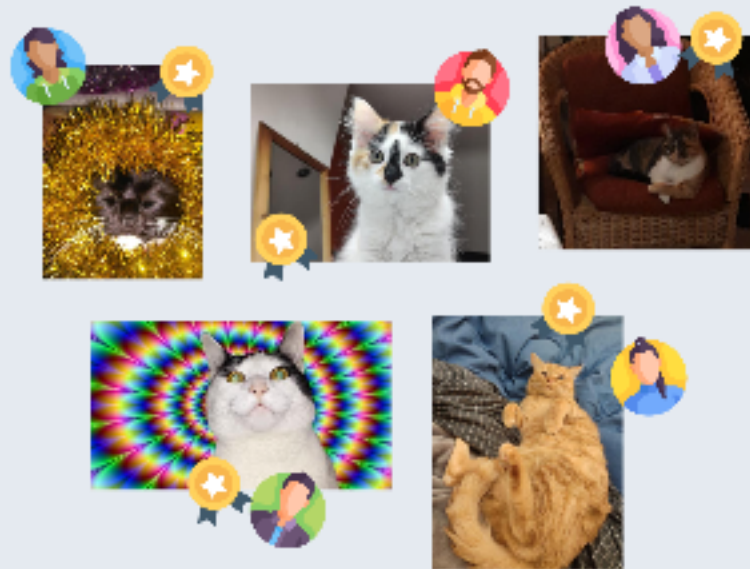


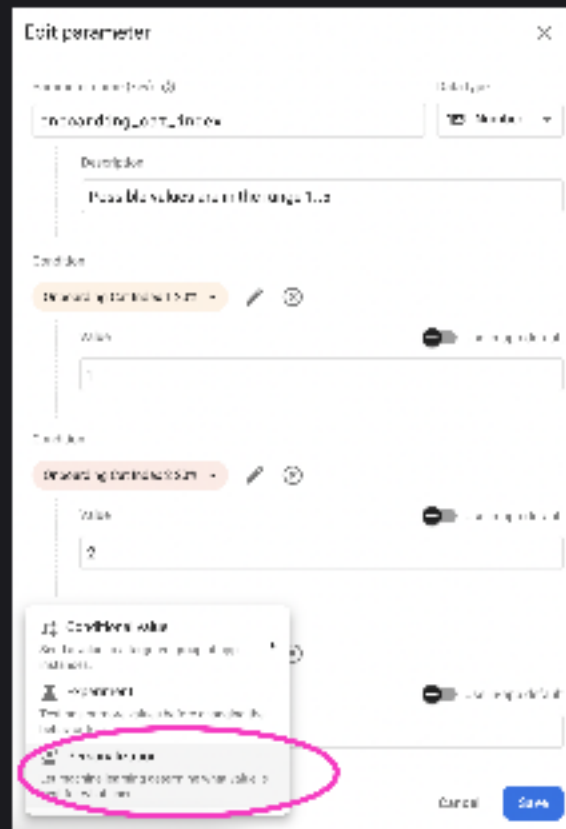


A/B Testing



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 desires. [Learn more](#) [?]

**1 Enter alternative values**

For each alternative value, you can enter a value that may be selected and returned to an end user by the personalization algorithm.

**1 Alternative 1**

Value

**2 Alternative 2**

Value

[+ Add alternative value](#) (5 max)

[Next](#)

**2 Choose objective** [?]

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 machine and device. [Learn more](#)

5 alternatives

2 Choose objective 

The algorithm finds which alternative is best to serve to each user to maximize your selected objective. Eg. The algorithm learns which ad frequency is best for user eye.

