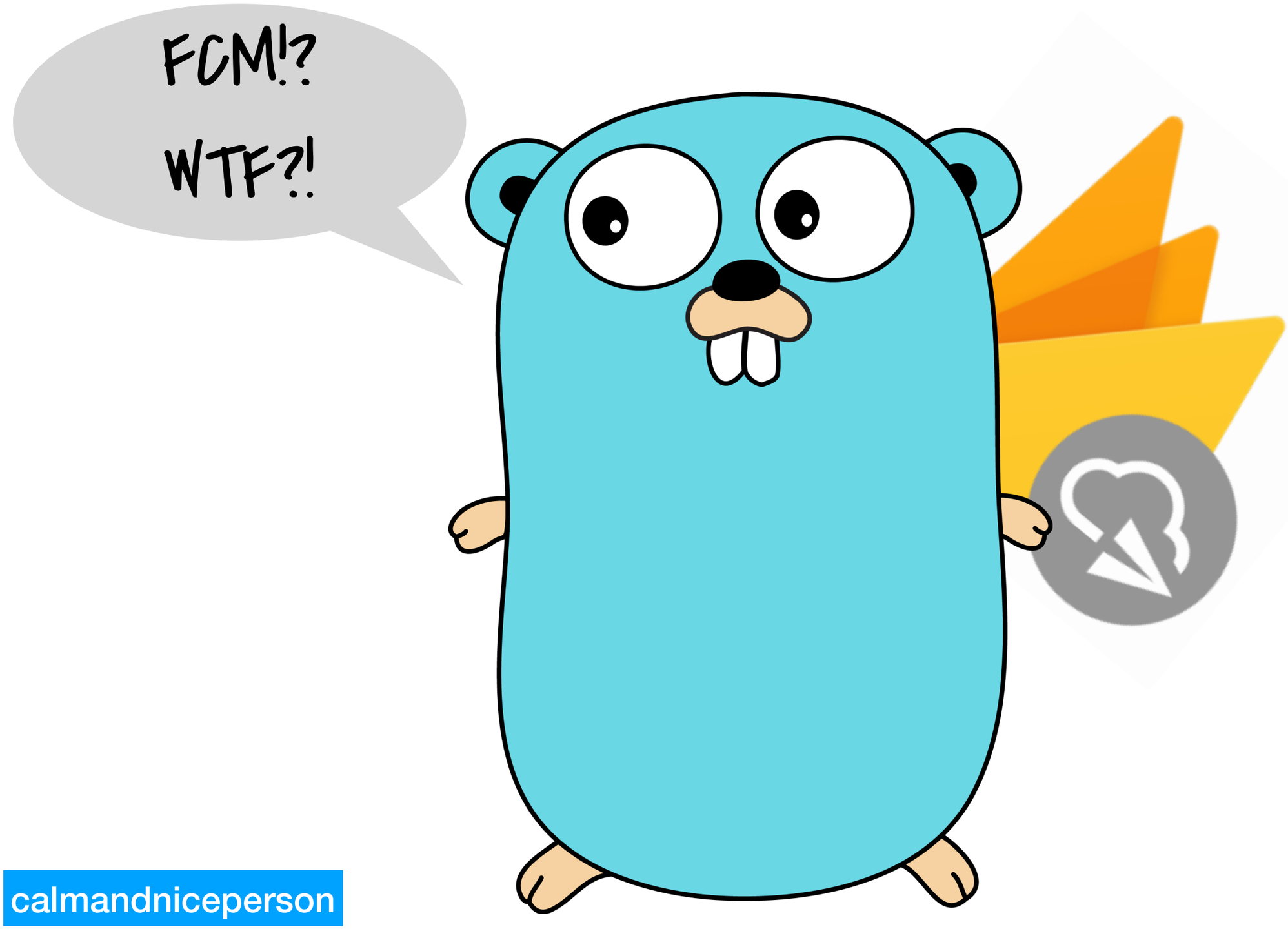


calmandniceperson

@rettetdemdativ

History lesson

- Android Cloud to Device Messaging in 2010
- Google Cloud Messaging in 2012
- Firebase Cloud Messaging 2016

What can we do with it?

- Send notifications
- Send data —> limitless possibilities
- ... to:
  - individual devices
  - topics
  - apps
  - platforms

Notifications

Compose message

A/B Testing

Message text

Hello, Vienna Go Meetup!

