Zephyr® Project

Developer Summit

Zephyr® Project
Developer Summit

Time-to-Market with Zephyr, Pigweed, and Firebase

Yuval Peress, *Google*

Email: peress@google.com
Discord: yuval#5515
GitHub: yperess

#EMBEDDEDOSSUMMIT

Timelines are everything

- In both big and small companies (even personal projects) time is expensive
- Making a great product is hard, iterating on it is harder
- Firebase is great for user facing projects, I wouldn’t start a new project without it:
    - remote configs
    - feature flags
    - analytics (user engagement)
    - crashlytics
- Pigweed bridges the gap for embedded applications wanting to get going quickly
- Can we have both?

Let's build a product

- What’s our product?
- Getting set up
- Configuring logging
- Application logic
- Tracking events
- Remote config (A/B testing or feature rollout)
- Crash reporting

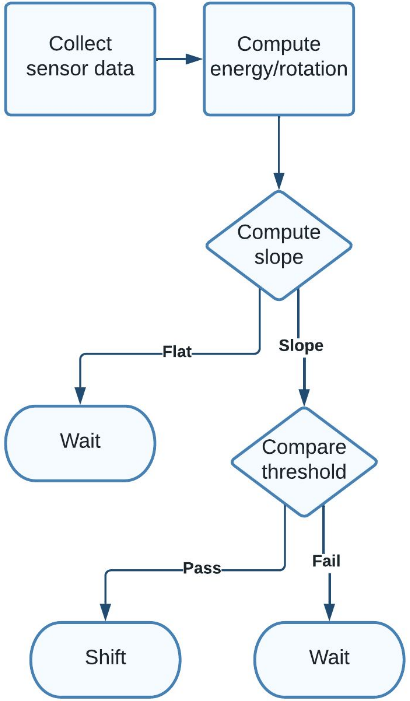
Zephyr® Project
Developer Summit

What’s our product?

Automatic gear shifter for cycling

- Inputs:
  - Cadence (RPMs)
  - Power (Watts)
  - Speed (m/s)
  - Gravity vector (climb gradient)
  - Manual gear override (up/down gear shifts)
- Output:
  - Gear selection (rear and front cassette)

