Google Cloud

Next '24

Use Firebase for faster, easier mobile application development

Proprietary

Jon Mensing

Group Product Manager,
Google Cloud

Rich Hyndman

DevRel Manager,
Google Cloud

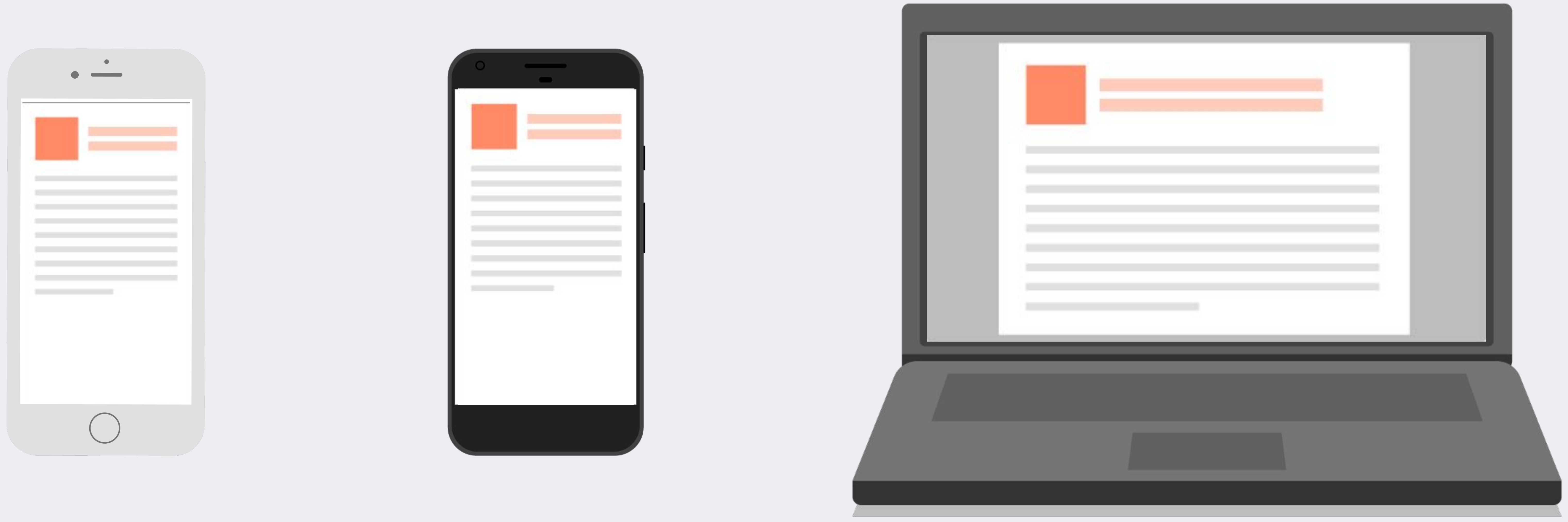
Firebase is Google’s app development platform

Our mission is to help developers build and grow apps users love

Who uses Firebase?

Firebase has significant adoption among many enterprises

algolia	Alibaba.com	Ctrip	UBISOFT	wayfair
The Economist	GAMELOFT	GAME NEXA	HALFBRICK	The New York Times
InsightTimer	JAM CITY	rowy	LE FIGARO	workday.
peoplefun	AHOY games	NEXON	lyft	magazineluiza
Pomelo Games		Squadle	Tapps Games	tapple
hotstar				



iOS

Apple

Android

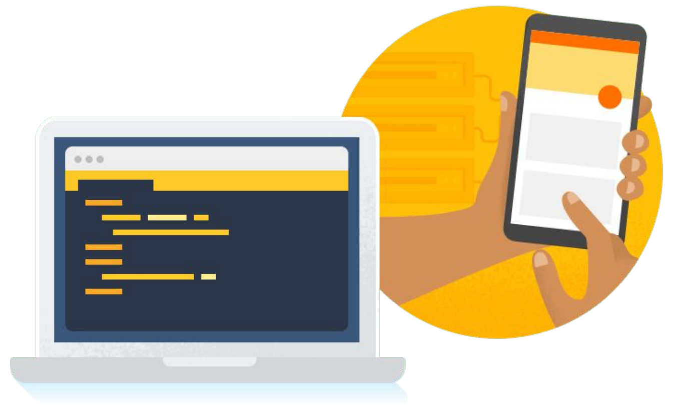
Web

C++

Unity

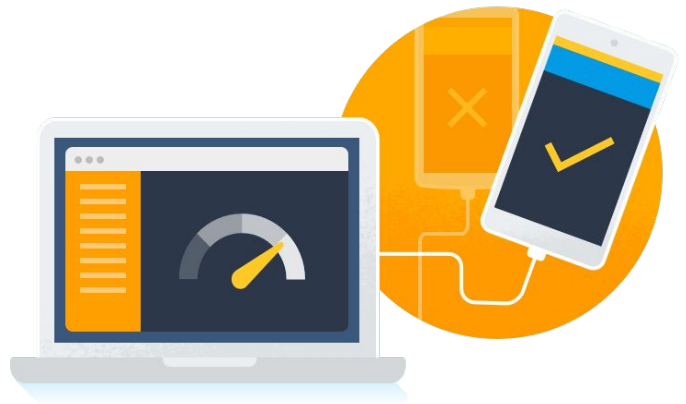
Flutter

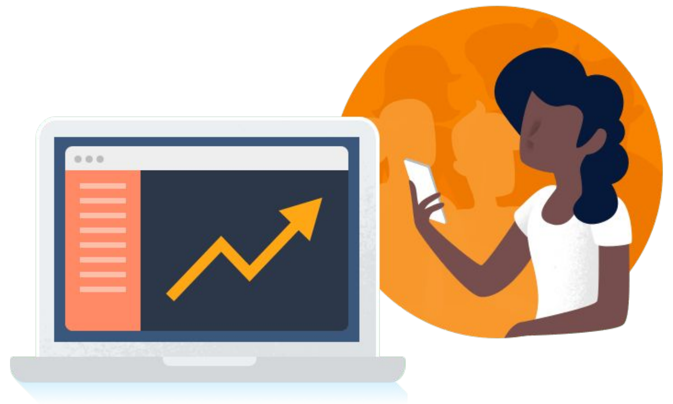
Firebase is a **platform of tools and cloud services** that helps solve **three** core problems in your app lifecycle



1. Develop apps faster with fully-managed backend services

2. Run apps with confidence through testing and monitoring





3. Engage users effectively with better insights and rollout control

Firebase products

Build your backend

- Authentication
- Firestore
- Realtime Database
- Cloud Storage
- Hosting
- App Check
- Cloud Functions
- Extensions
- Machine Learning

Improve app quality

- Crashlytics
- Performance Monitoring
- Test Lab
- App Distribution

Engage users effectively

- Remote Config
- A/B Testing
- Personalization
- Cloud Messaging
- In-App Messaging
- Analytics

Firebase products

Build your backend

- Authentication
- App Check
- Firestore
- Cloud Functions
- Realtime Database
- Extensions
- Cloud Storage
- Machine Learning
- Hosting

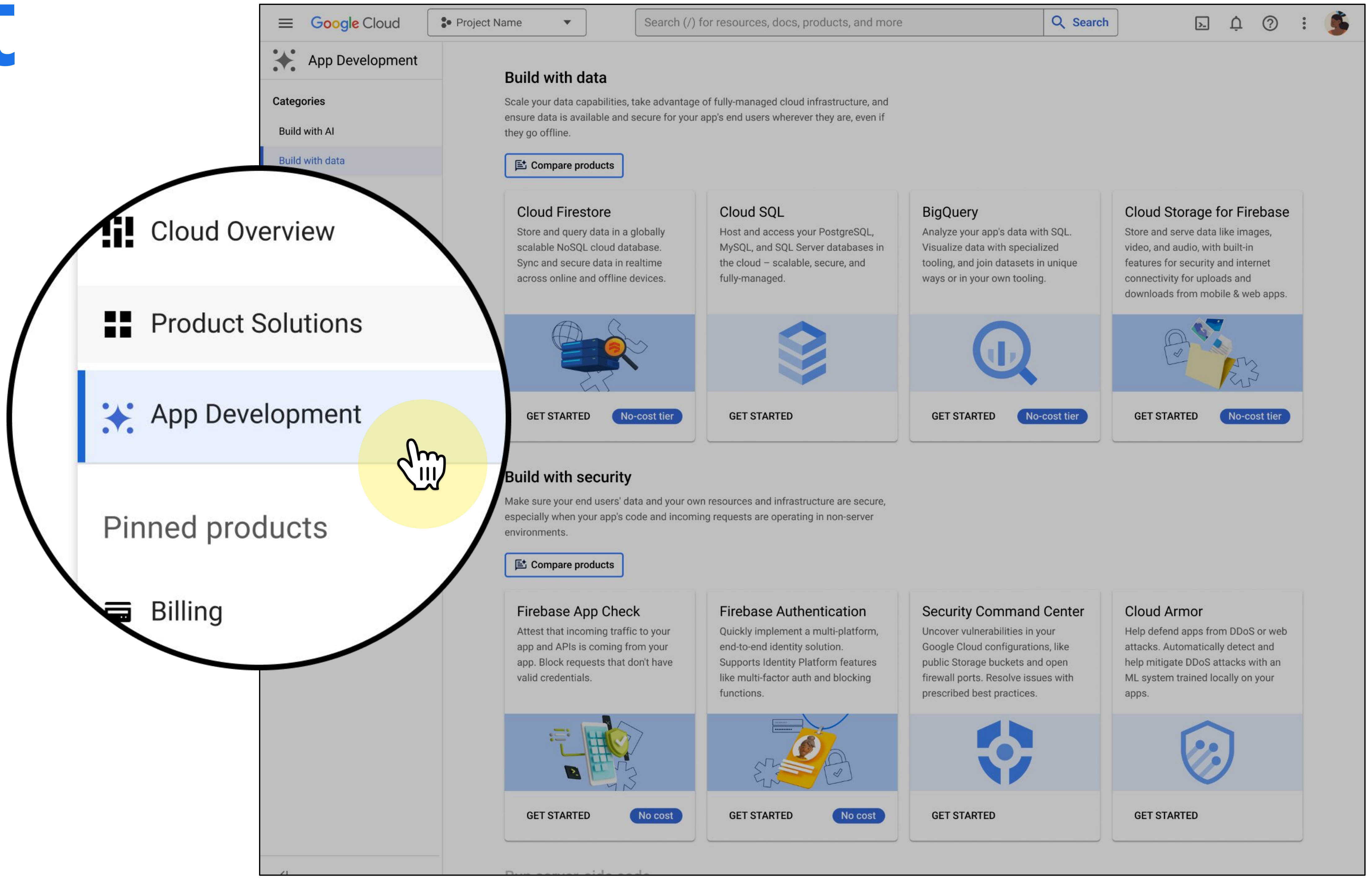
Improve app quality

- Crashlytics
- Performance Monitoring
- Test Lab
- App Distribution

Engage users effectively

- Remote Config
- Cloud Messaging
- A/B Testing
- In-App Messaging
- Personalization
- Analytics

App Development in Google Cloud



Build with data

Scale your data capabilities, take advantage of fully-managed cloud infrastructure, and ensure data is available and secure for your app's end users wherever they are, even if they go offline.

Compare products

Cloud Firestore	Cloud SQL	BigQuery	Cloud Storage for Firebase
Store and query data in a globally scalable NoSQL cloud database. Sync and secure data in realtime across online and offline devices.	Host and access your PostgreSQL, MySQL, and SQL Server databases in the cloud – scalable, secure, and fully-managed.	Analyze your app's data with SQL. Visualize data with specialized tooling, and join datasets in unique ways or in your own tooling.	Store and serve data like images, video, and audio, with built-in features for security and internet connectivity for uploads and downloads from mobile & web apps.
GET STARTED No-cost tier	GET STARTED	GET STARTED No-cost tier	GET STARTED No-cost tier

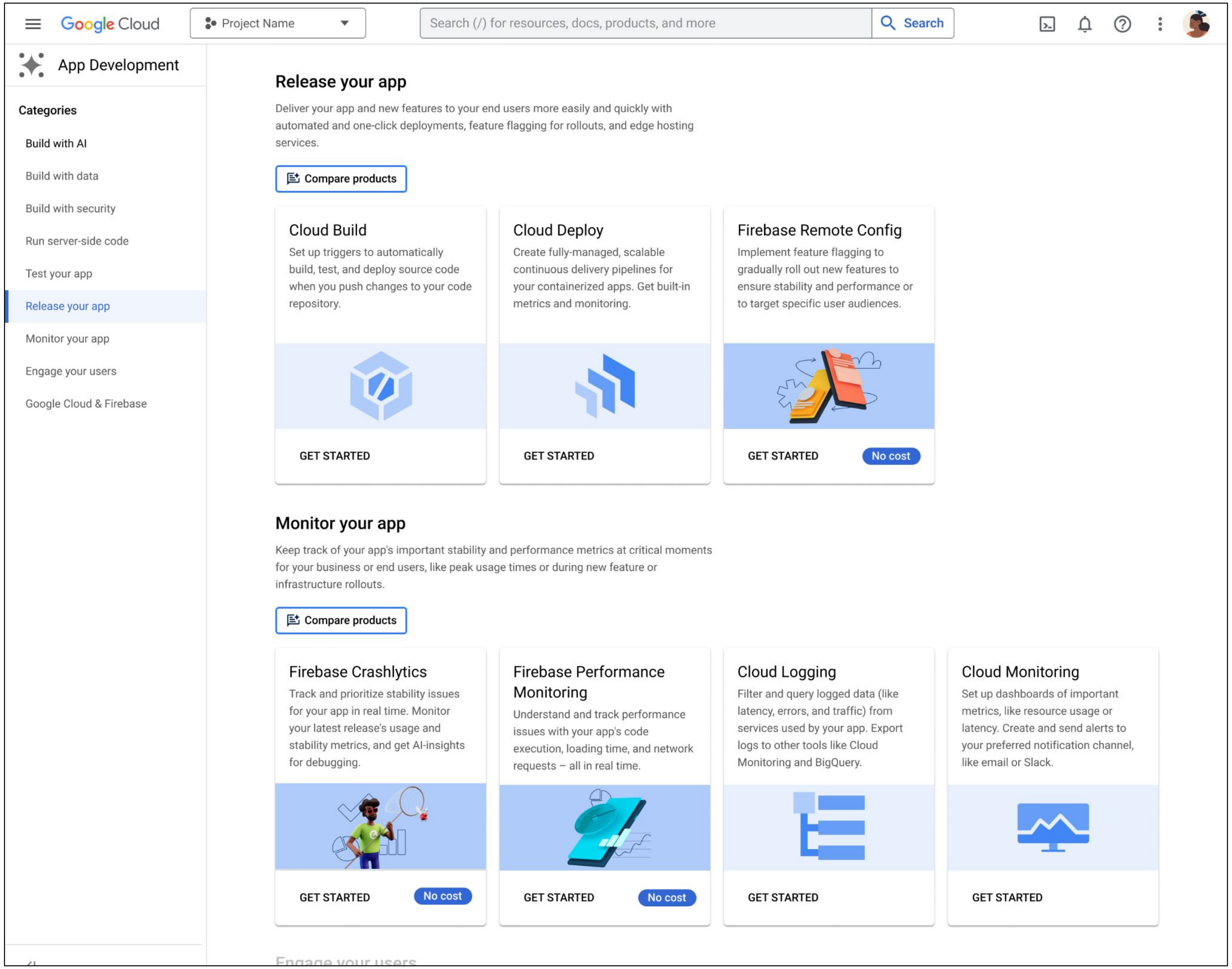
Build with security

Make sure your end users' data and your own resources and infrastructure are secure, especially when your app's code and incoming requests are operating in non-server environments.

Compare products

Firebase App Check	Firebase Authentication	Security Command Center	Cloud Armor
Attest that incoming traffic to your app and APIs is coming from your app. Block requests that don't have valid credentials.	Quickly implement a multi-platform, end-to-end identity solution. Supports Identity Platform features like multi-factor auth and blocking functions.	Uncover vulnerabilities in your Google Cloud configurations, like public Storage buckets and open firewall ports. Resolve issues with prescribed best practices.	Help defend apps from DDoS or web attacks. Automatically detect and help mitigate DDoS attacks with an ML system trained locally on your apps.
GET STARTED No cost	GET STARTED No cost	GET STARTED	GET STARTED

Release & monitor your app



App Development

Categories

- Build with AI
- Build with data
- Build with security
- Run server-side code
- Test your app
- Release your app
- Monitor your app
- Engage your users
- Google Cloud & Firebase

Release your app

Deliver your app and new features to your end users more easily and quickly with automated and one-click deployments, feature flagging for rollouts, and edge hosting services.

Compare products

Cloud Build
Set up triggers to automatically build, test, and deploy source code when you push changes to your code repository.
GET STARTED

Cloud Deploy
Create fully-managed, scalable continuous delivery pipelines for your containerized apps. Get built-in metrics and monitoring.
GET STARTED

Firebase Remote Config
Implement feature flagging to gradually roll out new features to ensure stability and performance or to target specific user audiences.
GET STARTED No cost

Monitor your app

Keep track of your app's important stability and performance metrics at critical moments for your business or end users, like peak usage times or during new feature or infrastructure rollouts.

Compare products

Firebase Crashlytics
Track and prioritize stability issues for your app in real time. Monitor your latest release's usage and stability metrics, and get AI-insights for debugging.
GET STARTED No cost

Firebase Performance Monitoring
Understand and track performance issues with your app's code execution, loading time, and network requests – all in real time.
GET STARTED No cost

Cloud Logging
Filter and query logged data (like latency, errors, and traffic) from services used by your app. Export logs to other tools like Cloud Monitoring and BigQuery.
GET STARTED

Cloud Monitoring
Set up dashboards of important metrics, like resource usage or latency. Create and send alerts to your preferred notification channel, like email or Slack.
GET STARTED

App Quality

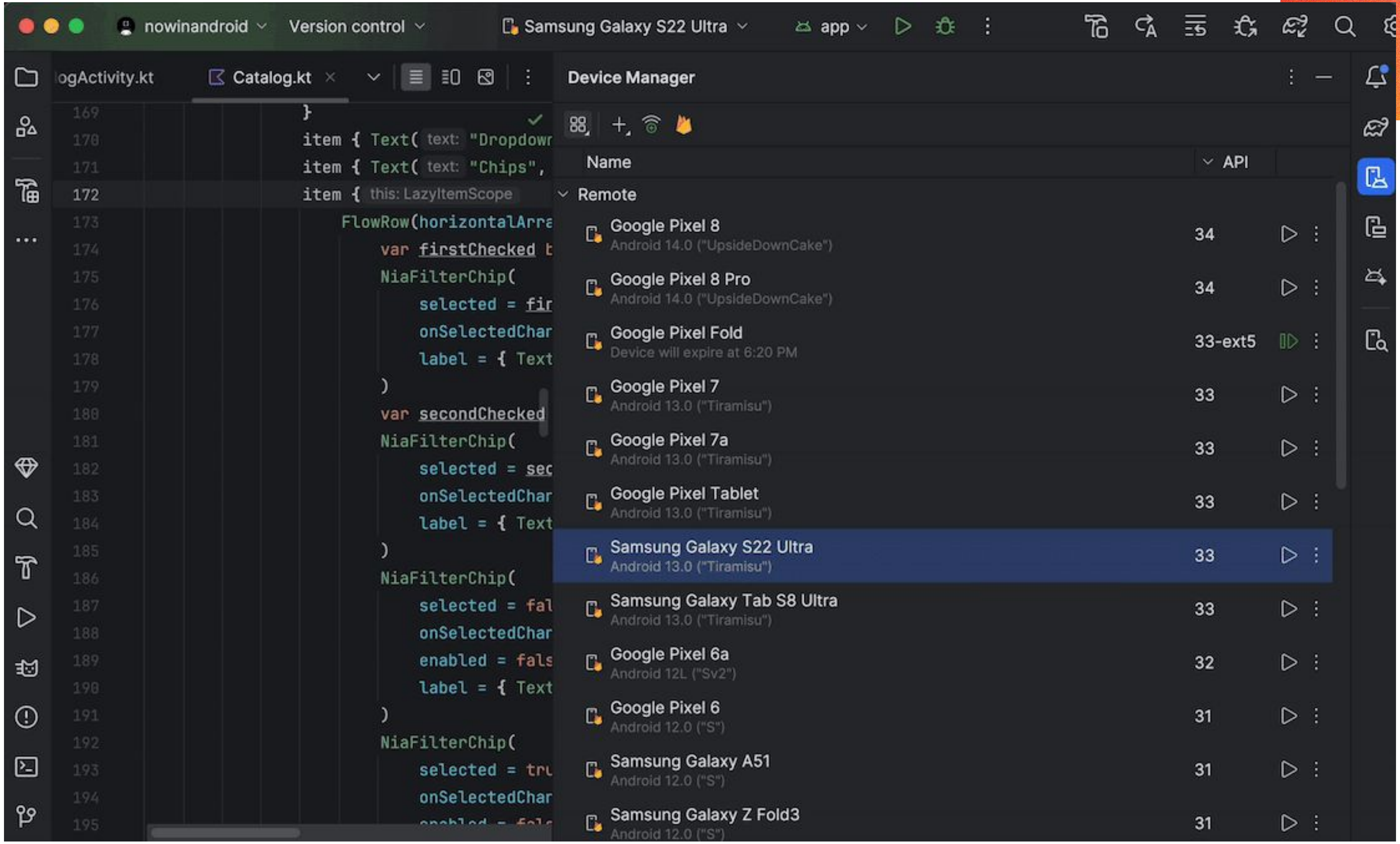
How to gain confidence in your app quality?