Message label (optional)

Enter message nickname

Delivery date | Time | Time zone

Send Later | 1/24/18 | 09:10 | (GMT+01:00) GMT+01:00

Recipient time zone

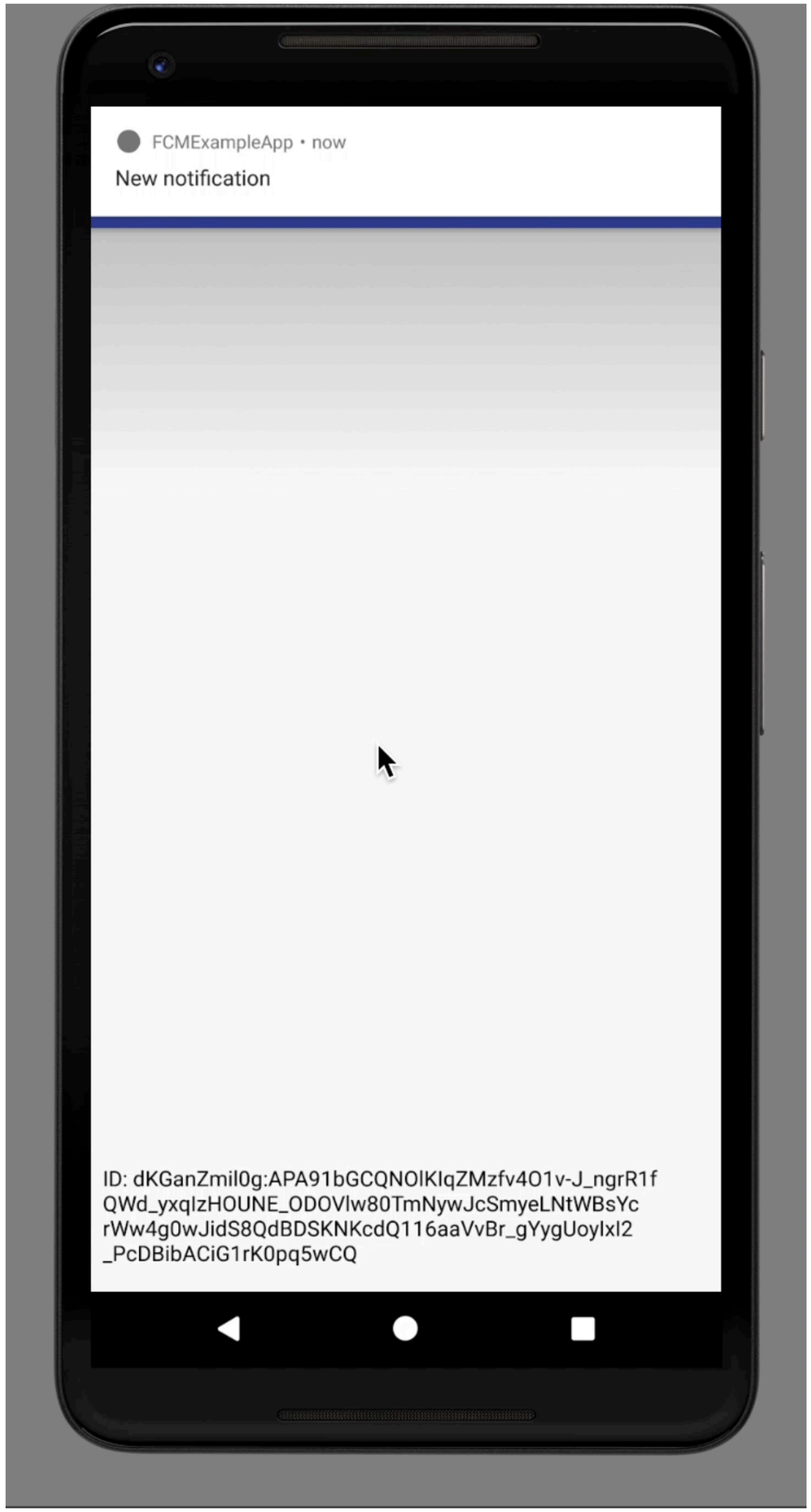
Target

User segment | Topic | Single device

Target user if...

App | go.vienna.fcmexample.fcmexampleapp | AND

Cannot add additional statements. All apps have been selected.