A simple decision graph



Collect sensor data

Compute energy/rotation

Compute slope

Flat

Slope

Wait

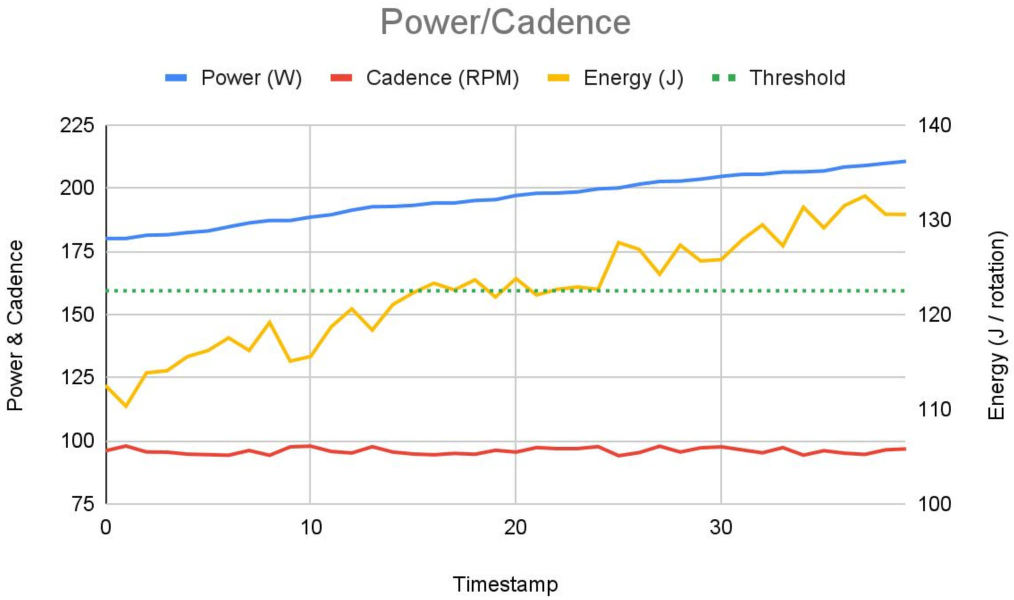
Compare threshold

Pass

Fail

Shift

Wait



Getting set up

Create a west.yml

```
manifest:
  remotes:
    - name: zephyr
      url-base: https://github.com/zephyrproject-rtos
    - name: pigweed
      url-base: https://pigweed.googlesource.com/pigweed

  projects:
    - name: zephyr
      remote: zephyr
      revision: main
      import: true
    - name: pigweed
      remote: pigweed
      revision: main
```

Select kConfig values

```
# Enable C++
CONFIG_CPP=y
CONFIG_STD_CPP17=y
CONFIG_CPP_MAIN=y
```

main.cc

```
#include <zephyr/logging/log.h>


int main() {
  LOG_INF(“Hello Zephyr logs”);
  return 0;
}
```

Building

- Environment setup:
  - On every run of west update:
    ```
    $ . ./pigweed/bootstrap.sh
    ```
  - When you start working:
    ```
    $ . ./pigweed/activate.sh
    ```
- Build:
  ```
  $ west build -p -b ${BOARD} app
  ```

Zephyr® Project
Developer Summit

Configuring logging

Select kConfig values

```
# Enable C++
CONFIG_CPP=y
CONFIG_STD_CPP17=y
CONFIG_CPP_MAIN=y
```

```
# Enable tokenized RPC logging
CONFIG_PIGWEED_LOG_TOKENIZED_RPC=y
```

Experimental, details may change

What just happened?

- `CONFIG_PIGWEED_LOG_TOKENIZED_RPC` enables by default tokenized logging.
- Pigweed logging is intercepting the Zephyr logs too
- Pigweed generated a hash based token database for logs
- All logs are put into a proto [message LogEntry]
- Log messages are sent via a service stream to the host on a common HDLC server
- Messages can be decoded via python or webconsole
  - Python: `$ python3 -m pw_system.console --device /dev/ttyUSB1 --token-database build/database.csv`
  - Webconsole…

Experimental, details may change

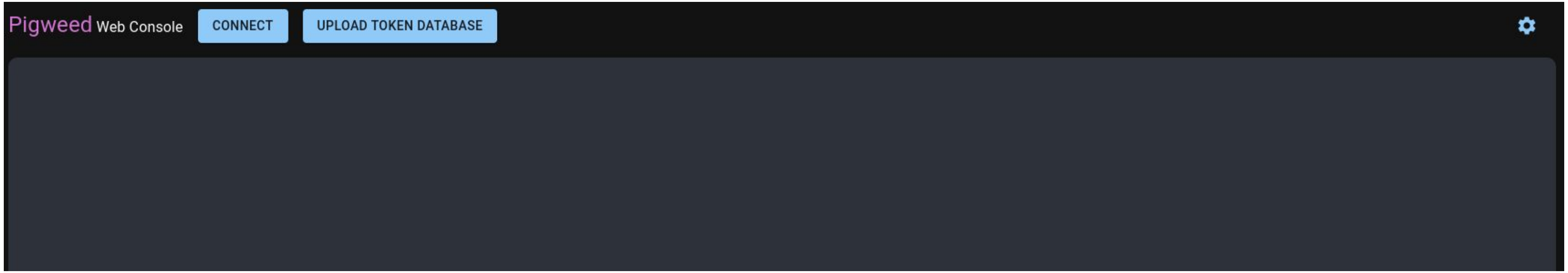
Decoding (python)

```
$ python3 -m pw_system.console \
  --device /dev/ttyUSB0 \
  --token-database build/database.csv
```

Experimental, details may change

Decoding (Webconsole)

```
$ cd pigweed/pw_web/webconsole
$ npm install --force
$ npm run dev
```



Experimental, details may change

Decoding (anywhere else)

- The generated database can be shared
- Decoding can be done in:
  - Python
  - C++
  - TypeScript / JavaScript
  - Java (Android)