1. Test your app on lots of different devices
2. Get your app in the hands of real people for testing
3. Understand how your app behaves in the real world and troubleshoot as needed

Firebase Test Lab

Test Lab

Device Streaming



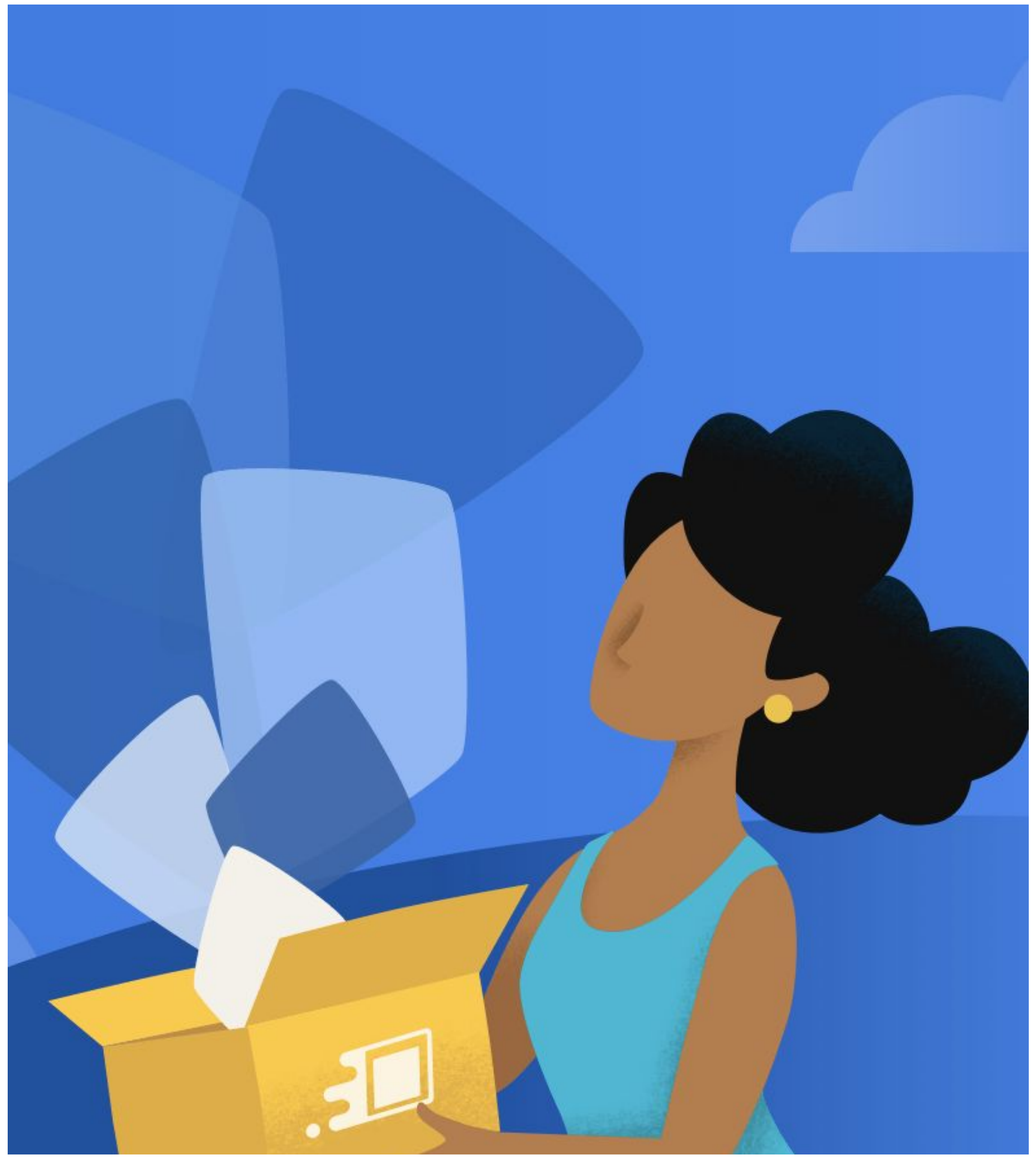
```
            }
            item { Text( text: "Dropdowr
            item { Text( text: "Chips",
            item { this: LazyItemScope
                FlowRow(horizontalArra
                    var firstChecked b
                    NiaFilterChip(
                        selected = fir
                        onSelectedChar
                        label = { Text
                    )
                    var secondChecked
                    NiaFilterChip(
                        selected = sec
                        onSelectedChar
                        label = { Text
                    )
                    NiaFilterChip(
                        selected = fal
                        onSelectedChar
                        enabled = fals
                        label = { Text
                    )
                    NiaFilterChip(
                        selected = tru
                        onSelectedChar
```

Name	API
Google Pixel 8 Android 14.0 ("UpsideDownCake")	34
Google Pixel 8 Pro Android 14.0 ("UpsideDownCake")	34
Google Pixel Fold Device will expire at 6:20 PM	33-ext5
Google Pixel 7 Android 13.0 ("Tiramisu")	33
Google Pixel 7a Android 13.0 ("Tiramisu")	33
Google Pixel Tablet Android 13.0 ("Tiramisu")	33
Samsung Galaxy S22 Ultra Android 13.0 ("Tiramisu")	33
Samsung Galaxy Tab S8 Ultra Android 13.0 ("Tiramisu")	33
Google Pixel 6a Android 12L ("Sv2")	32
Google Pixel 6 Android 12.0 ("S")	31
Samsung Galaxy A51 Android 12.0 ("S")	31
Samsung Galaxy Z Fold3 Android 12.0 ("S")	31

App Distribution

App Distribution

- Across iOS and Android ecosystems
- Email invitations
- Instant app delivery
- Individual and group stats



Inbox
Firebase

Automated Tester for App Distribution

Why Automated Smoke Testing for Android Apps?

Early Bug Detection: Catch critical flaws before they reach a wider audience, saving you time, money, and headaches.

Faster Test Cycles: Automate those essential pre-release checks, freeing up your time for more strategic testing.

Confidence in Releases: Your app is tested across a variety of API levels and devices of your choice using the Firebase console or the Firebase CLI.

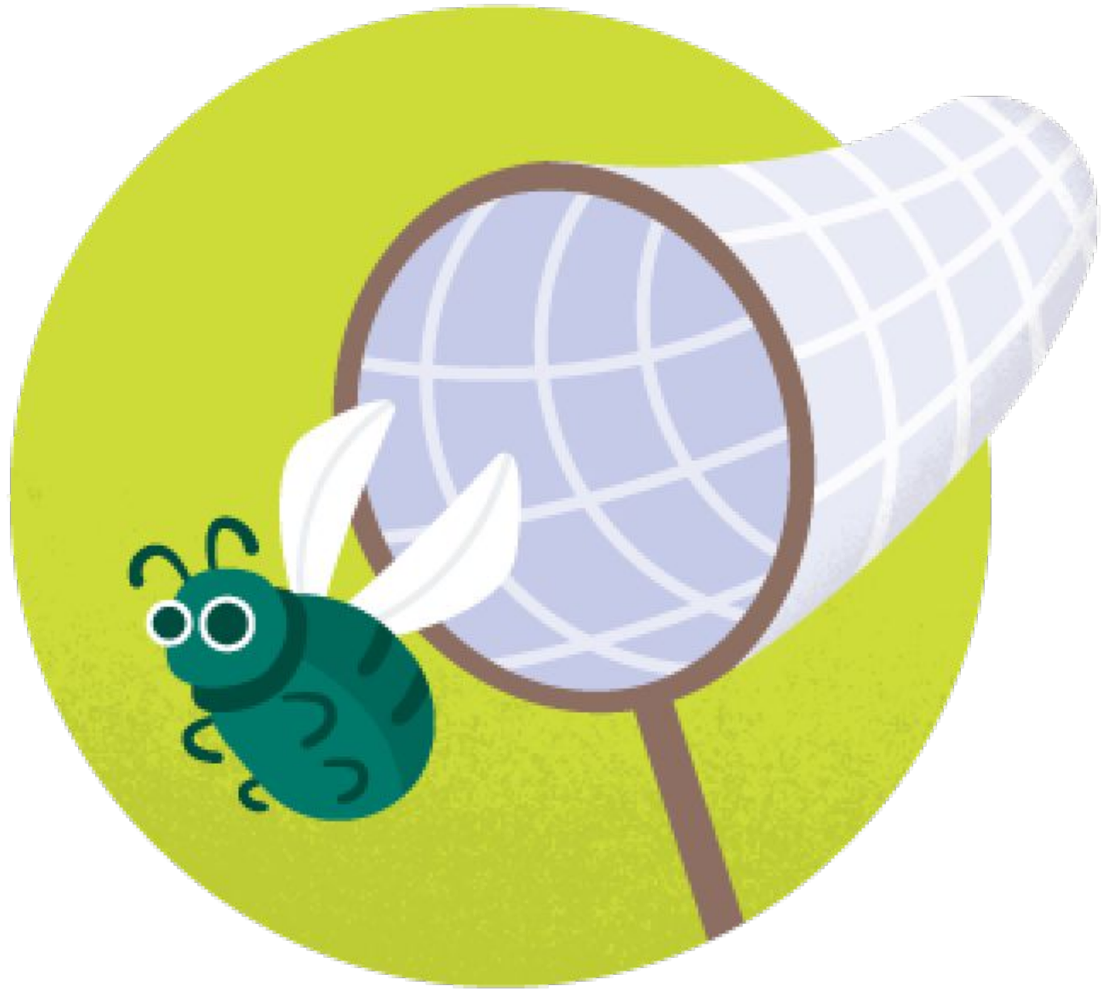
Crashlytics

Firebase Crashlytics

Crashlytics helps developers find and fix bugs quickly, so they can focus on making great apps.

Enables you to:

- ✓ **Monitor** app stability
- ✓ **Understand** errors
- ✓ **Prioritize** which errors to solve
- ✓ **Debug & Fix** errors



Common use cases

Get alerted when percentage of user sessions being affected by a crash crosses a threshold

Monitor **crash free users** as top level stability metric

Reproduce crashes faster by following user actions (using breadcrumbs)

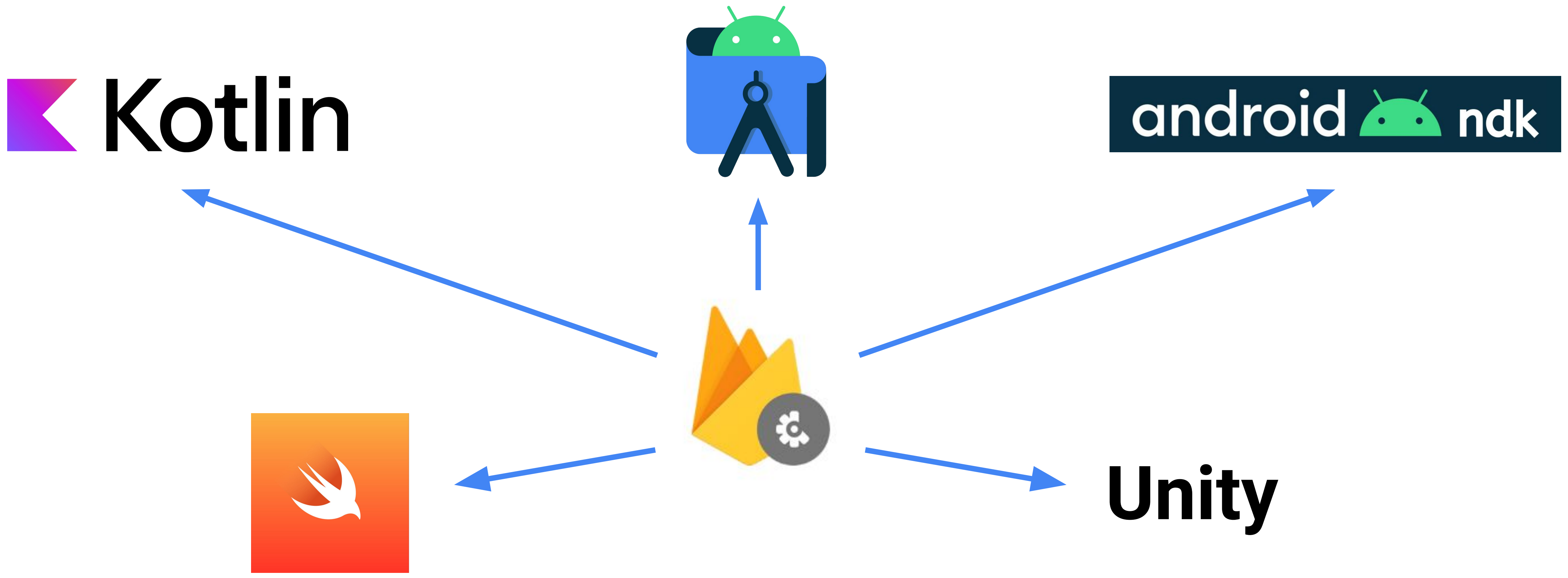
Find crashes in **common user workflows**

Monitor release stability and user adoption as **new versions rolls out**

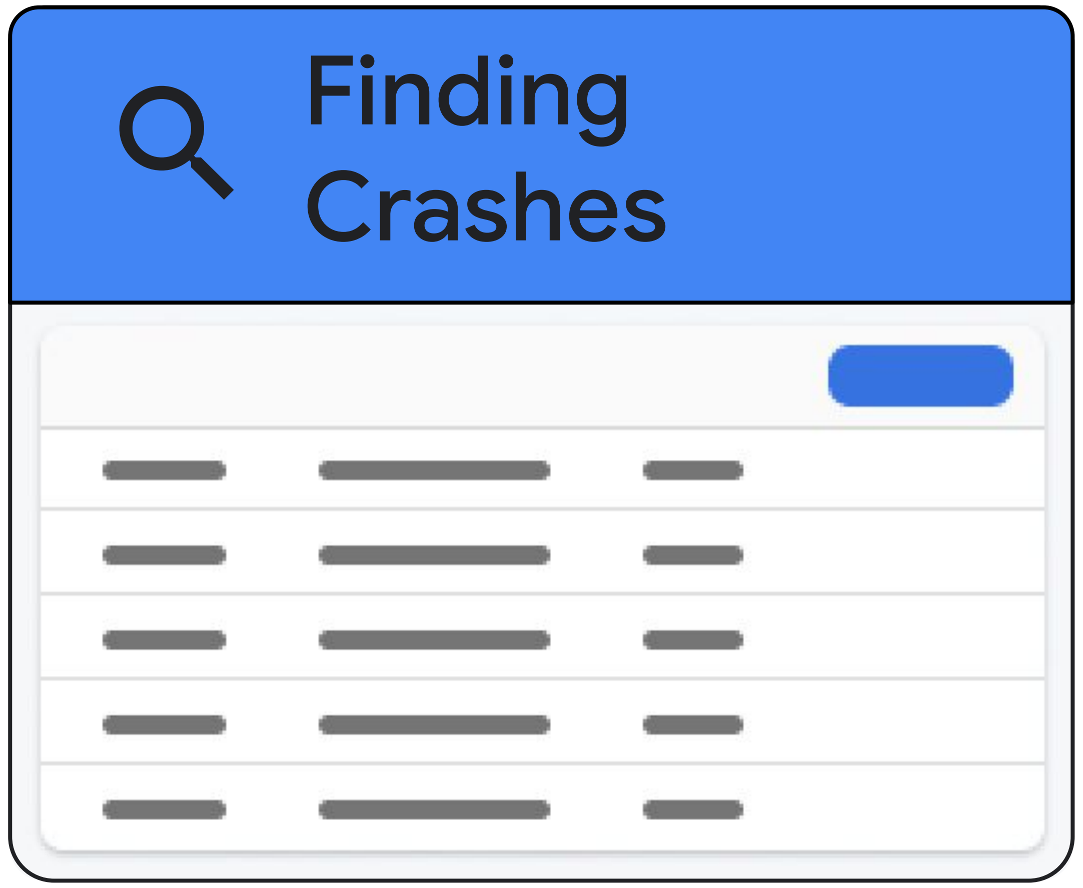
Find which crashes affected a **user or set of users**