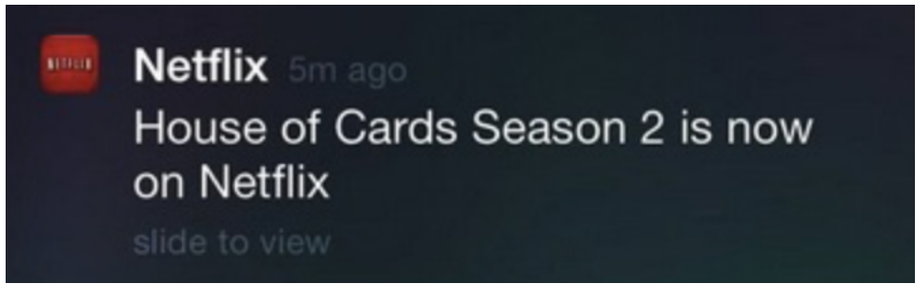


What can we do with it?

- Send notifications
- Send data —> limitless possibilities
- ... to:
  - individual devices
  - topics
  - apps
  - platforms

Real-life examples

- Friends nearby? Notify your user!
- Let ‘em chat!



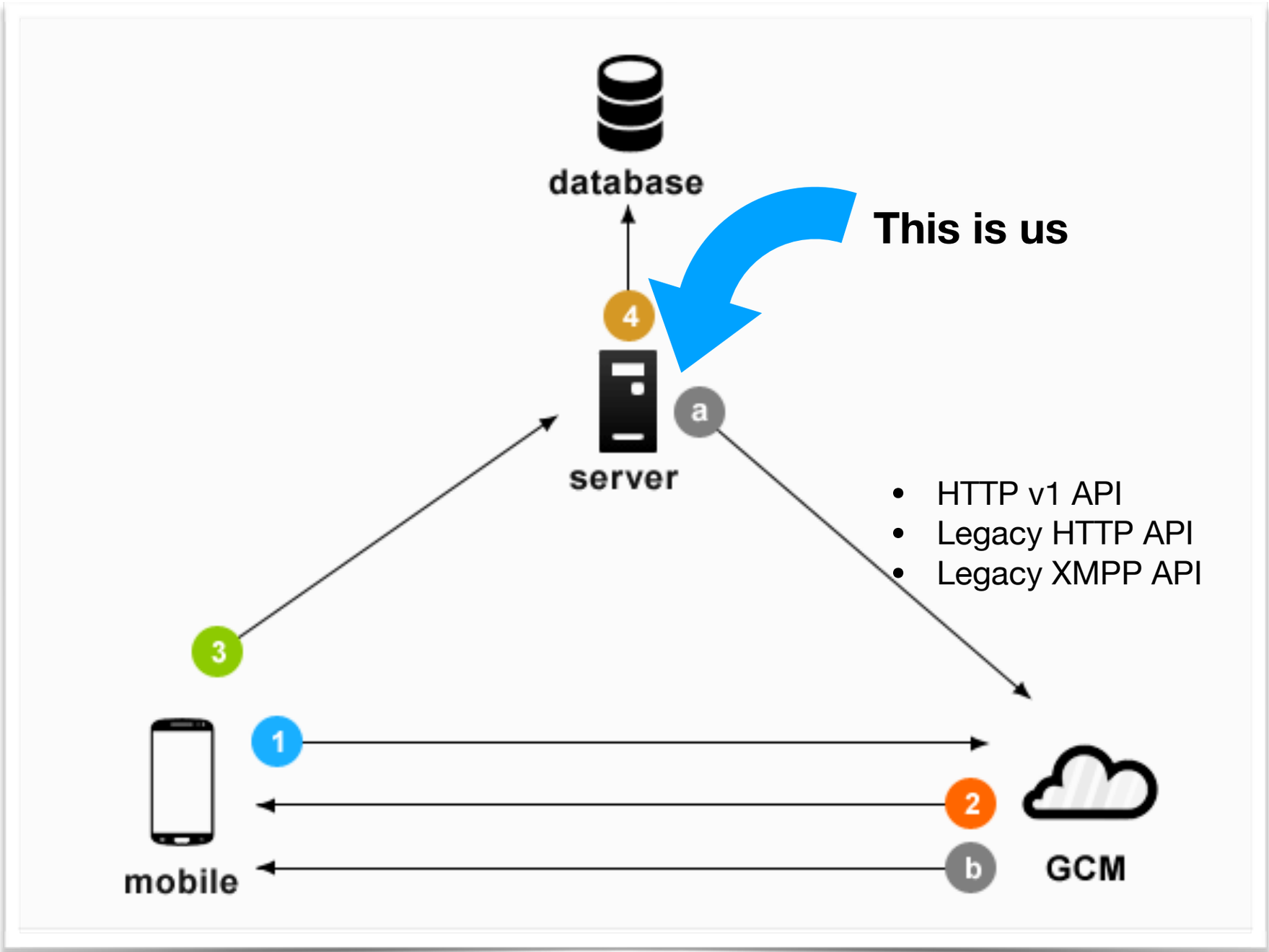
When to use it

- Ease of development
- Battery efficiency