Experimental, details may change

Application logic

After sensor processing

- Energy data will be placed in a buffer of q31_t values
  - Range is [-1,1]
  - Requires an int8_t shift value for scaling
- Get the currently configured threshold (using RemoteConfig RPC)
- Iterate through the buffer & compare against threshold
- On gear shift, register event

Experimental, details may change

Zephyr® Project
Developer Summit

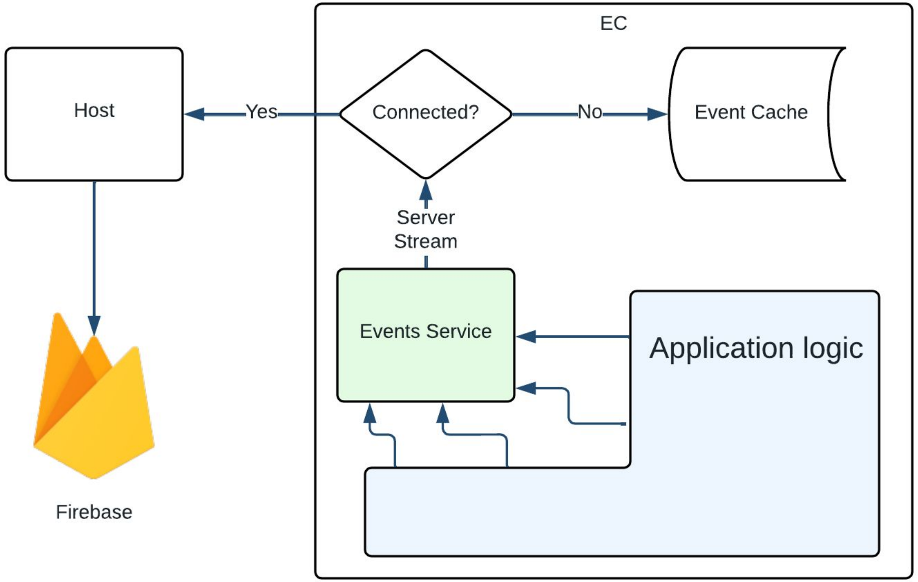
Tracking events

#EMBEDDEDOSSUMMIT

Event logging data path

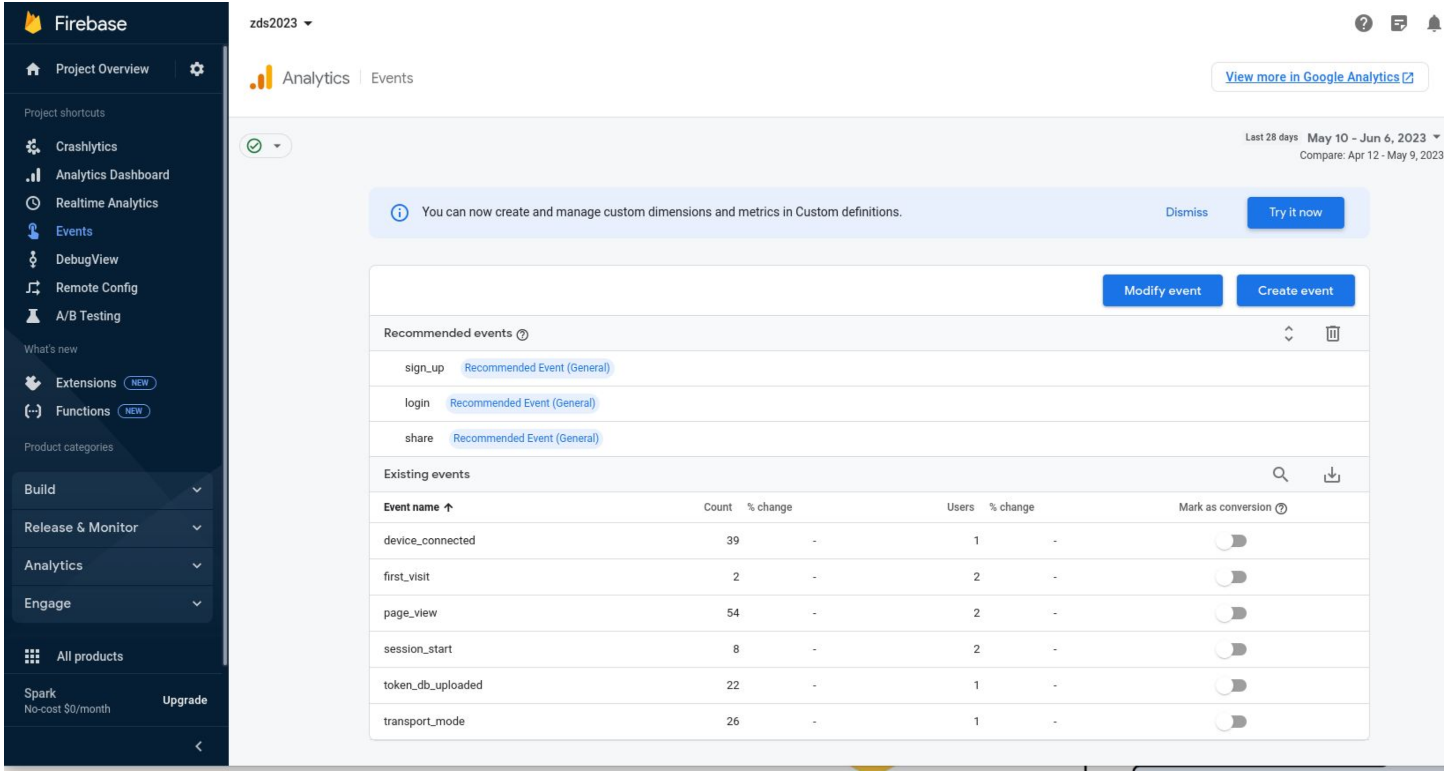
Application logic passes event data to the FirebaseEventsService

```
message FirebaseEventParam {