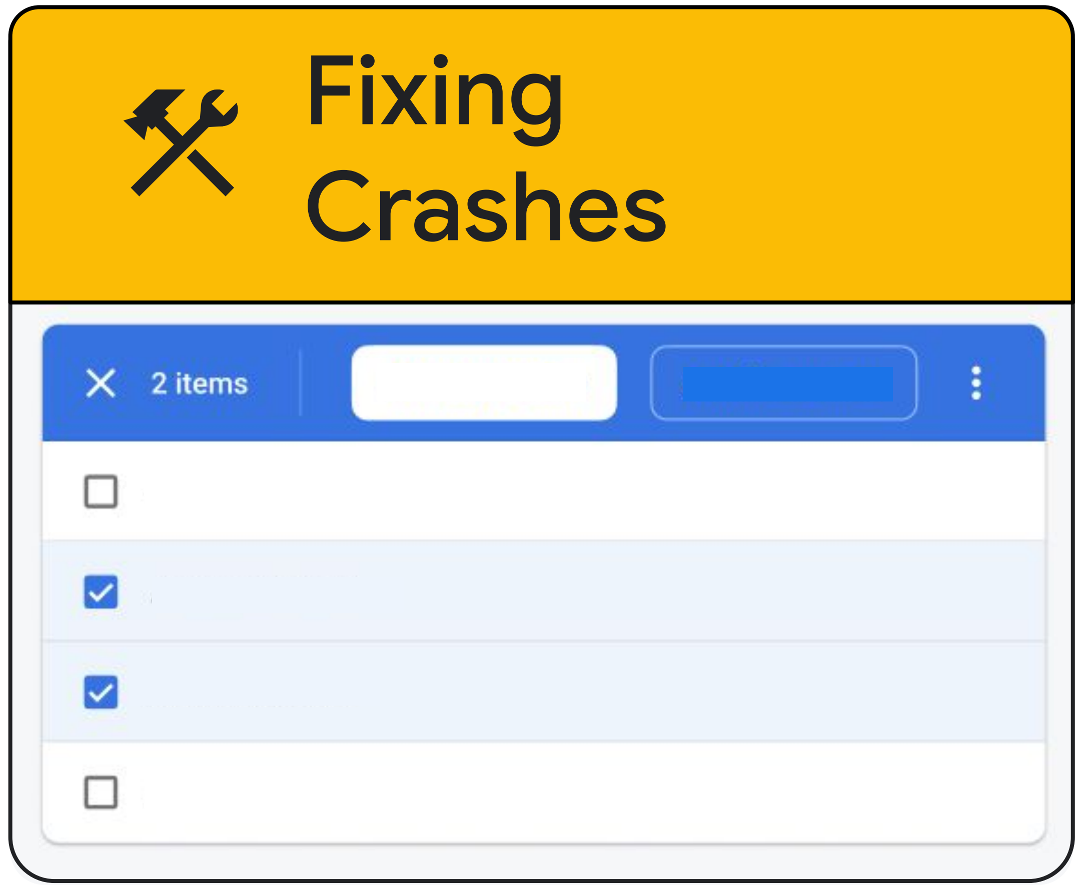
Crashlytics Support

Popular Google developer products



Open ecosystem of languages and frameworks


Finding Crashes


Fixing Crashes
2 items

Finding Crashes

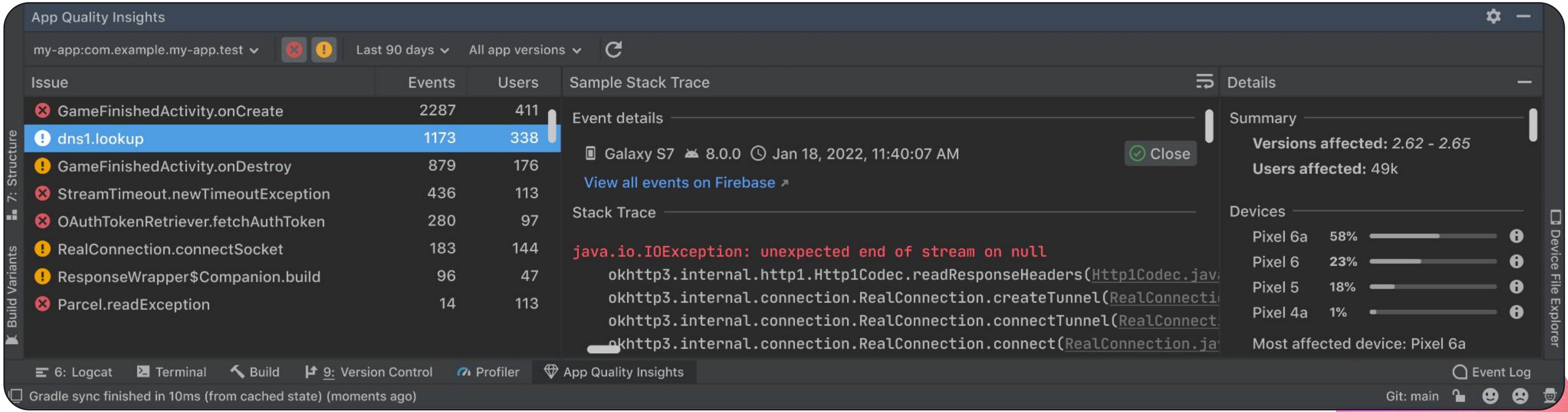
- Android Studio
- Memory bugs

Fixing Crashes

- Improved Stack traces
- Crash grouping

App Quality Insights

App Quality Insights



Issue	Events	Users
GameFinishedActivity.onCreate	2287	411
dns1.lookup	1173	338
GameFinishedActivity.onDestroy	879	176
StreamTimeout.newTimeoutException	436	113
OAuthTokenRetriever.fetchAuthToken	280	97
RealConnection.connectSocket	183	144
ResponseWrapper$Companion.build	96	47
Parcel.readException	14	113

Event details

Galaxy S7 8.0.0 Jan 18, 2022, 11:40:07 AM Close

View all events on Firebase ↗

Stack Trace

```
java.io.IOException: unexpected end of stream on null
    okhttp3.internal.http1.Http1Codec.readResponseHeaders(Http1Codec.java
    okhttp3.internal.connection.RealConnection.createTunnel(RealConnecti
    okhttp3.internal.connection.RealConnection.connectTunnel(RealConnect
    okhttp3.internal.connection.RealConnection.connect(RealConnection.ja
```

Summary

Versions affected: *2.62 - 2.65*

Users affected: 49k

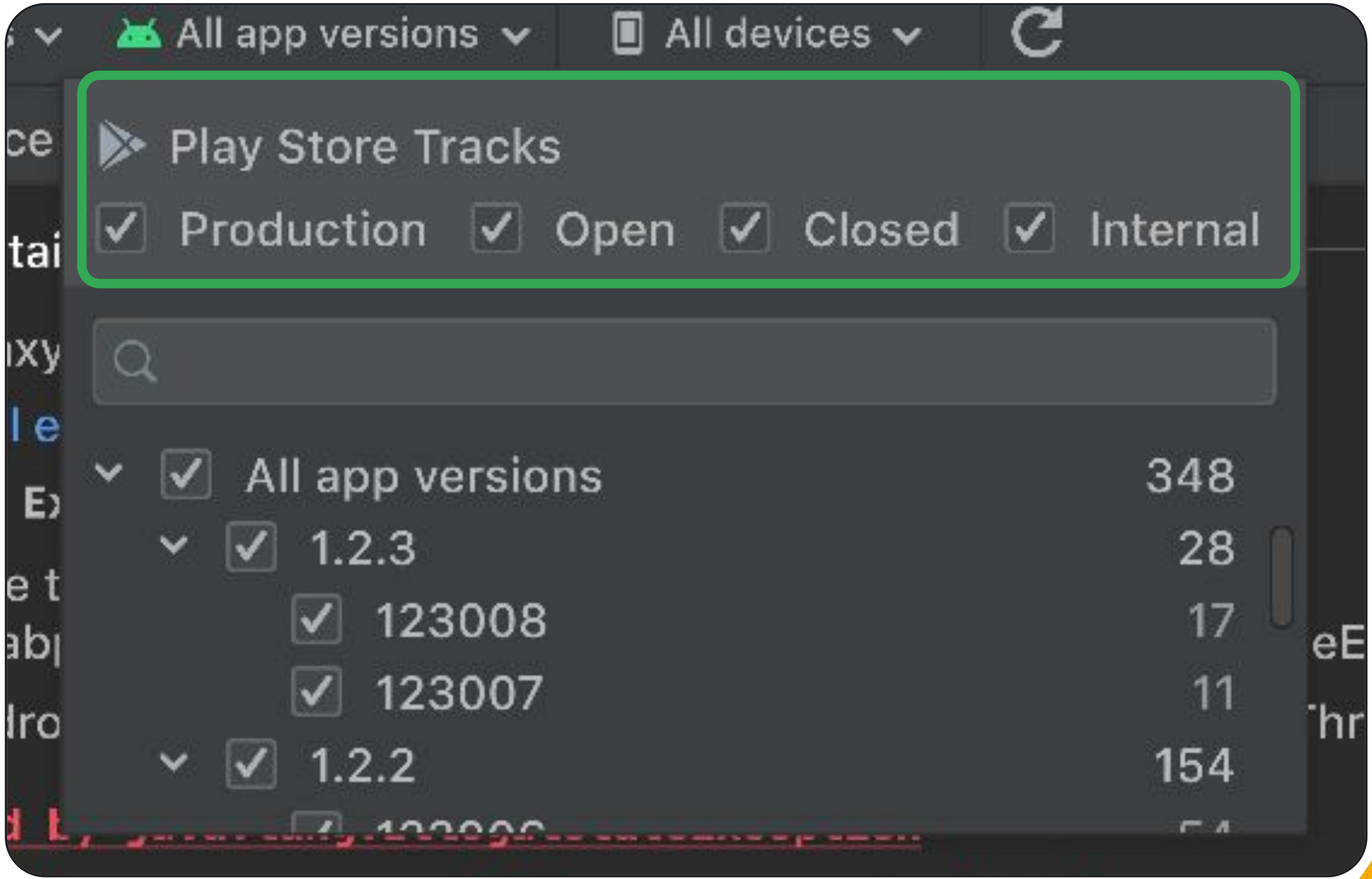
Devices

Pixel 6a	58%
Pixel 6	23%
Pixel 5	18%
Pixel 4a	1%

Most affected device: Pixel 6a

Gradle sync finished in 10ms (from cached state) (moments ago)

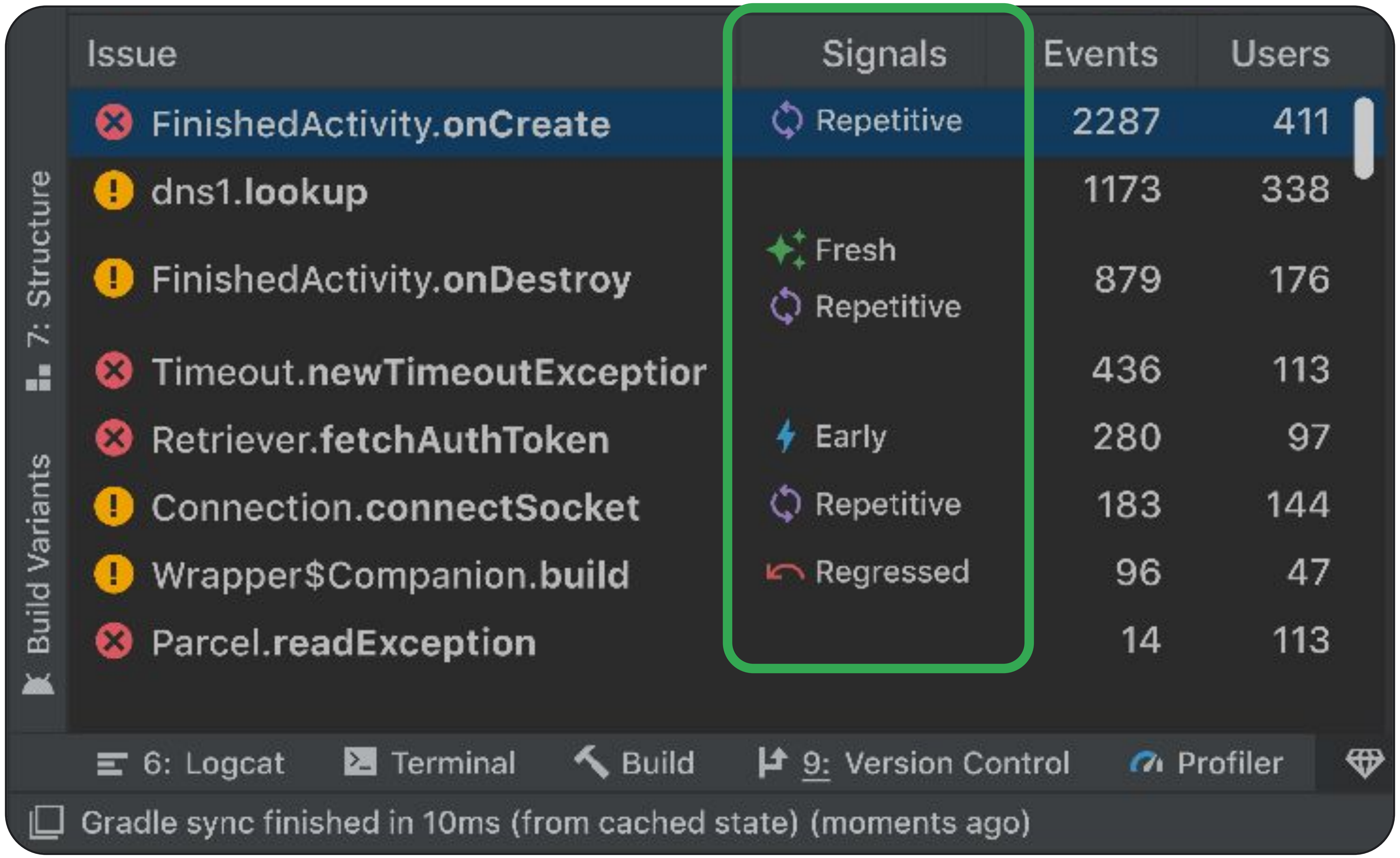
Play Tracks Filtering



✓ Production ✓ Open ✓ Closed ✓ Internal

✓ All app versions	348
✓ 1.2.3	28
✓ 123008	17
✓ 123007	11
✓ 1.2.2	154

Signals Filtering

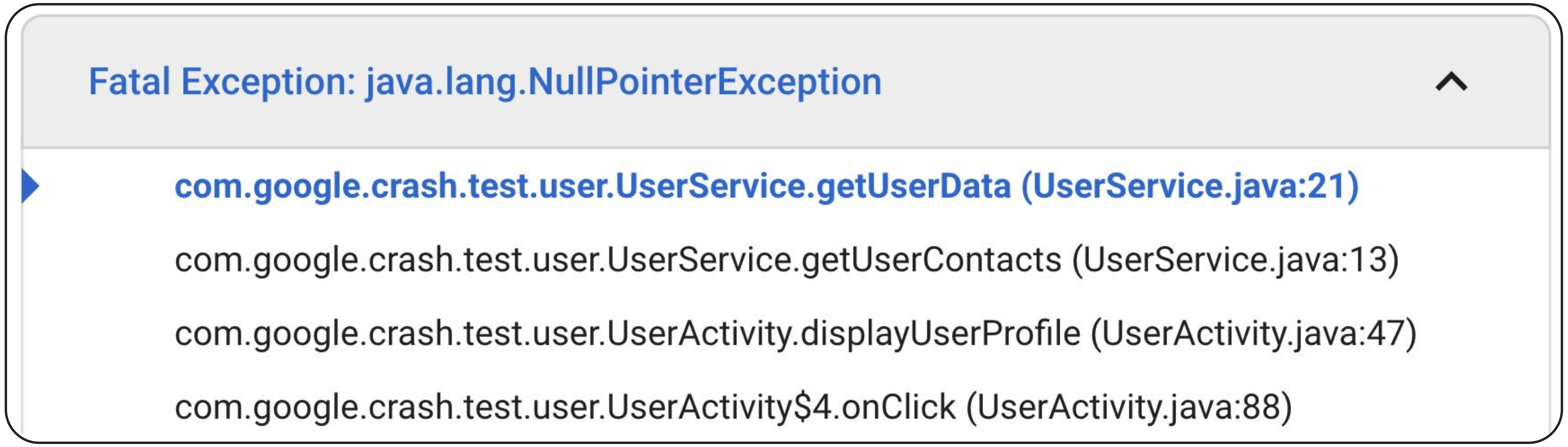


Issue	Signals	Events	Users
⊗ FinishedActivity.onCreate	Repetitive	2287	411
! dns1.lookup		1173	338
! FinishedActivity.onDestroy	Fresh Repetitive	879	176
⊗ Timeout.newTimeoutExceptior		436	113
⊗ Retriever.fetchAuthToken	Early	280	97
! Connection.connectSocket	Repetitive	183	144
! Wrapper$Companion.build	Regressed	96	47
⊗ Parcel.readException		14	113

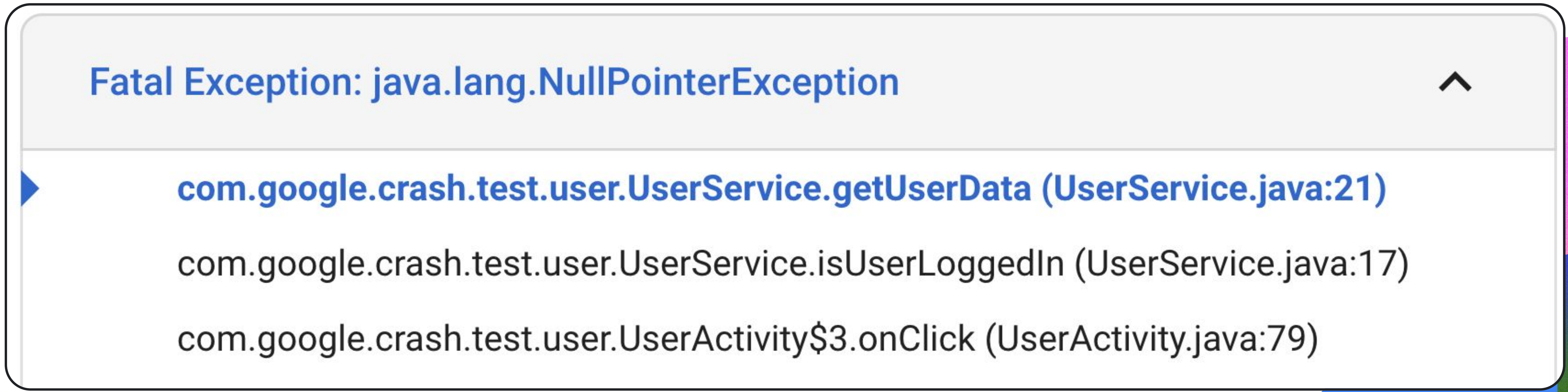
Gradle sync finished in 10ms (from cached state) (moments ago)

Crash Grouping

Similar stack traces

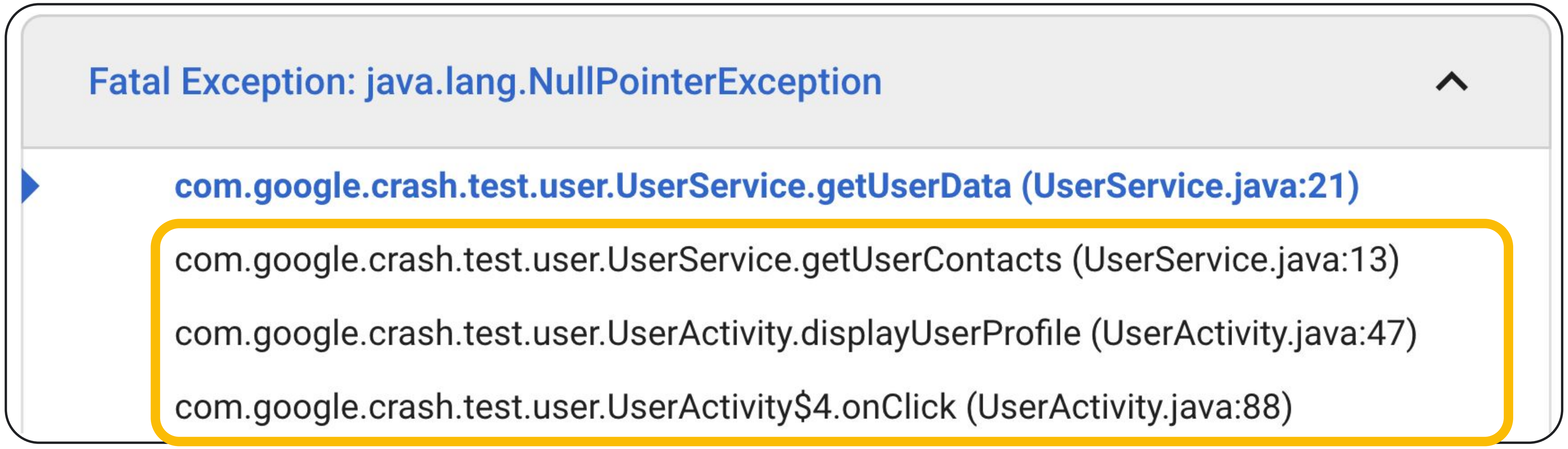


```
Fatal Exception: java.lang.NullPointerException
com.google.crash.test.user.UserService.getUserData (UserService.java:21)
com.google.crash.test.user.UserService.getUserContacts (UserService.java:13)
com.google.crash.test.user.UserActivity.displayUserProfile (UserActivity.java:47)
com.google.crash.test.user.UserActivity$4.onClick (UserActivity.java:88)
```