Objective

start swiping

Optimize for

COUNT SUM

Additional metrics

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

Add metric +

Next

3 Target condition

4 Name personalization

Cancel Save

### 3. Select an objective to track





← Create personalization

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

- ✓ Enter alternative values  
Tabbed view
- ✓ Choose objective  
User segment
- 3 Target condition**  
This condition will identify the users who will be personalized. A portion of this group will be reserved for the baseline and will receive random values to act as a benchmark for the performance of your personalization.  
Onboarding personalization
- 4 Name personalization

Next


Cancel Save

## 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
- 1 Name personalization**

Name

Description (optional)

Cancel Save

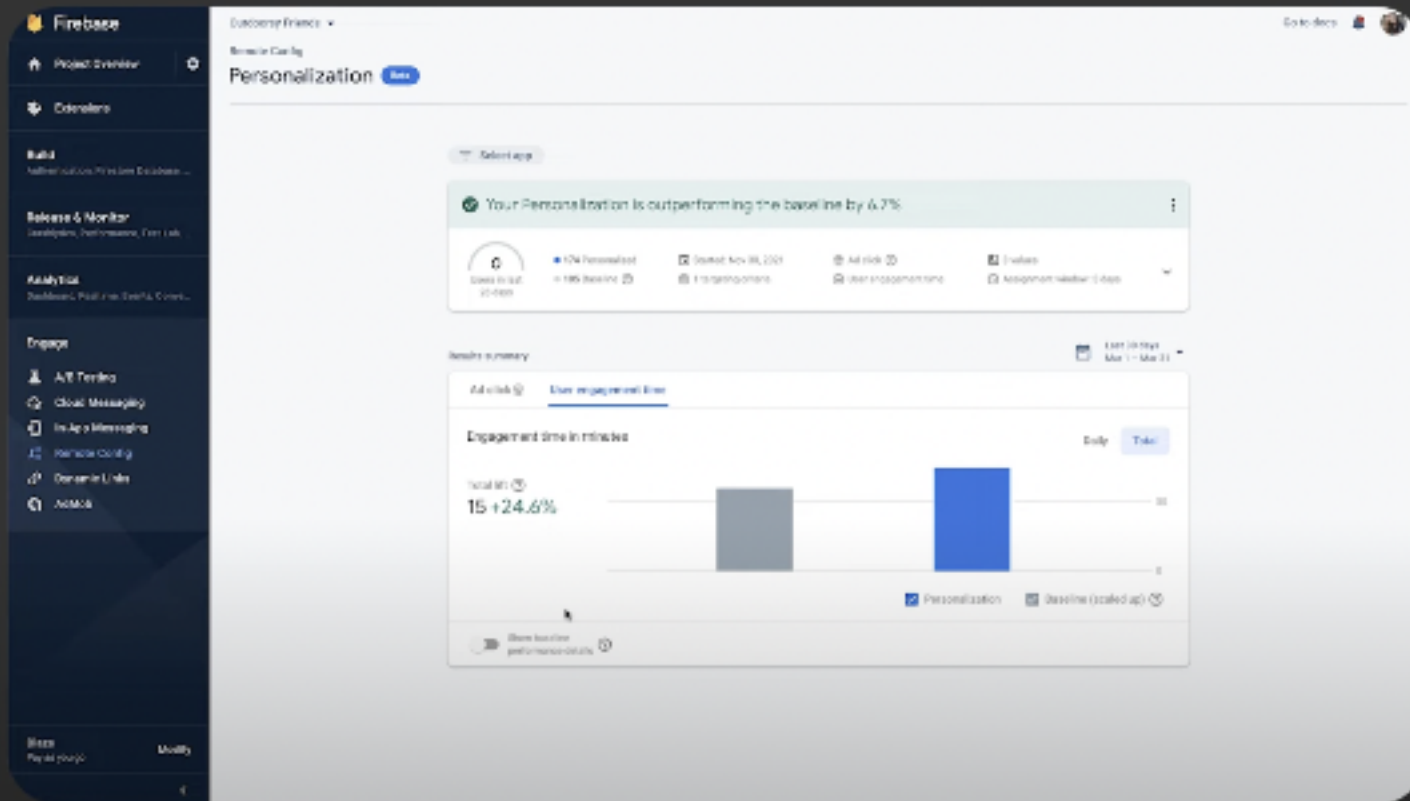
## 5. Set name and description





# See applied personalization in the console





12:22 / 13:13



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