  string key = 1;
  oneof value {
    int64 int_value = 2;
    bool bool_value = 3;
    string string_value = 4;
  };
};

message FirebaseEvent {
  string name = 1;
  repeated FirebaseEventParam params = 2;
};

service FirebaseEventService {
  rpc Listen(FirebaseEventRequest)
      returns (stream FirebaseEvent);
}
```



Experimental, details may change

Zephyr® Project
Developer Summit



You can now create and manage custom dimensions and metrics in Custom definitions.

Recommended events

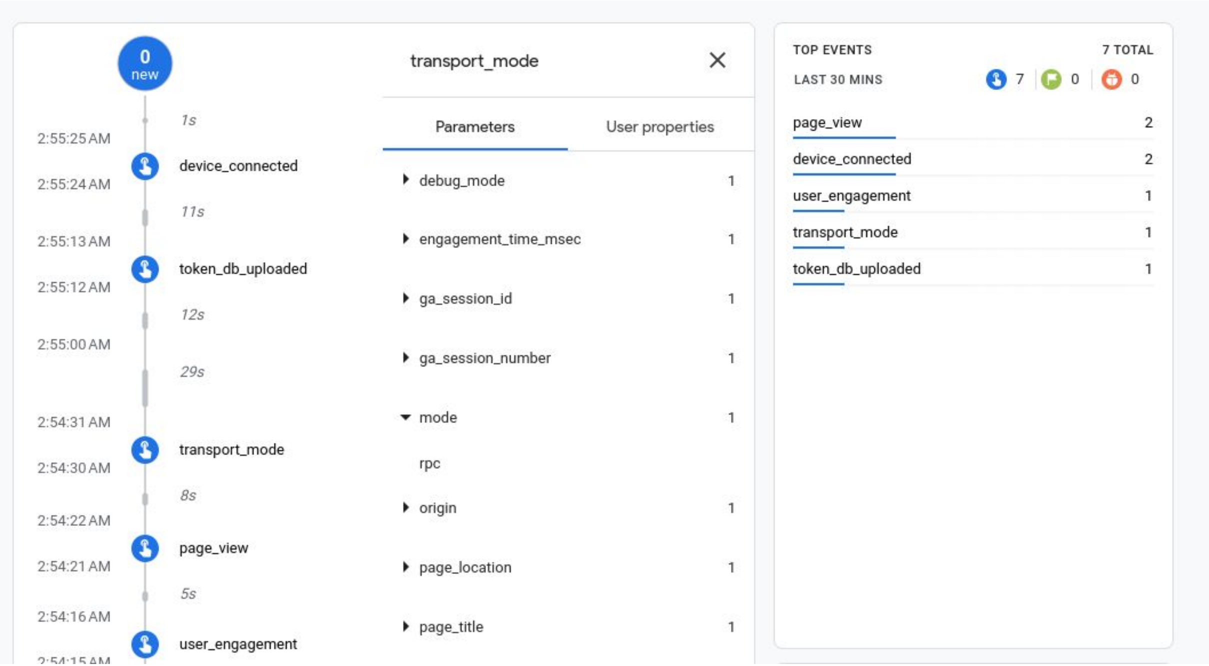
sign_up	Recommended Event (General)
login	Recommended Event (General)
share	Recommended Event (General)