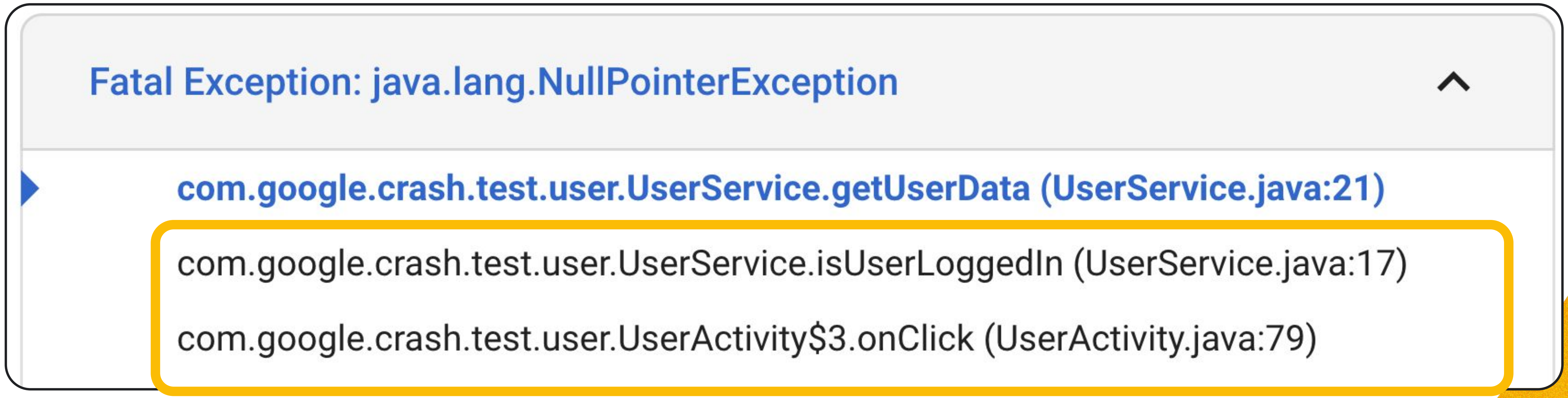
```
Fatal Exception: java.lang.NullPointerException
com.google.crash.test.user.UserService.getUserData (UserService.java:21)
com.google.crash.test.user.UserService.isUserLoggedIn (UserService.java:17)
com.google.crash.test.user.UserActivity$3.onClick (UserActivity.java:79)
```

Different code paths



Fatal Exception: java.lang.NullPointerException

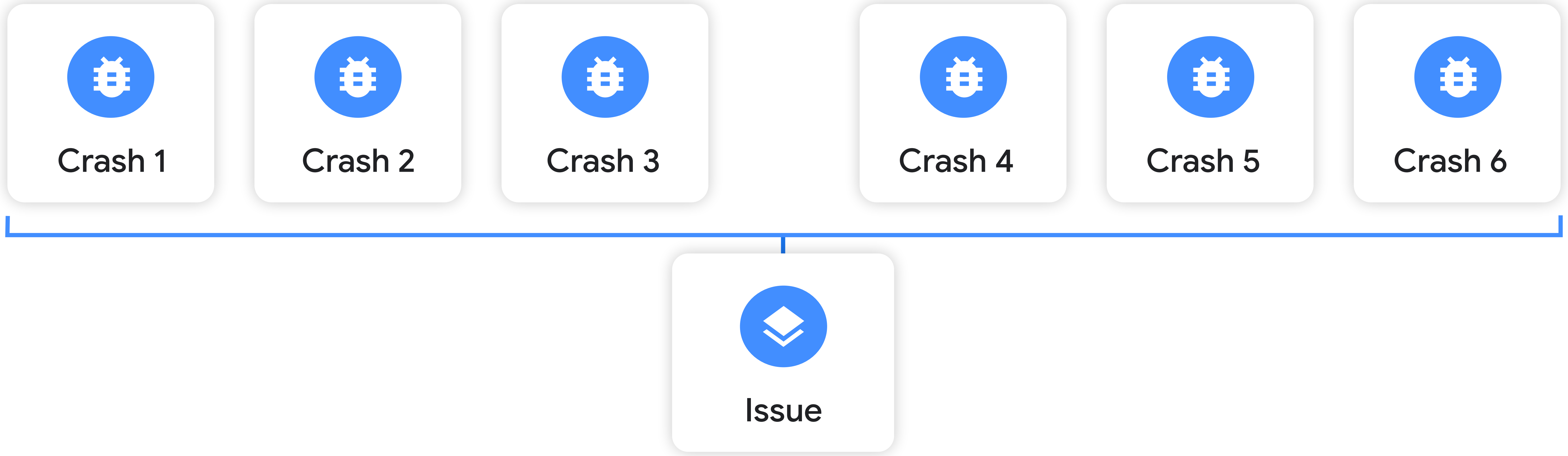
```
com.google.crash.test.user.UserService.getUserData (UserService.java:21)
com.google.crash.test.user.UserService.getUserContacts (UserService.java:13)
com.google.crash.test.user.UserActivity.displayUserProfile (UserActivity.java:47)
com.google.crash.test.user.UserActivity$4.onClick (UserActivity.java:88)
```



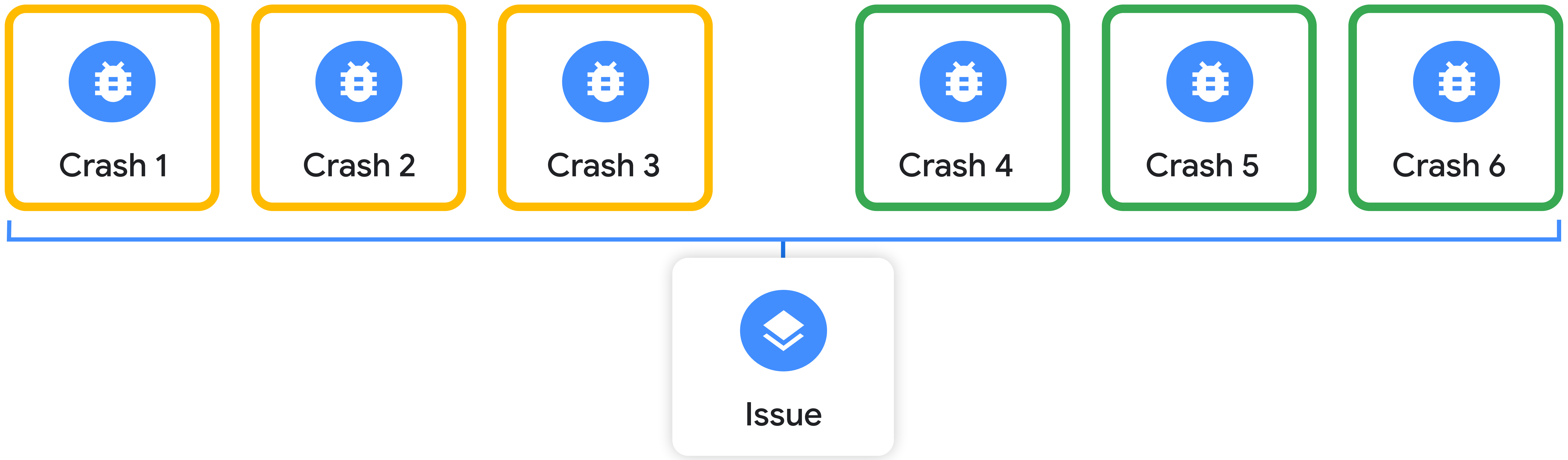
Fatal Exception: java.lang.NullPointerException

```
com.google.crash.test.user.UserService.getUserData (UserService.java:21)
com.google.crash.test.user.UserService.isUserLoggedIn (UserService.java:17)
com.google.crash.test.user.UserActivity$3.onClick (UserActivity.java:79)
```

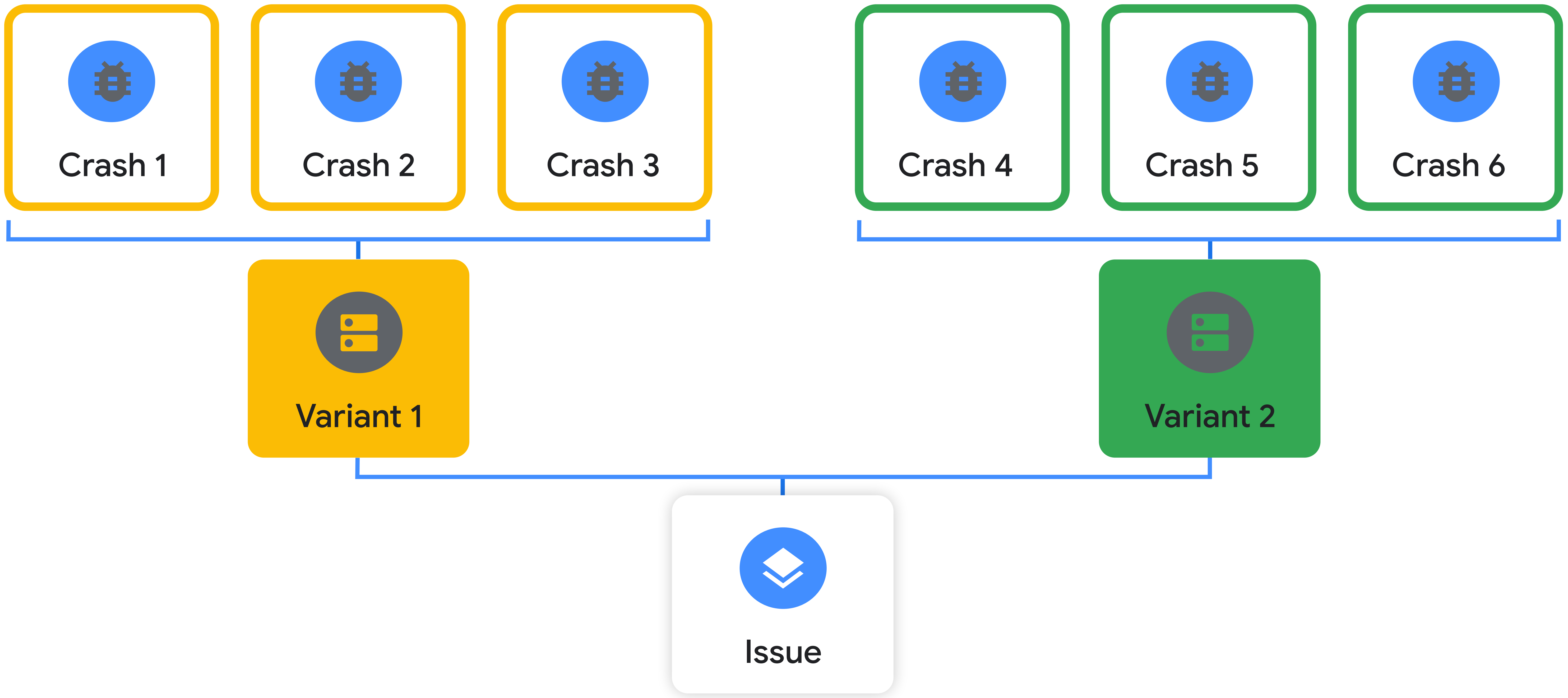
Crash grouping



Code Paths



Variants


Crash 1
Crash 2
Crash 3
Crash 4
Crash 5
Crash 6
Variant 1
Variant 2
Issue

Improved grouping algorithm

Fewer Duplicates

Line number change doesn't cause a new issue

Meaningful alerts and signals

New issue actually represents a new bug

More powerful search

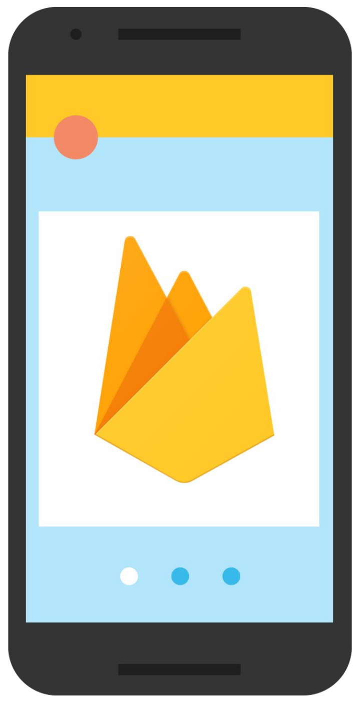
Each issue contains more searchable metadata, like exception type and package name

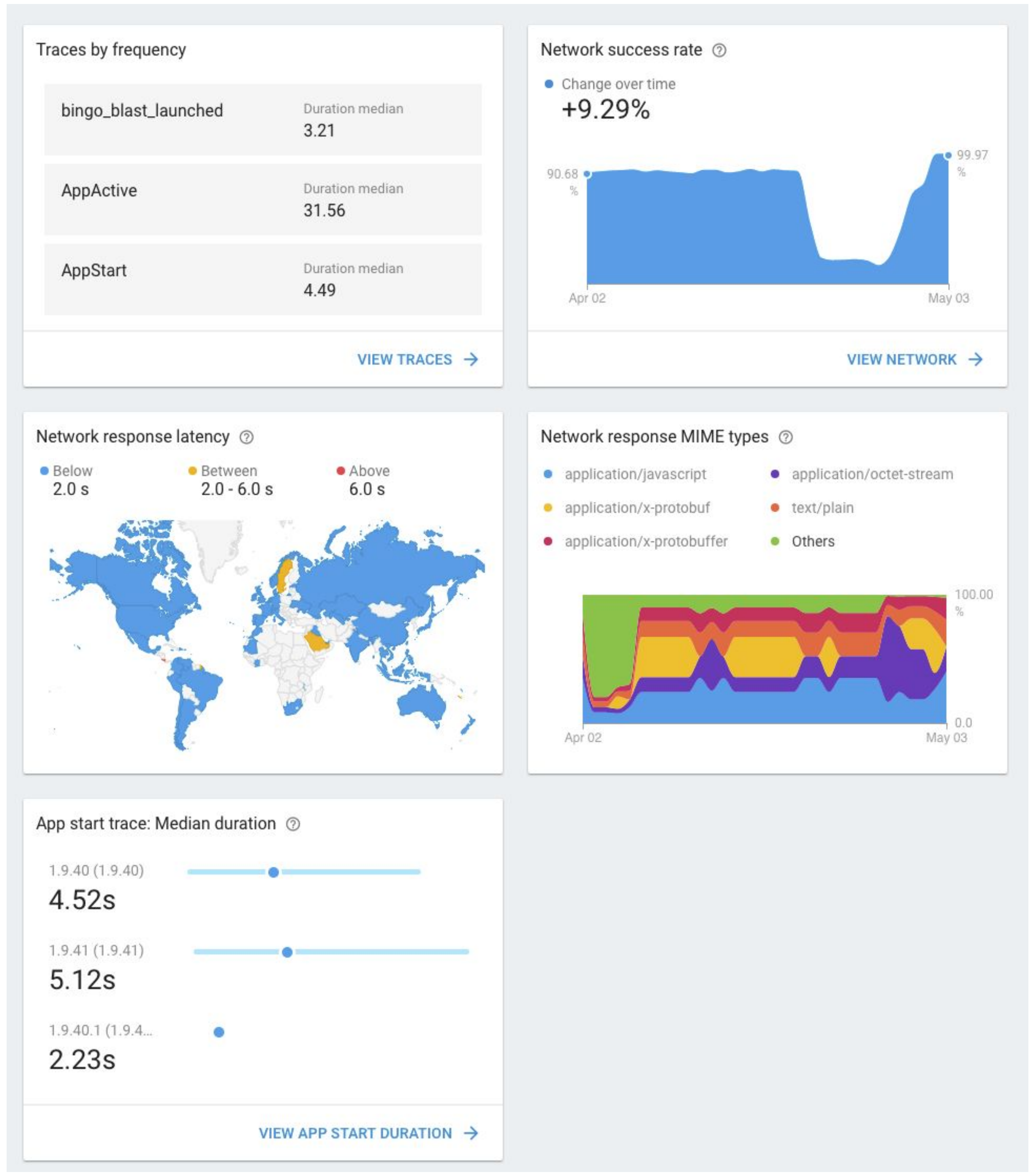
Performance Monitoring

Performance Monitoring

Performance Monitoring helps developers monitor the real-time performance of their applications

- App performance monitoring
- App start and network latencies
- Custom traces





Traces by frequency

bingo_blast_launched	Duration median 3.21
AppActive	Duration median 31.56
AppStart	Duration median 4.49

VIEW TRACES →

Network success rate

● Change over time
+9.29%

VIEW NETWORK →

Network response latency

● Below 2.0 s
● Between 2.0 - 6.0 s
● Above 6.0 s

Network response MIME types

● application/javascript
● application/octet-stream
● application/x-protobuf
● text/plain
● application/x-protobuffer
● Others

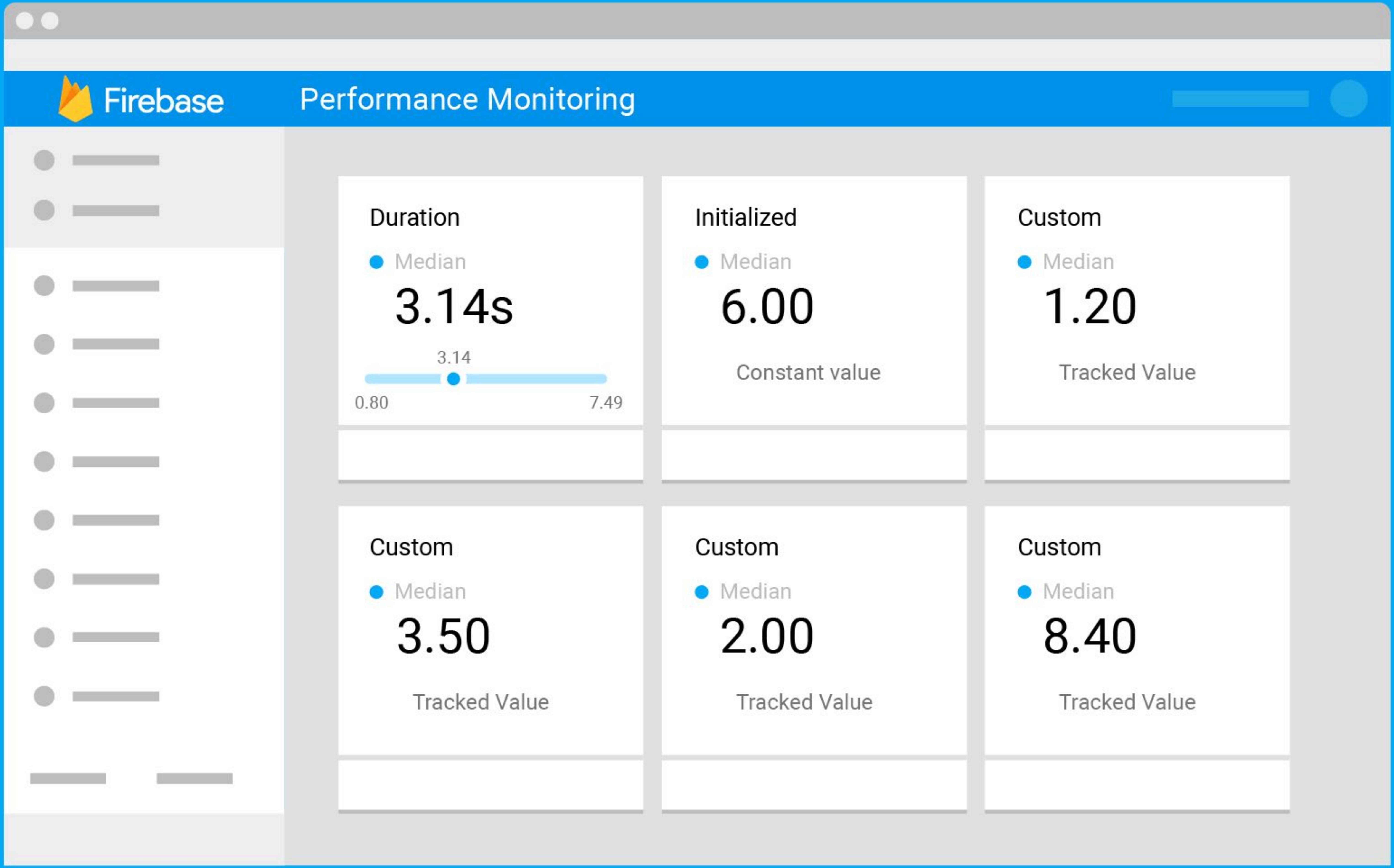
App start trace: Median duration

1.9.40 (1.9.40)
4.52s

1.9.41 (1.9.41)
5.12s

1.9.40.1 (1.9.4…
2.23s

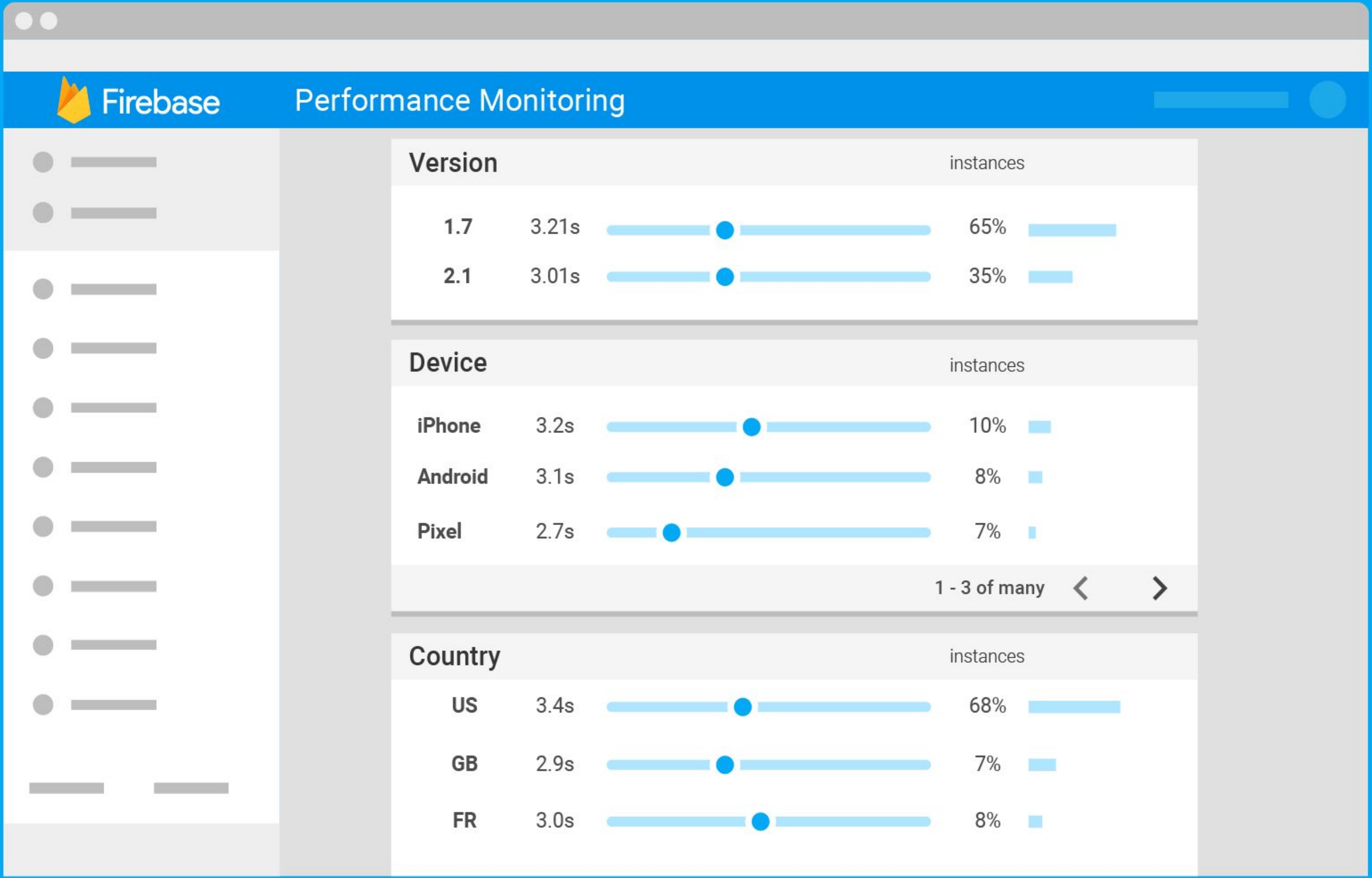
VIEW APP START DURATION →



Firebase Performance Monitoring

Duration	Initialized	Custom
Median	Median	Median
3.14s	6.00	1.20
3.14 (0.80 – 7.49)	Constant value	Tracked Value

Custom	Custom	Custom
Median	Median	Median
3.50	2.00	8.40
Tracked Value	Tracked Value	Tracked Value


Firebase Performance Monitoring

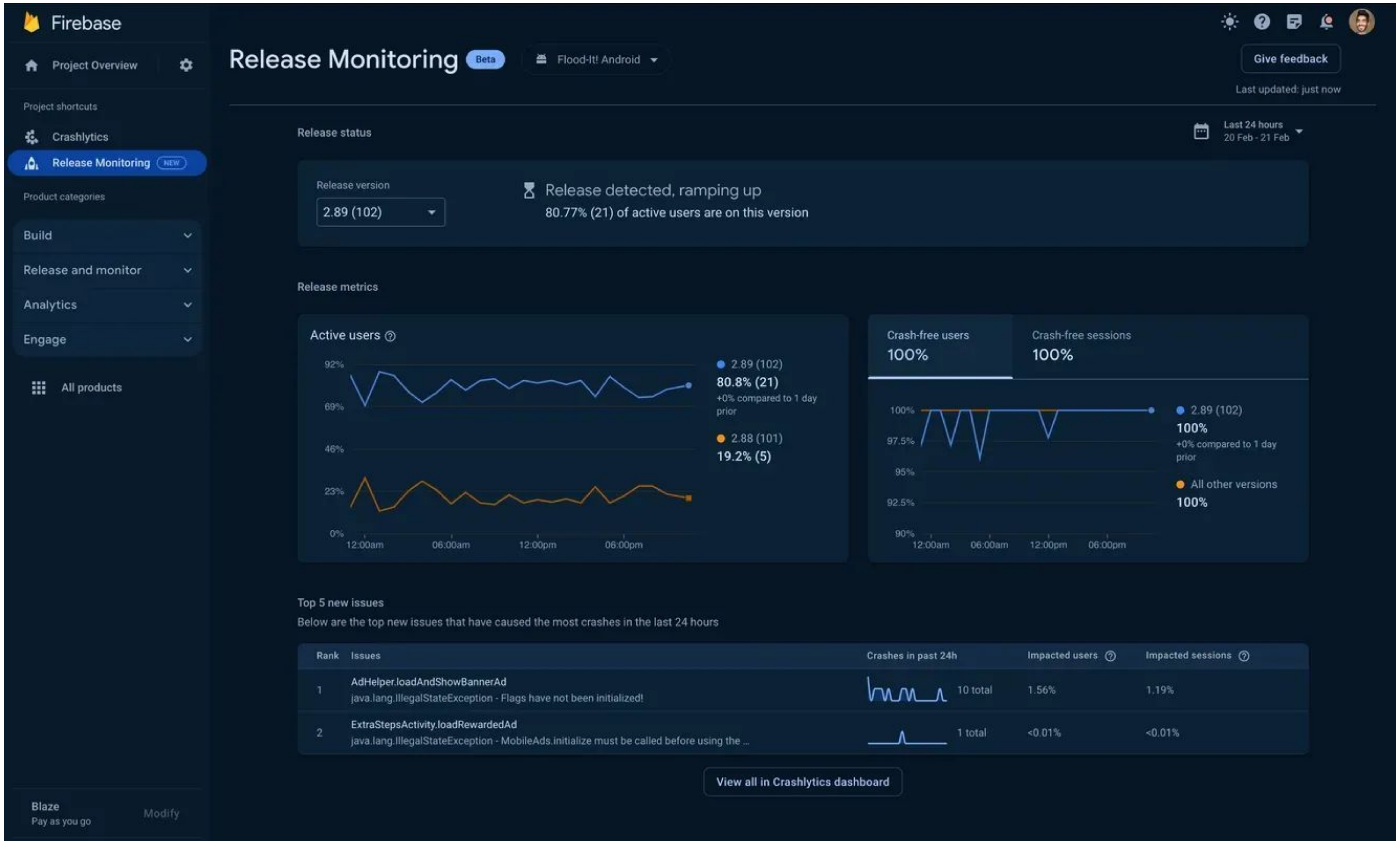
Version		instances
1.7	3.21s	65%
2.1	3.01s	35%

Device		instances
iPhone	3.2s	10%
Android	3.1s	8%
Pixel	2.7s	7%

1 - 3 of many

Country		instances
US	3.4s	68%
GB	2.9s	7%
FR	3.0s	8%

Release Monitoring



Release Monitoring Beta

Flood-It! Android

Give feedback

Last updated: just now

Release status

Last 24 hours 20 Feb - 21 Feb

Release version: 2.89 (102)

Release detected, ramping up
80.77% (21) of active users are on this version

Release metrics

Active users

● 2.89 (102)
80.8% (21)
+0% compared to 1 day prior

● 2.88 (101)
19.2% (5)

Crash-free users 100%

Crash-free sessions 100%

● 2.89 (102)
100%
+0% compared to 1 day prior

● All other versions
100%

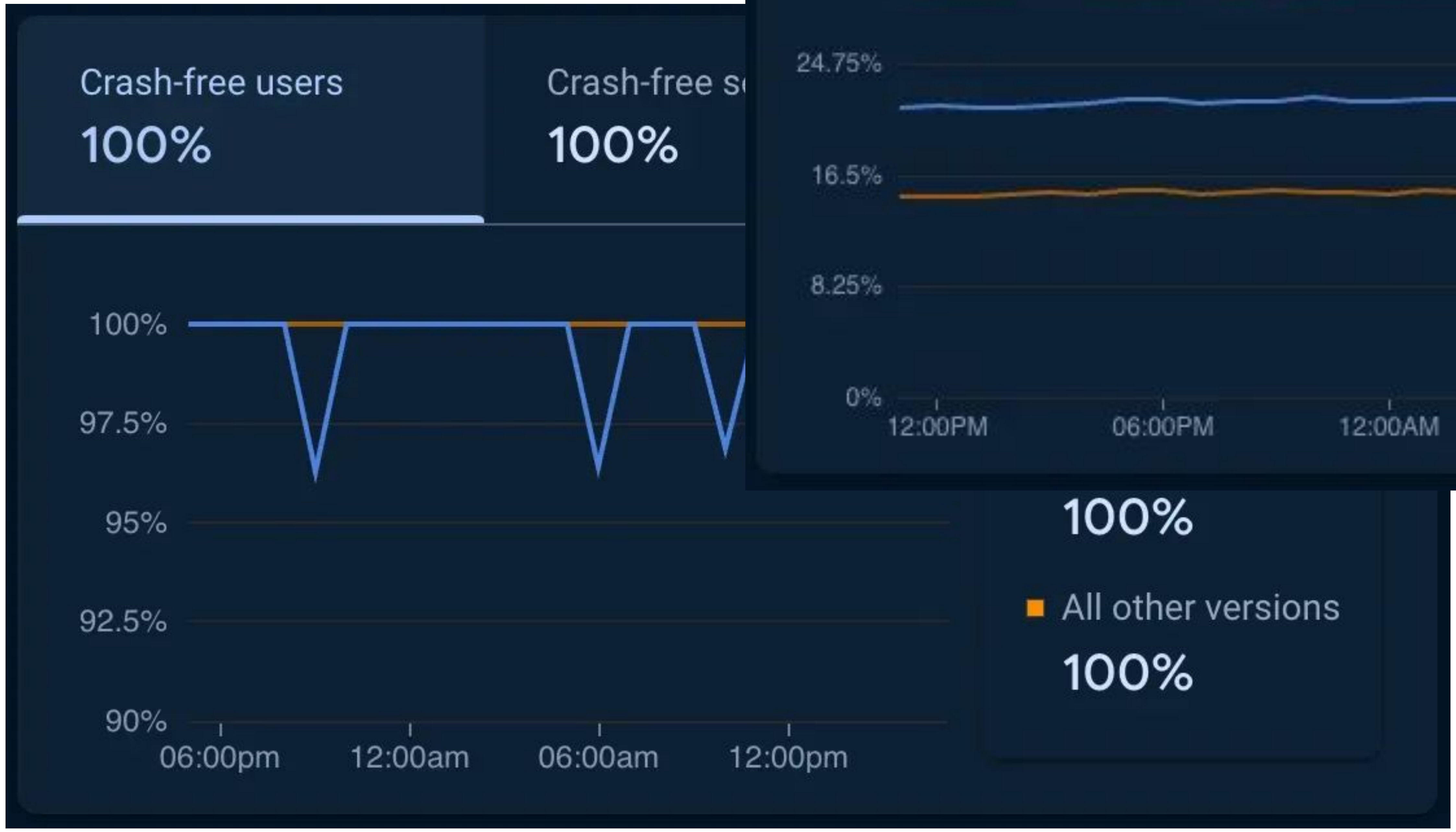
Top 5 new issues

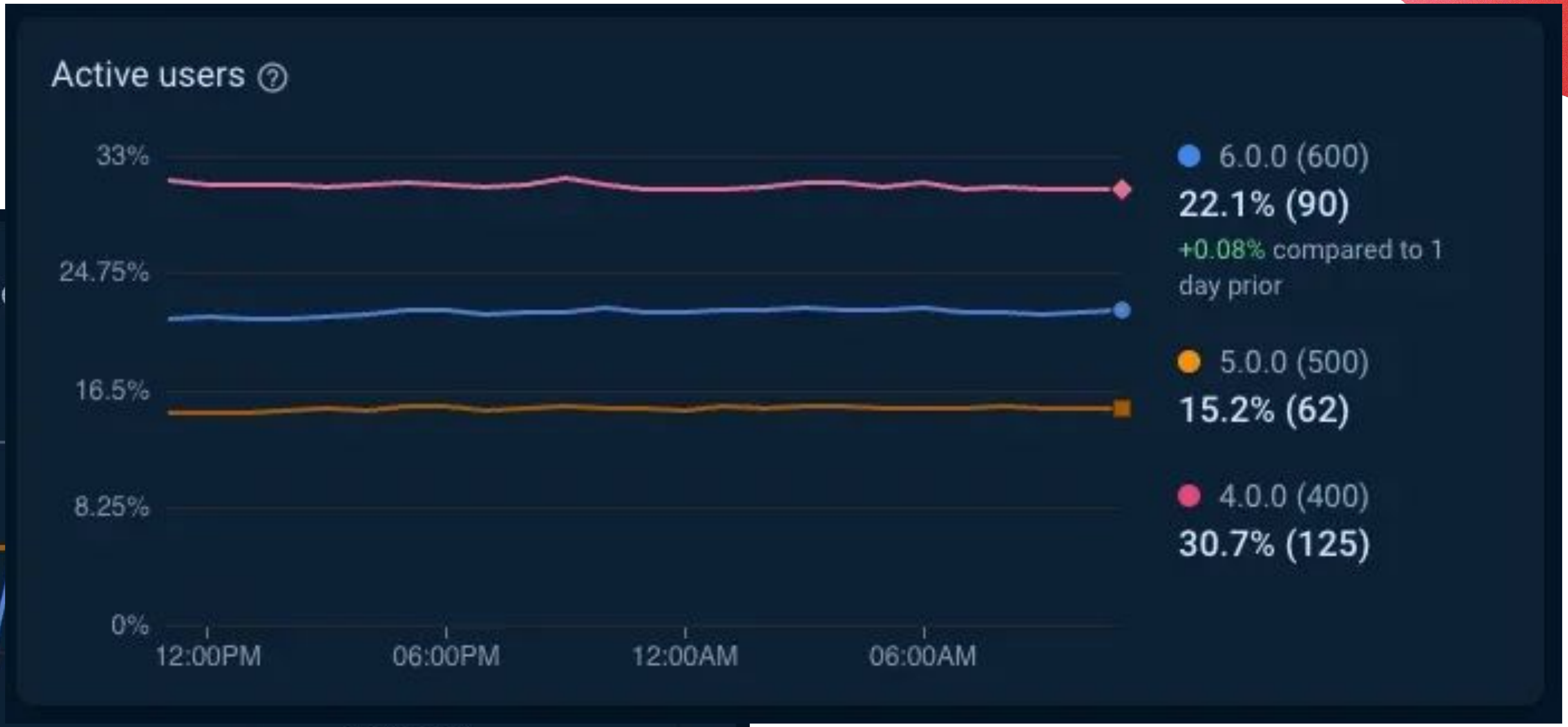
Below are the top new issues that have caused the most crashes in the last 24 hours

Rank	Issues	Crashes in past 24h	Impacted users	Impacted sessions
1	AdHelper.loadAndShowBannerAd java.lang.IllegalStateException - Flags have not been initialized!	10 total	1.56%	1.19%
2	ExtraStepsActivity.loadRewardedAd java.lang.IllegalStateException - MobileAds.initialize must be called before using the ...	1 total	<0.01%	<0.01%

View all in Crashlytics dashboard

Release Monitoring





Remote Config

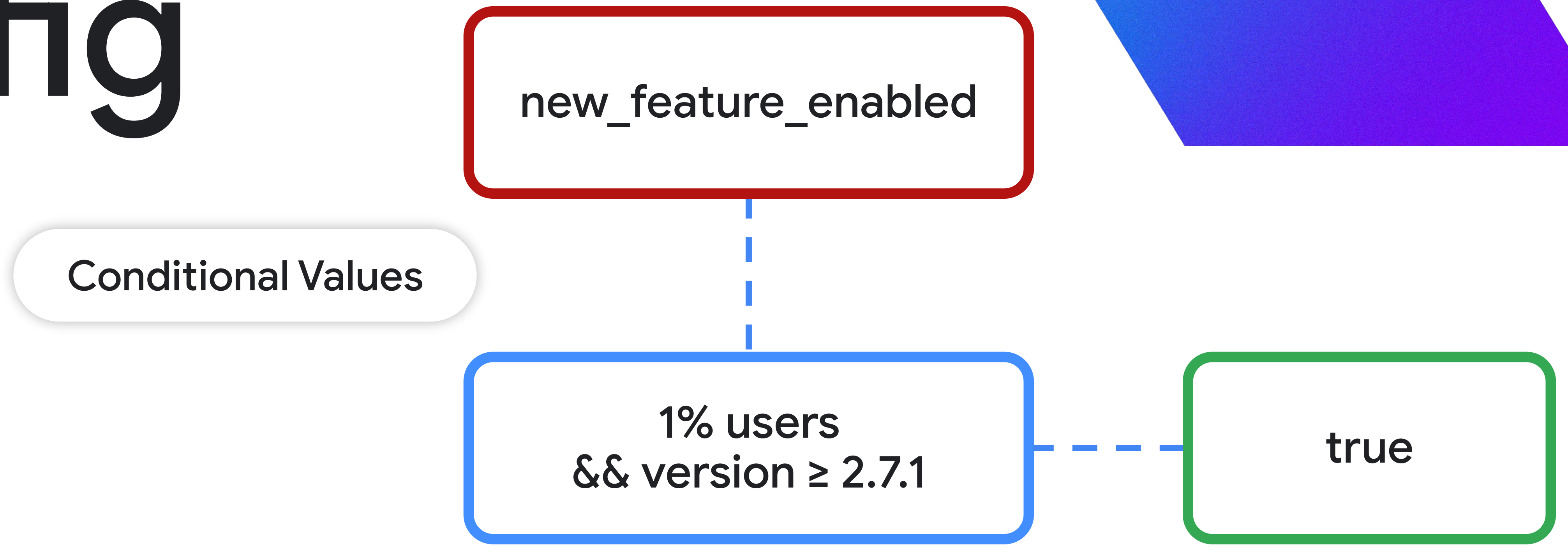
About Remote Config

Remote Config helps app developers safely roll out new features and optimize apps, by providing real-time visibility control over app configuration

Jobs to be done:

- **Configure** apps without performing a release
- **Target** user experiences
- **Manage Features** with feature flags and rollouts
- **Optimize** apps with A/B testing and Personalization