Existing events

Event name ↑	Count	% change	Users	% change	Mark as conversion
device_connected	39	-	1	-	
first_visit	2	-	2	-	
page_view	54	-	2	-	
session_start	8	-	2	-	
token_db_uploaded	22	-	1	-	
transport_mode	26	-	1	-	

Experimental, details may change

Sample of events in DebugView



0
new

1s
2:55:25 AM
device_connected
2:55:24 AM
11s
2:55:13 AM
token_db_uploaded
2:55:12 AM
12s
2:55:00 AM
29s
2:54:31 AM
transport_mode
2:54:30 AM
8s
2:54:22 AM
page_view
2:54:21 AM
5s
2:54:16 AM
user_engagement
2:54:15 AM

transport_mode

▸ debug_mode	1
▸ engagement_time_msec	1
▸ ga_session_id	1
▸ ga_session_number	1
▾ mode	1
rpc	
▸ origin	1
▸ page_location	1
▸ page_title	1

TOP EVENTS 7 TOTAL

LAST 30 MINS 7 0 0

page_view	2
device_connected	2
user_engagement	1
transport_mode	1
token_db_uploaded	1

Experimental, details may change

Zephyr® Project
Developer Summit

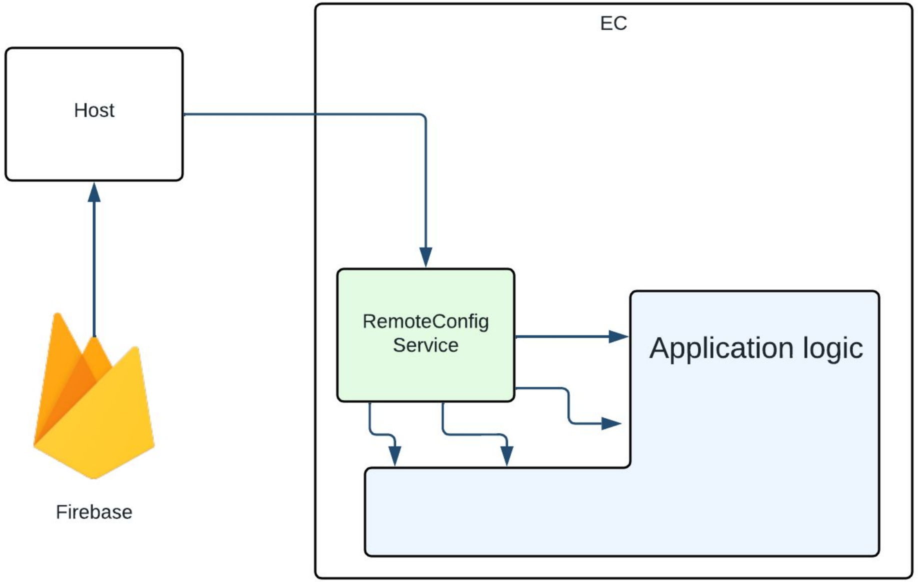
Remote config & A/B testing

#EMBEDDEDOSSUMMIT

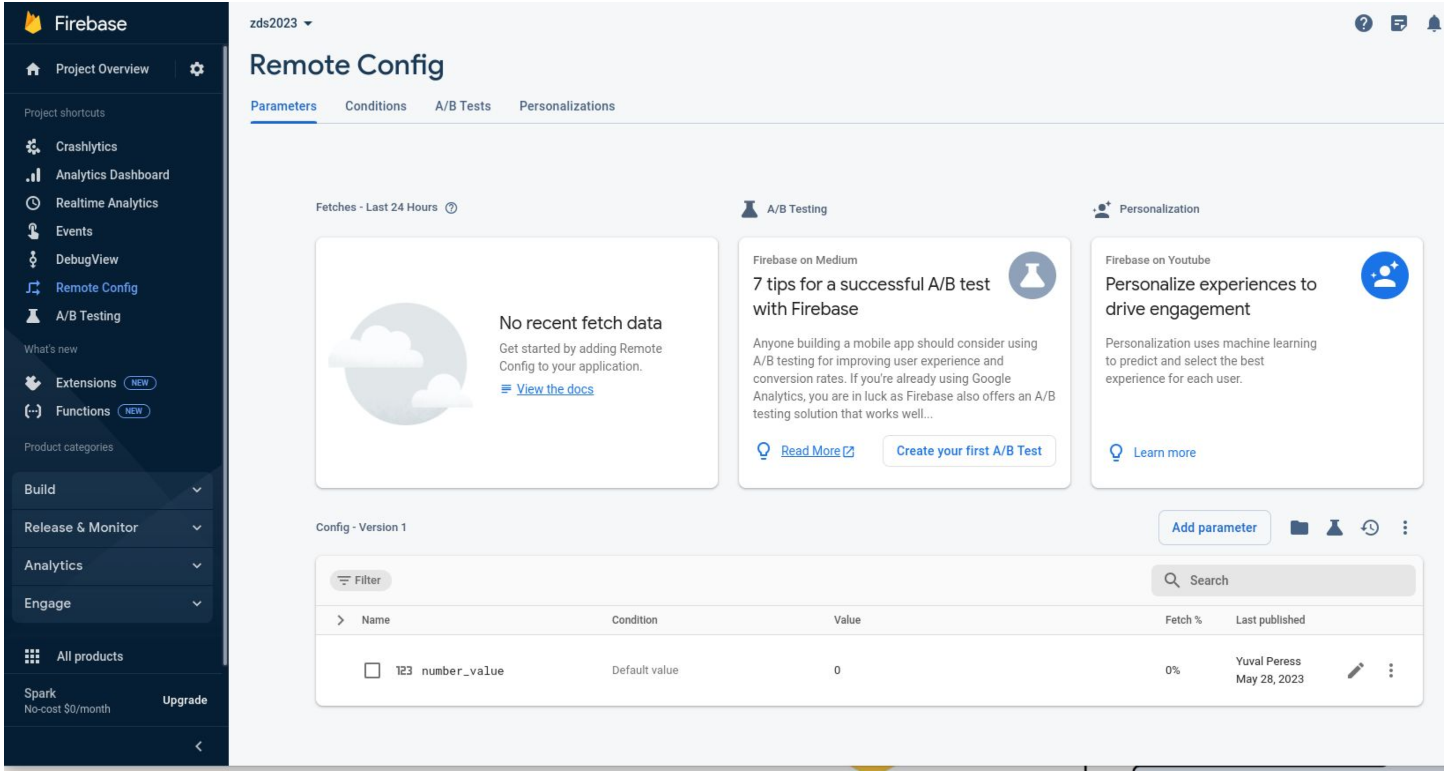
Remote config data path

Host passes config data to the FirebaseRemoteConfigService

```
message FirebaseRemoteConfig {
  string key = 1;
  oneof value {
    int64 int_value = 2;
    bool bool_value = 3;
    string string_value = 4;
  };
};

message FirebaseRemoteConfigResponse {};

service FirebaseRemoteConfigService {
  rpc Set(stream FirebaseRemoteConfig)
      returns
(FirebaseRemoteConfigResponse);
}
```