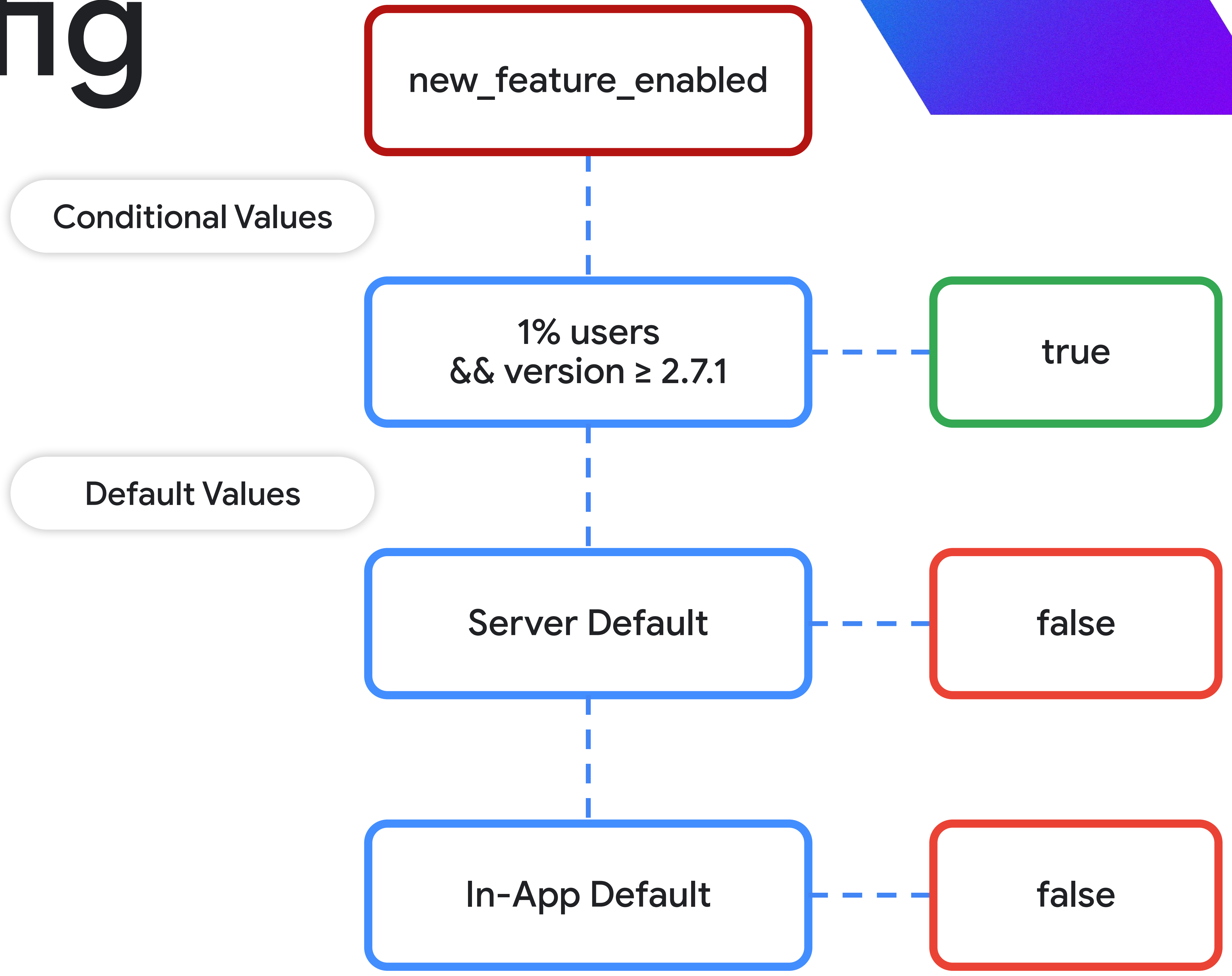
Remote Config



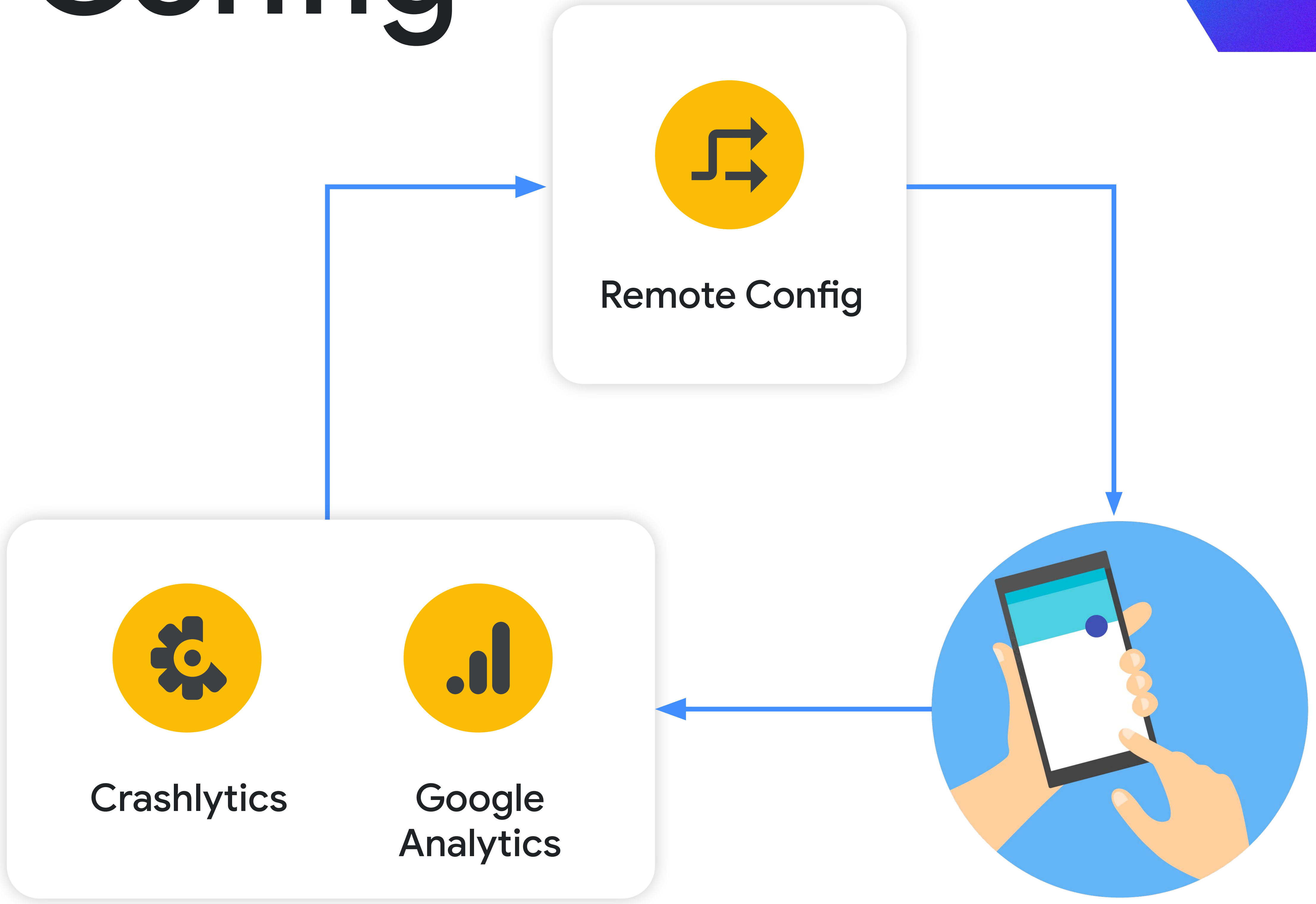
Create key-value pairs with user-targeted server overrides and in-app defaults.

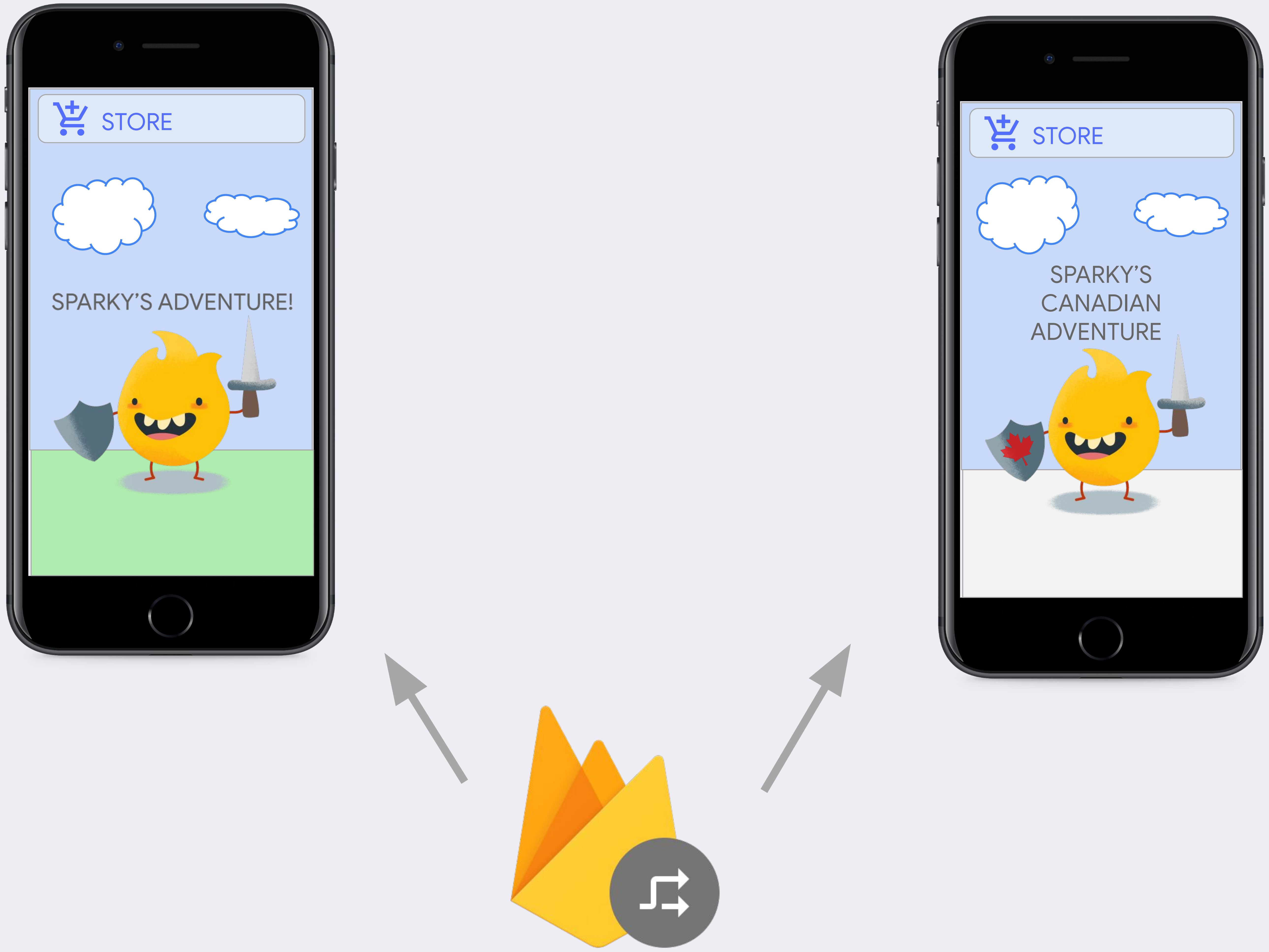
Remote Config

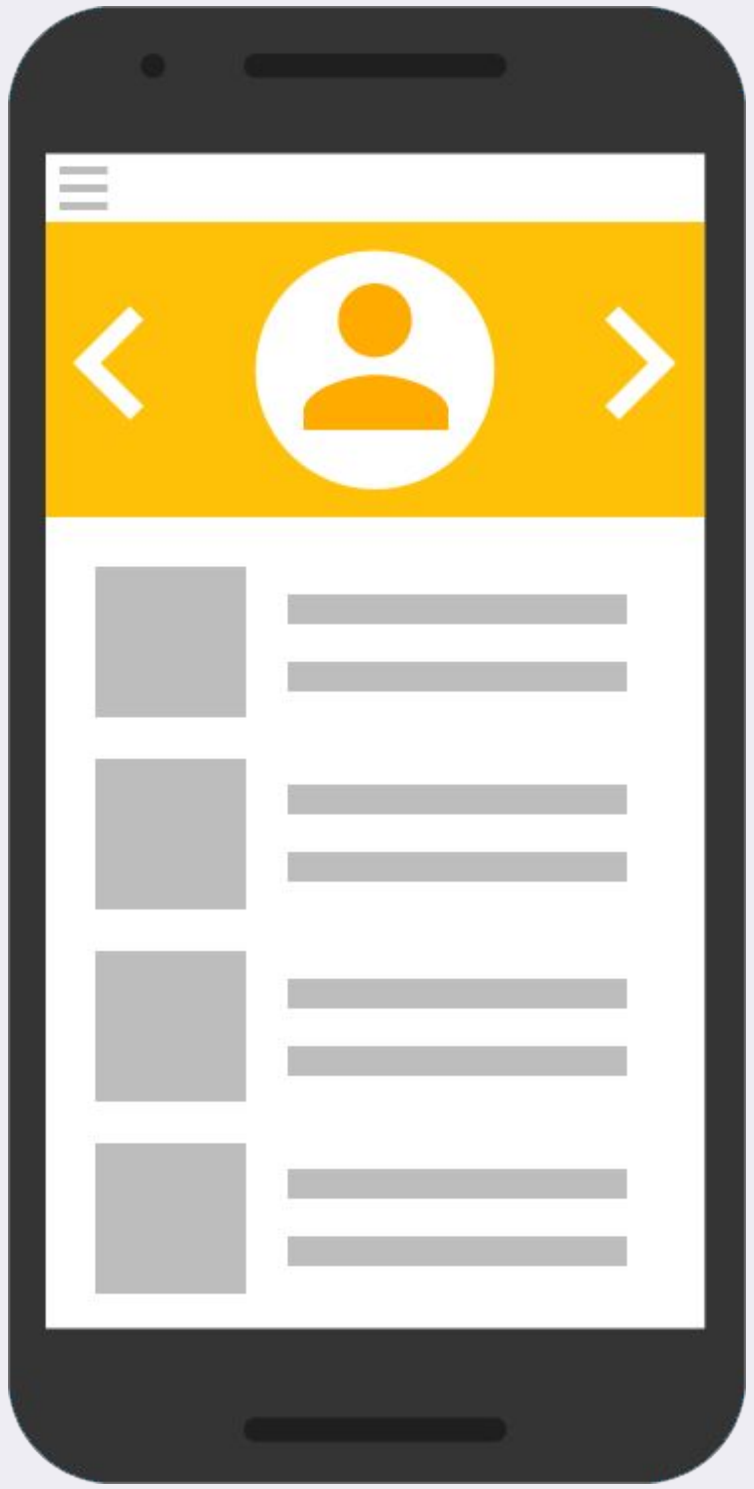
Create key-value pairs with user-targeted server overrides and in-app defaults.



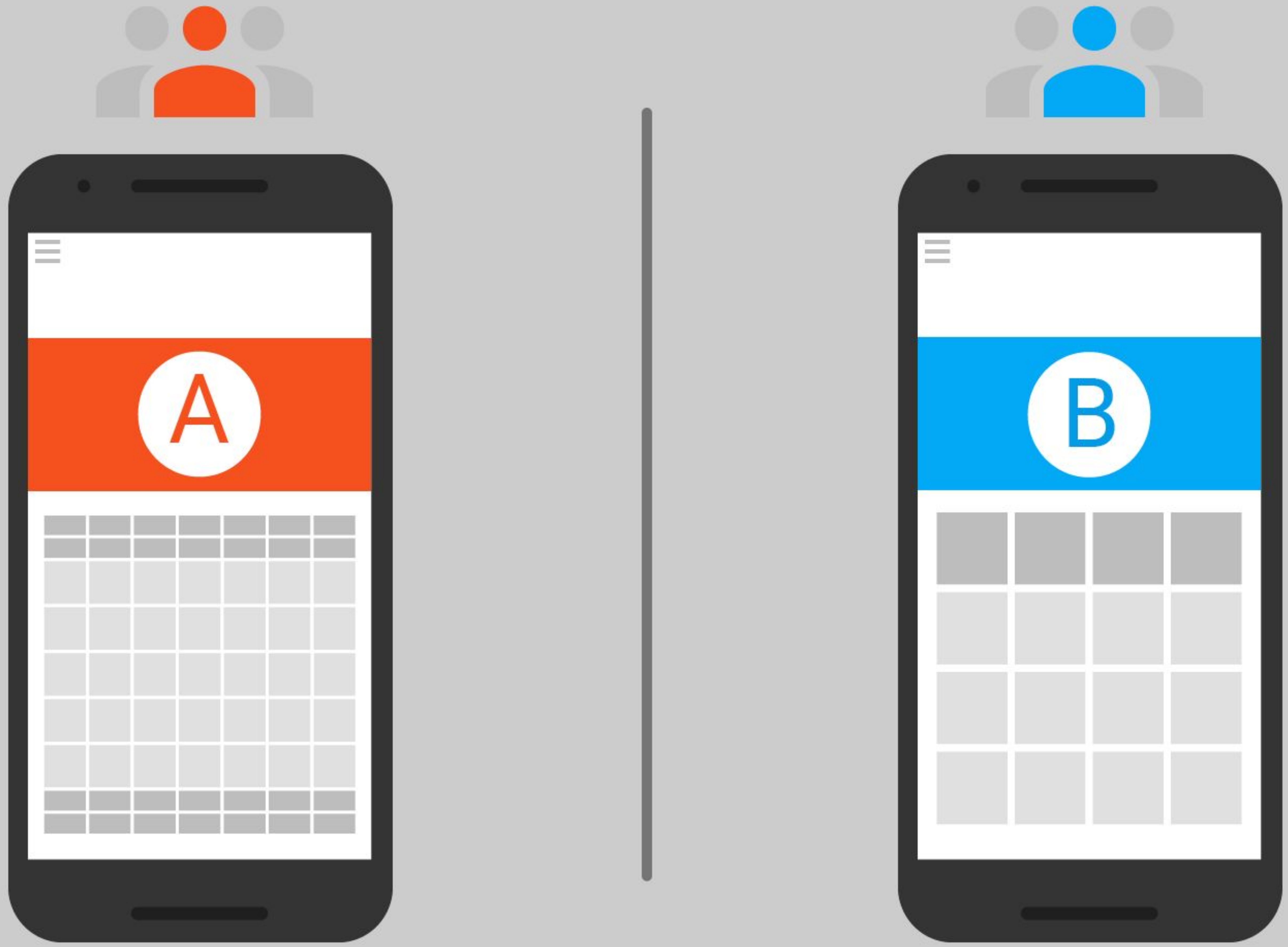
Remote Config




STORE
SPARKY'S ADVENTURE!
STORE
SPARKY'S CANADIAN ADVENTURE


is_spender = false

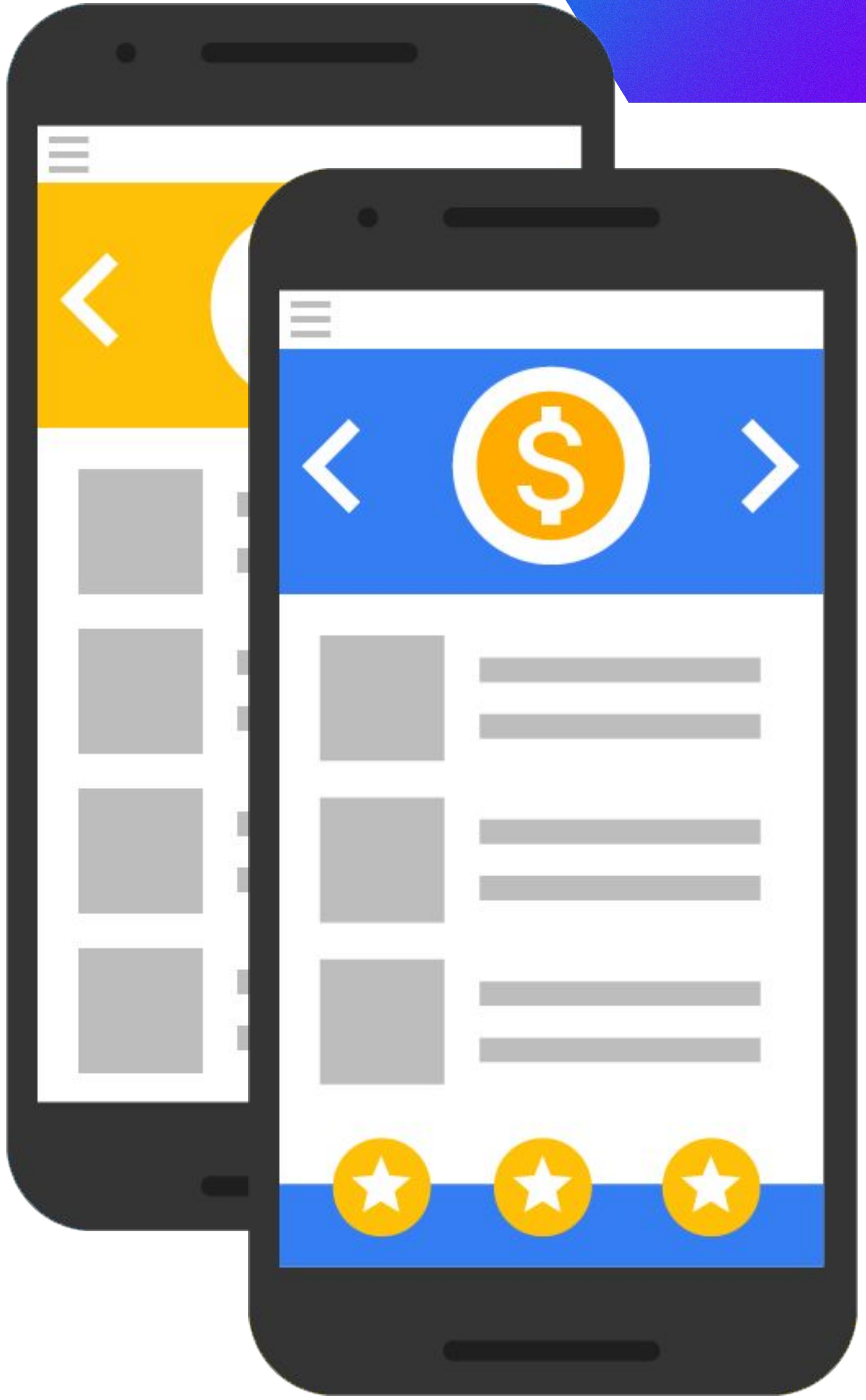
is_spender = true


A
B

Feature Flags

Feature Flagging

- Decouple feature launches from app version releases
- Gradually enable new functionality
- Roll back if there are issues