Experimental, details may change



Firebase

Project Overview

Project shortcuts

- Crashlytics
- Analytics Dashboard
- Realtime Analytics
- Events
- DebugView
- Remote Config
- A/B Testing

What's new

- Extensions NEW
- Functions NEW

Product categories

- Build
- Release & Monitor
- Analytics
- Engage

All products

Spark
No-cost $0/month
Upgrade

zds2023

Remote Config

Parameters | Conditions | A/B Tests | Personalizations

Fetches - Last 24 Hours

No recent fetch data

Get started by adding Remote Config to your application.

View the docs

A/B Testing

Firebase on Medium

7 tips for a successful A/B test with Firebase

Anyone building a mobile app should consider using A/B testing for improving user experience and conversion rates. If you're already using Google Analytics, you are in luck as Firebase also offers an A/B testing solution that works well...

Read More

Create your first A/B Test

Personalization

Firebase on Youtube

Personalize experiences to drive engagement

Personalization uses machine learning to predict and select the best experience for each user.

Learn more

Config - Version 1

Add parameter

Filter

Search

Name	Condition	Value	Fetch %	Last published
☐ 123 `number_value`	Default value	0	0%	Yuval Peress May 28, 2023

Experimental, details may change

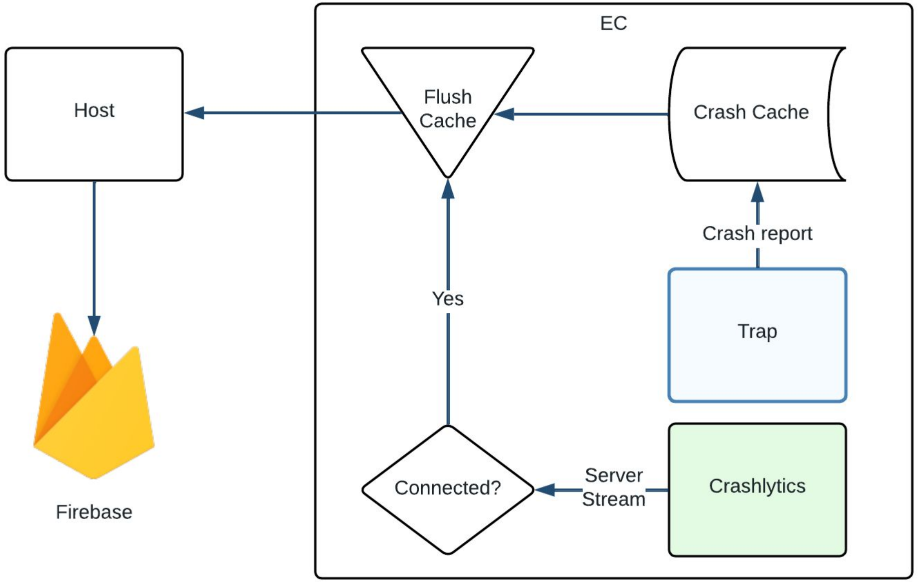
Zephyr® Project
Developer Summit

Crash reporting

#EMBEDDEDOSSUMMIT

Crash reporting concepts

- It should work like the events service
- On crash, stack information is saved to persistent memory
- On host connection, flush the last crash event to the host



Experimental, details may change

Zephyr® Project
Developer Summit

Putting it all together

- Use RemoteConfig:
  - to parameterize the algorithms
  - when introducing new features
  - with A/B testing to check if a new configuration is better
- Use events:
  - to see common user journeys in the field
  - to reproduce crashes
  - to help inform your application logic and focus future investment
- Use crashlytics:
  - to understand when users are running into issues
  - to provide improved and more reliable user experience

In our example

- Add event logging for manual gear override
- Gear switching algorithm will have a RemoteConfig downshift and upshift energy/rotation threshold
- When we improve the algorithm (to include the gradient in the equation) we can put the new calculation behind a bool RemoteConfig for a slower rollout

Zephyr® Project
Developer Summit

Questions?