Feature Flagging

Reduce risk

Roll out gradually to catch production bugs before they affect most users

Feature Flagging

Reduce risk

Roll out gradually to catch production bugs before they affect most users

Launch confidently

Launch new features to a limited audience to gather feedback early

Feature Flagging

Reduce risk

Roll out gradually to catch production bugs before they affect most users

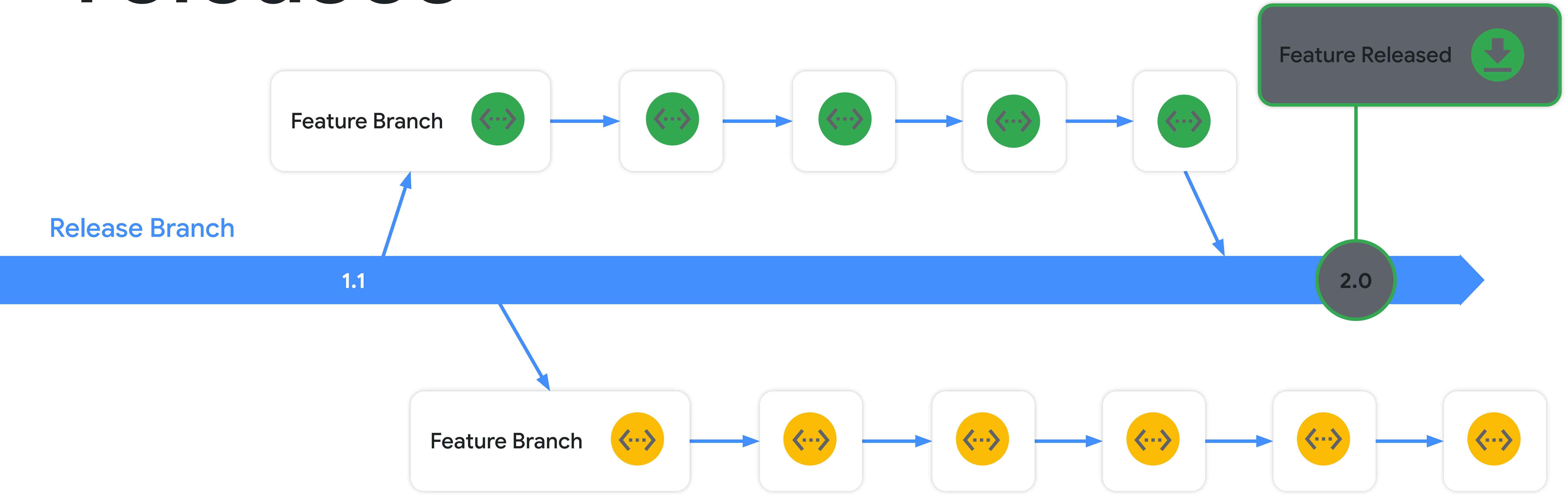
Launch confidently

Launch new features to a limited audience to gather feedback early

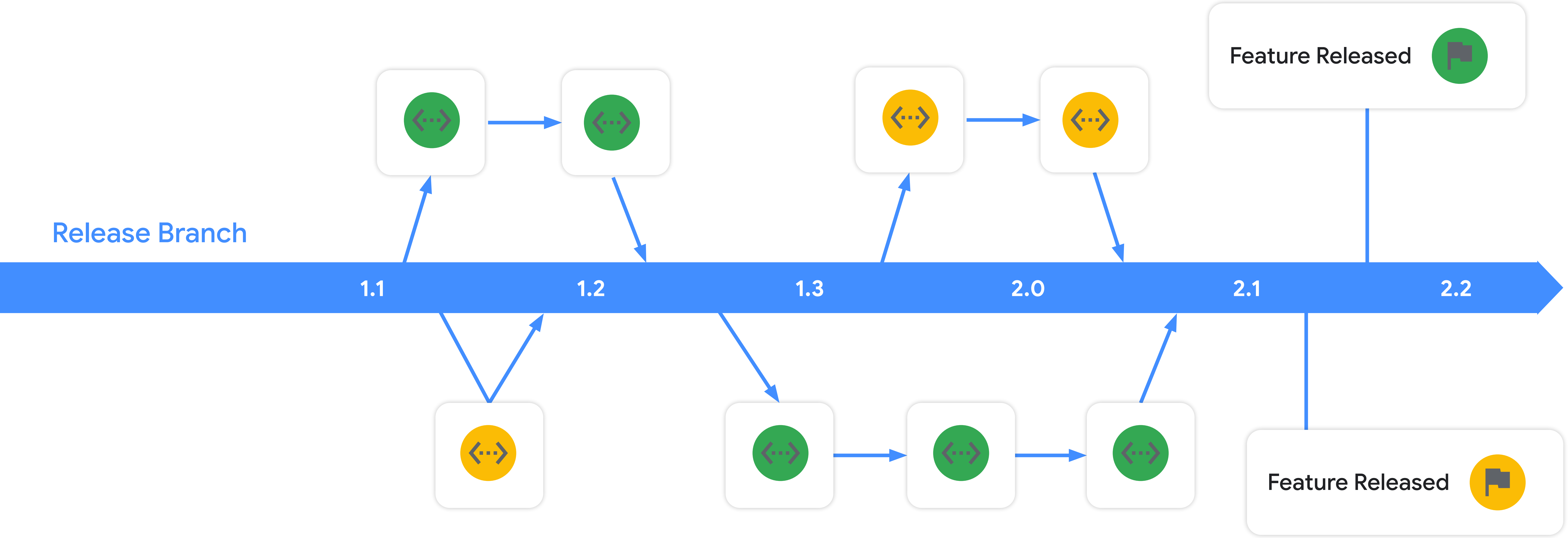
Ship faster

Develop incrementally without tying new features to new app versions

App version-based releases



Feature flagged releases

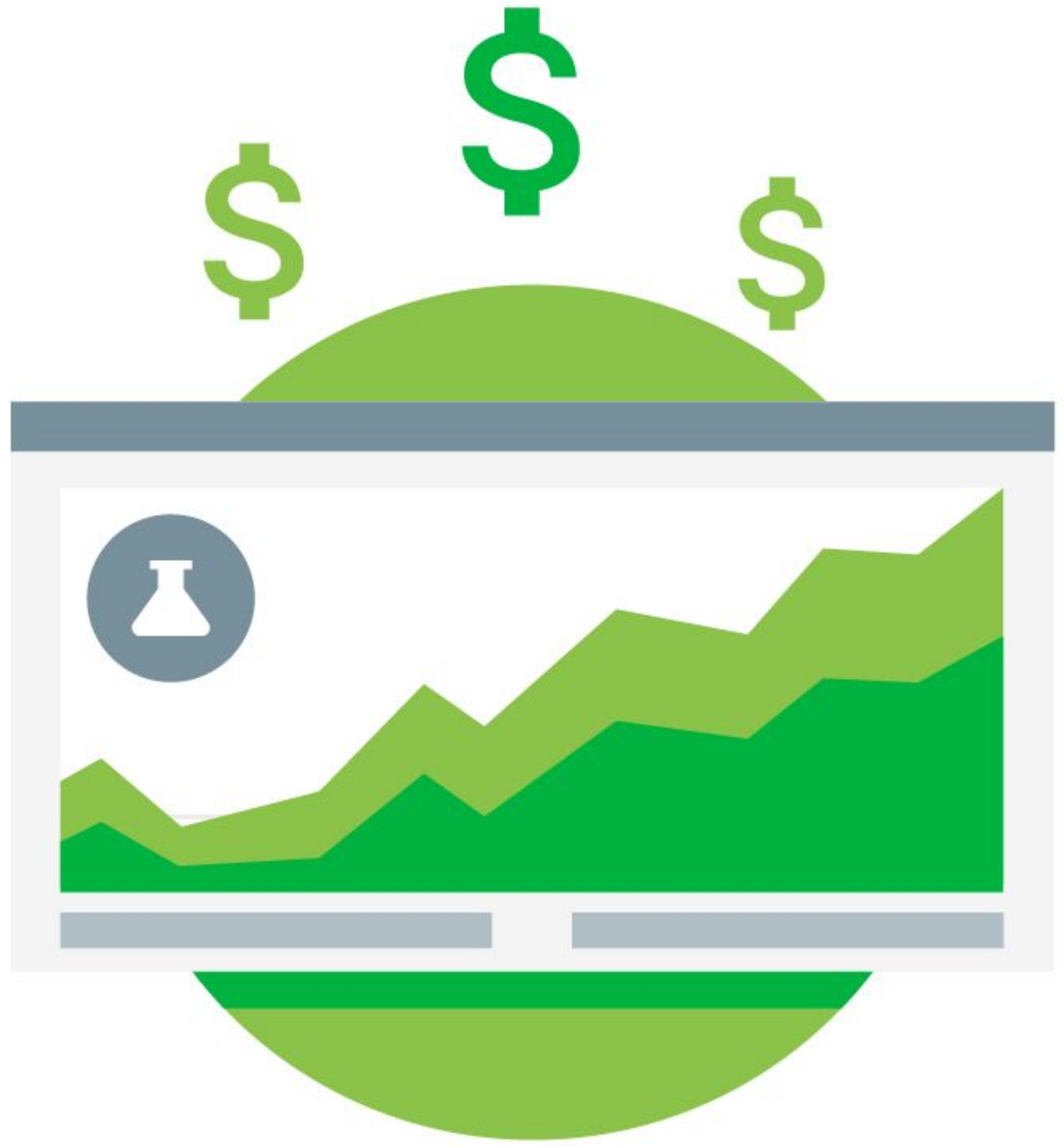


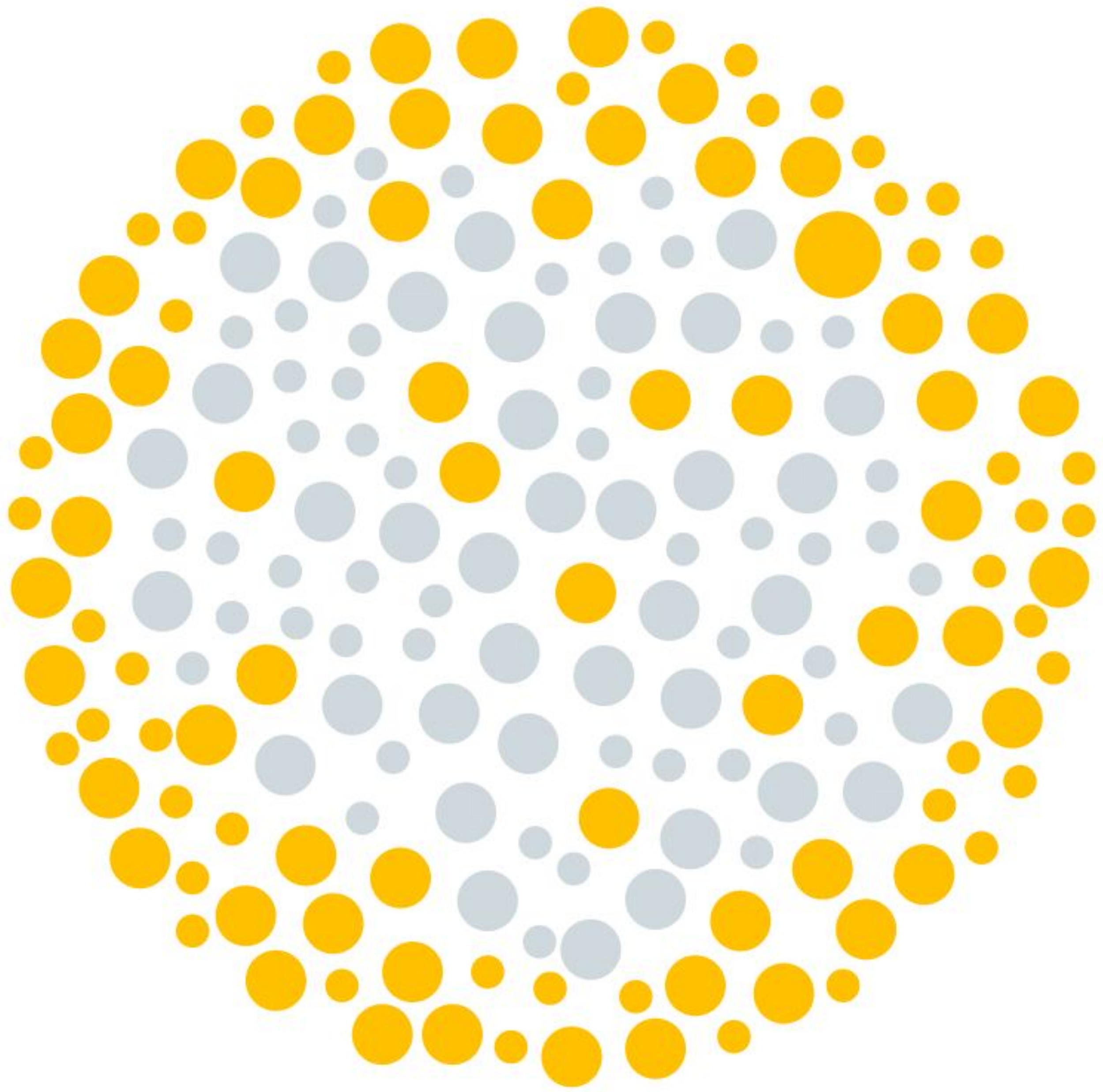
A/B Testing

Firebase A/B Testing

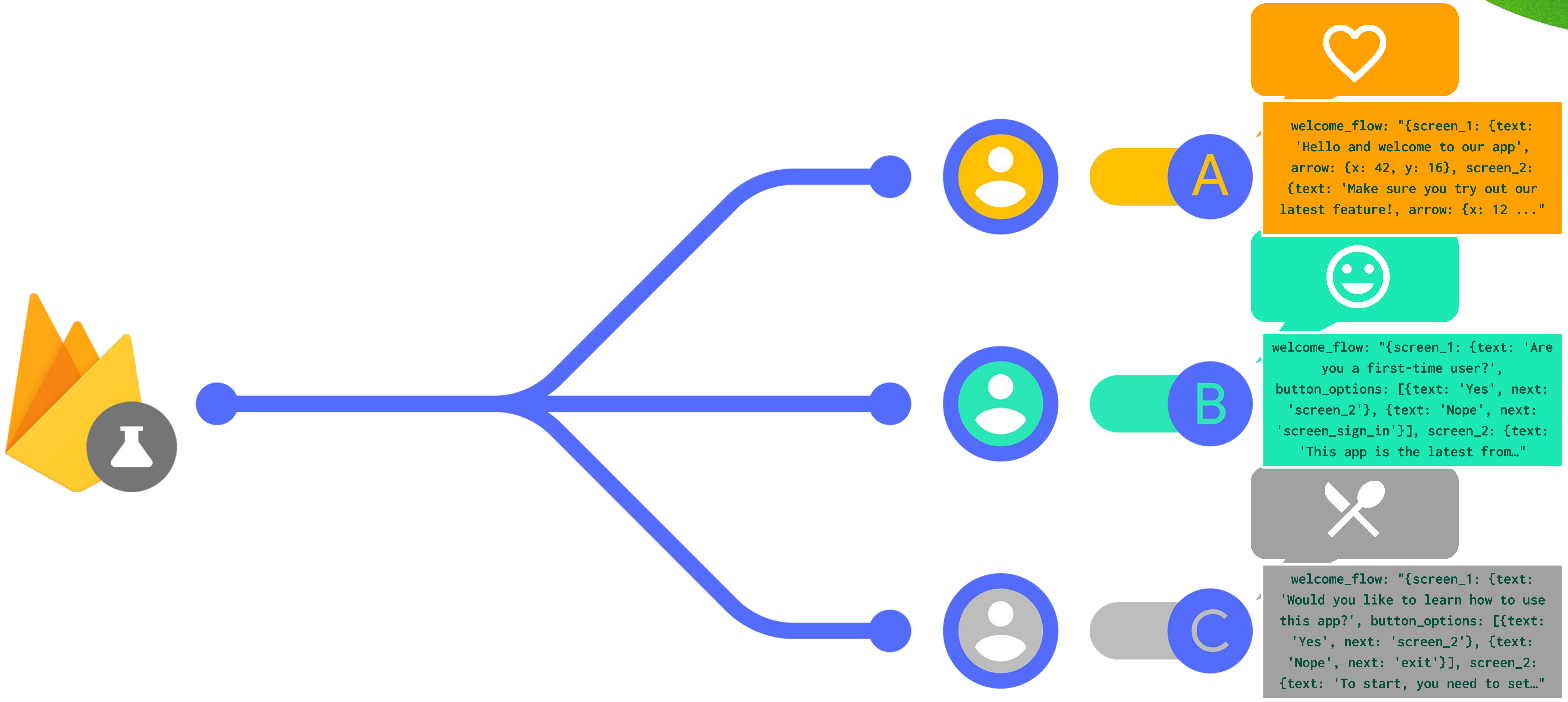
- Optimize app with multivariate experiments
- Automatic winner determination
- Integrated with Analytics, Remote Config and Cloud Messaging

1. Define your target





2. Create variants

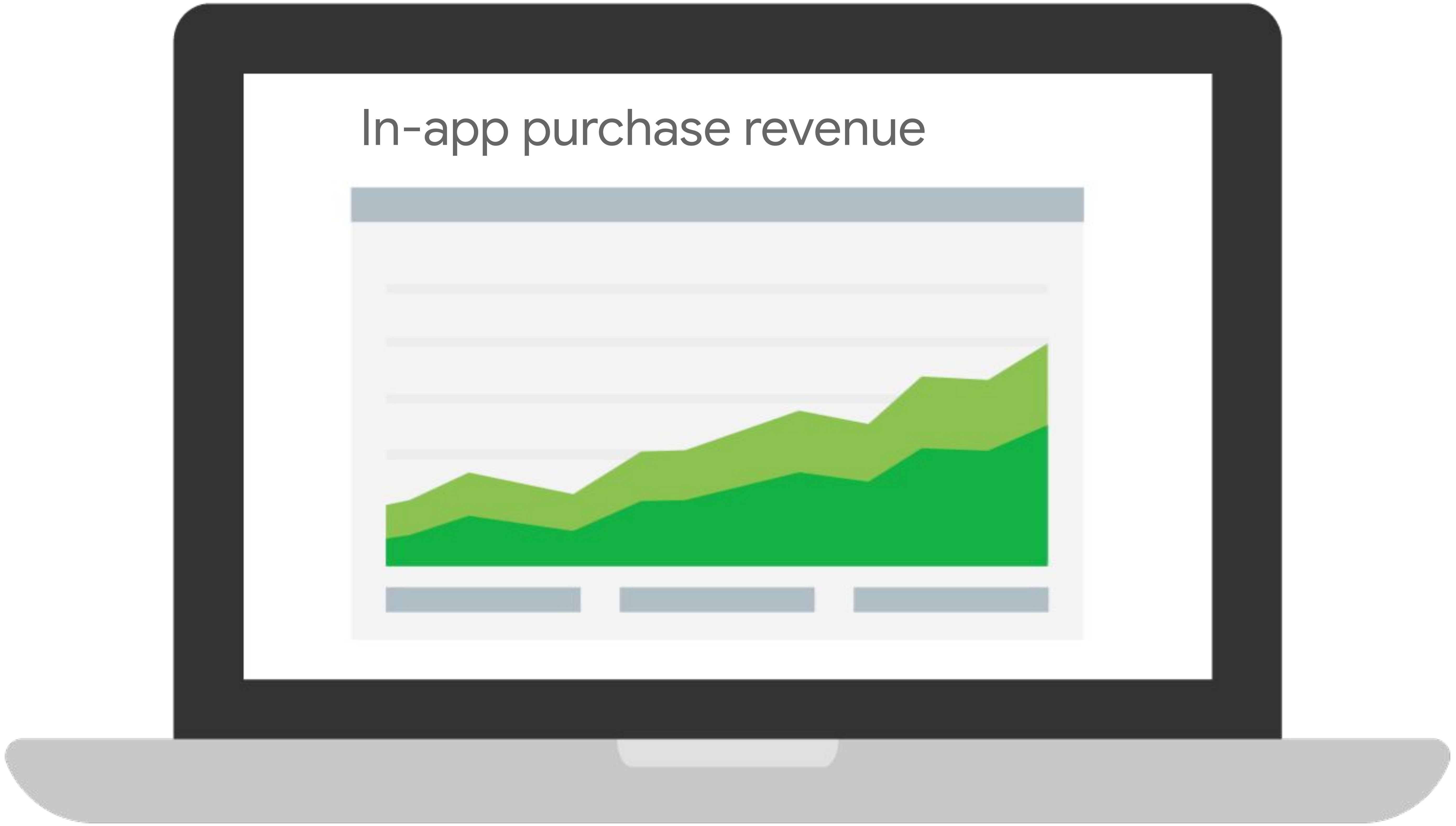


```
welcome_flow: "{screen_1: {text:
'Hello and welcome to our app',
arrow: {x: 42, y: 16}, screen_2:
{text: 'Make sure you try out our
latest feature!, arrow: {x: 12 ..."
```

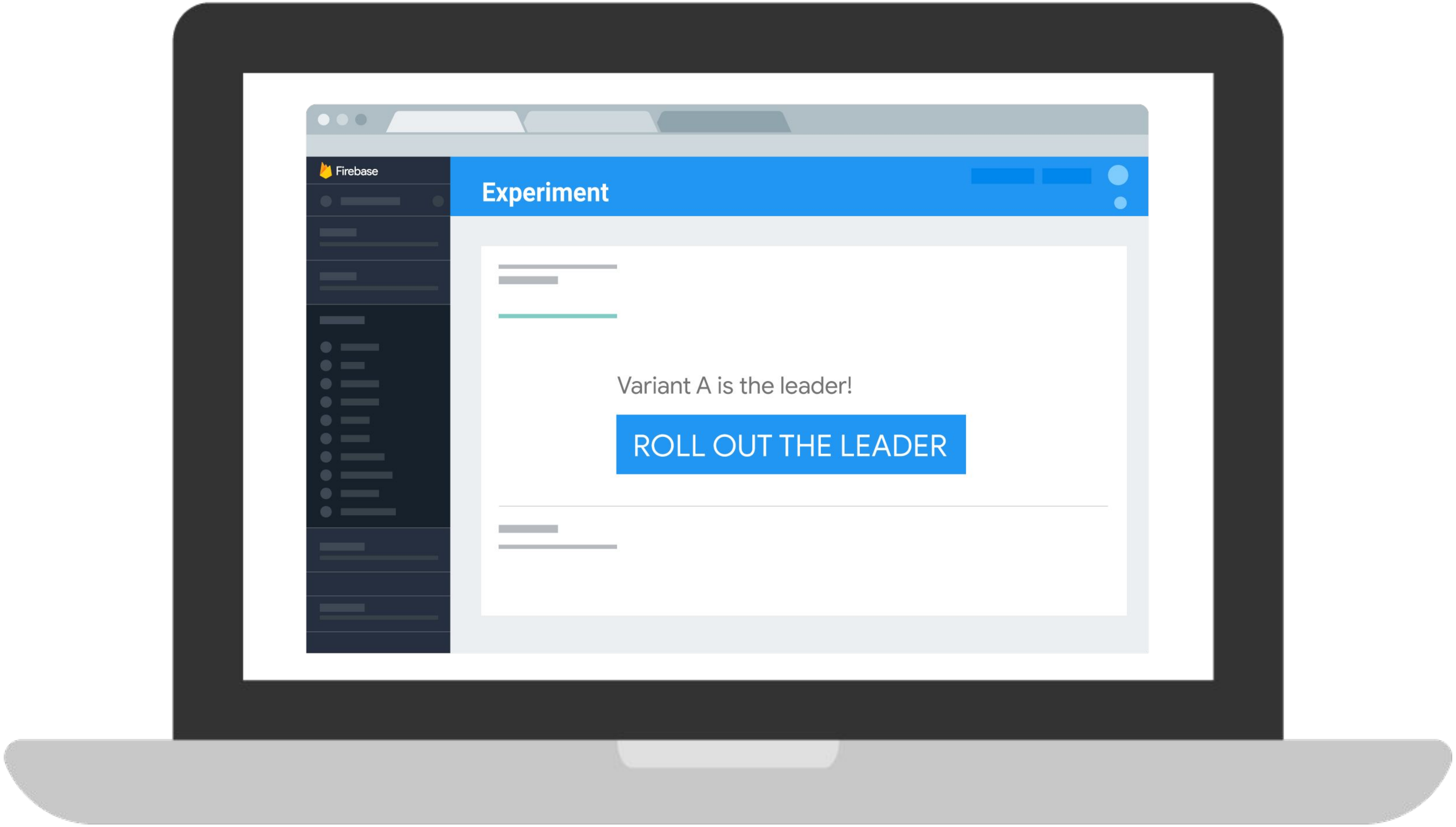
```
welcome_flow: "{screen_1: {text: 'Are
you a first-time user?',
button_options: [{text: 'Yes', next:
'screen_2'}, {text: 'Nope', next:
'screen_sign_in'}], screen_2: {text:
'This app is the latest from…"
```

```
welcome_flow: "{screen_1: {text:
'Would you like to learn how to use
this app?', button_options: [{text:
'Yes', next: 'screen_2'}, {text:
'Nope', next: 'exit'}], screen_2:
{text: 'To start, you need to set…"
```

3. Determine goals



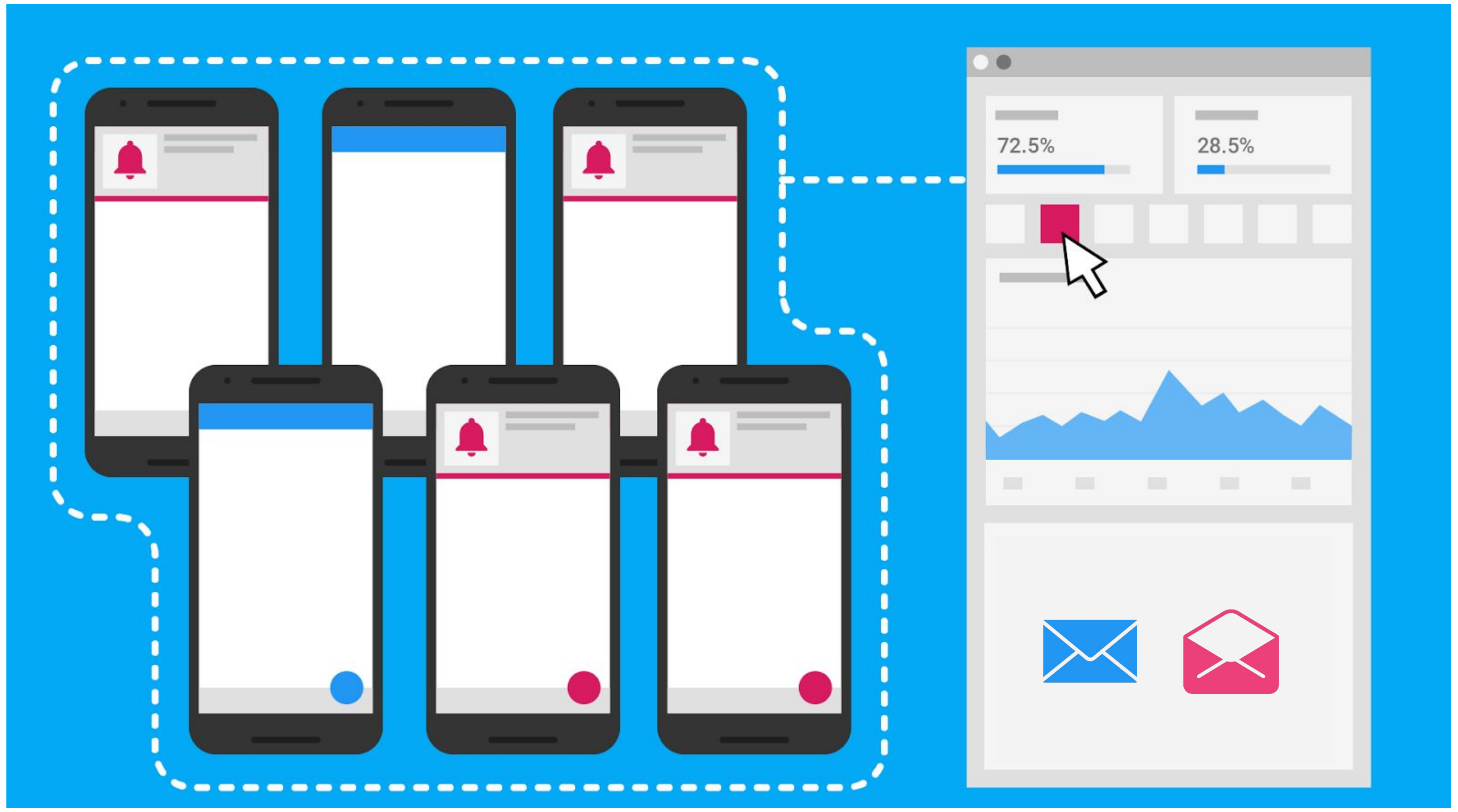
WINNER!

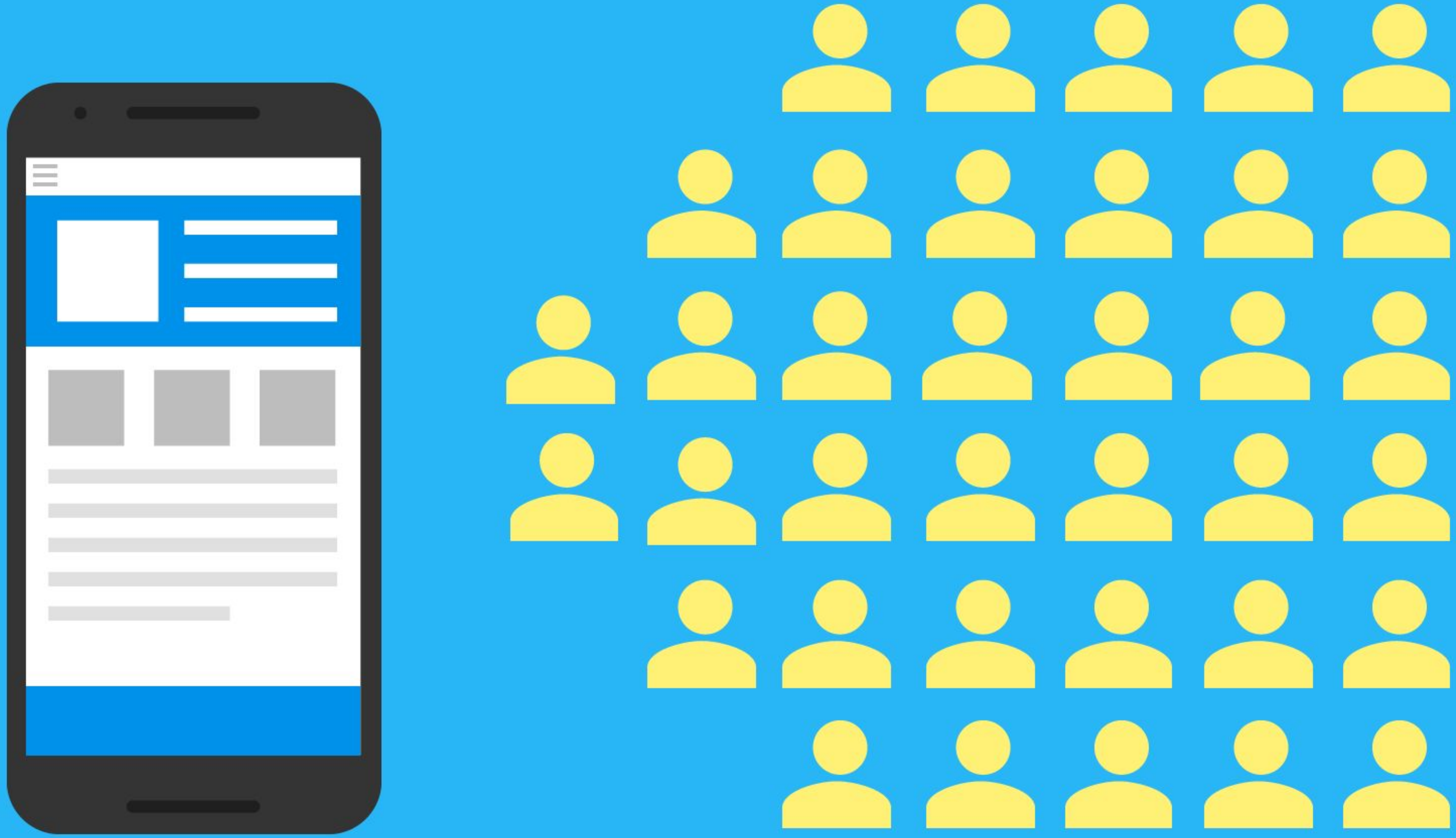

Firebase
Experiment
Variant A is the leader!
ROLL OUT THE LEADER

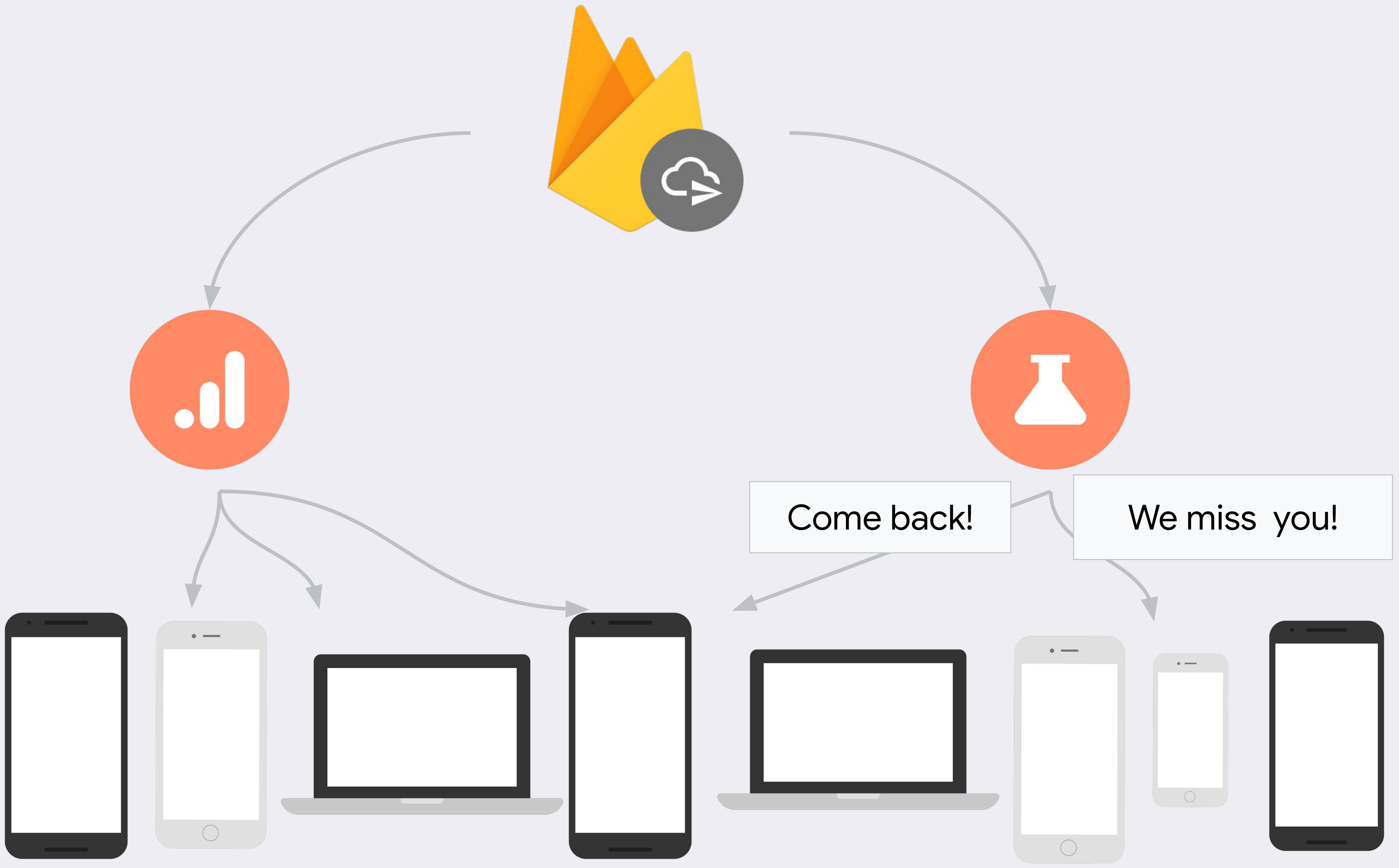
Cloud Messaging

Cloud Messaging

- Notification across Android, iOS and Web
- Audiences and custom targeting
- Engagement analytics






Come back!
We miss you!

Thank you