- Lower data consumption

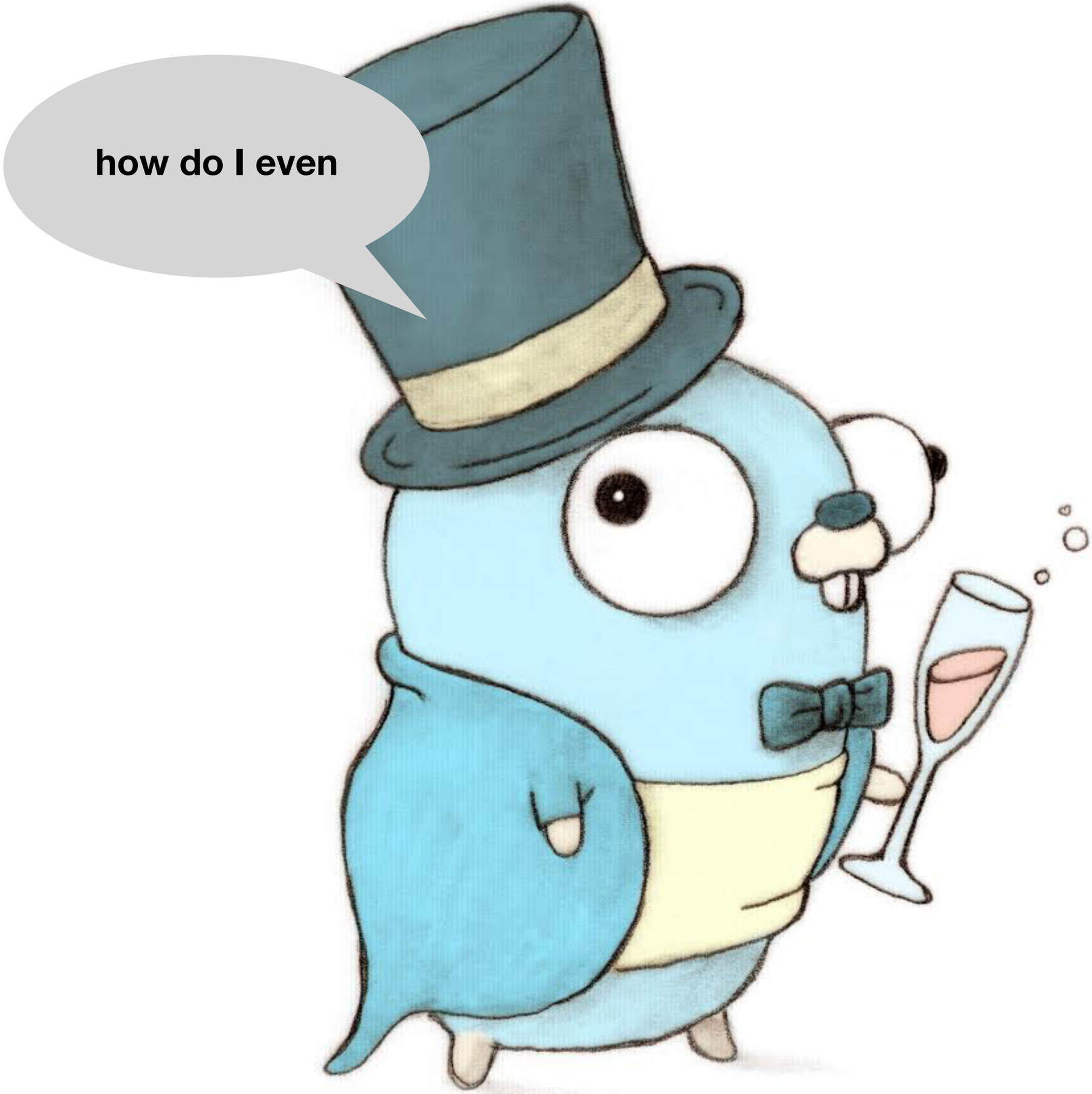
When **NOT** to use it

- When you really want to be in control

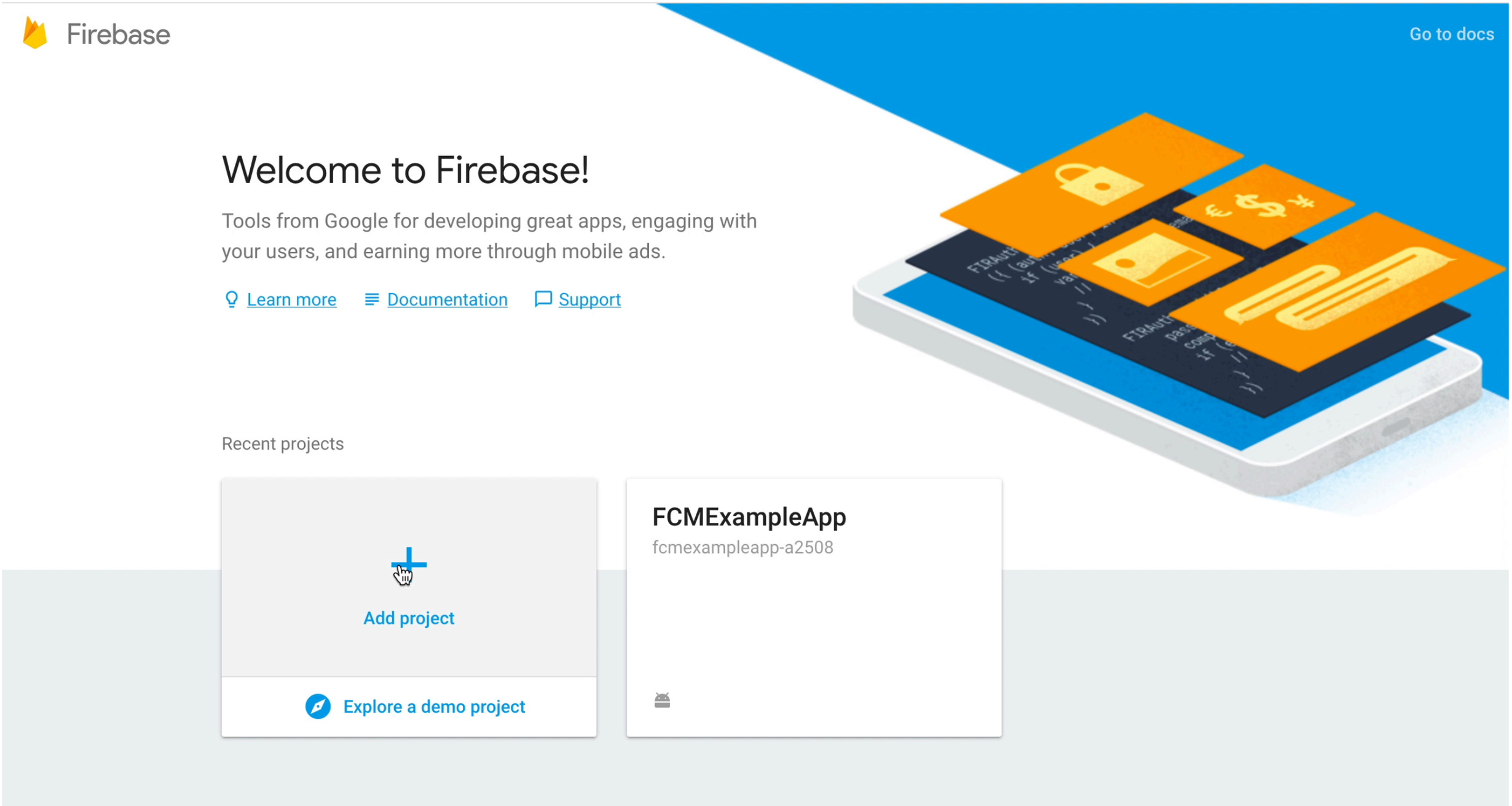
But… how does it work?



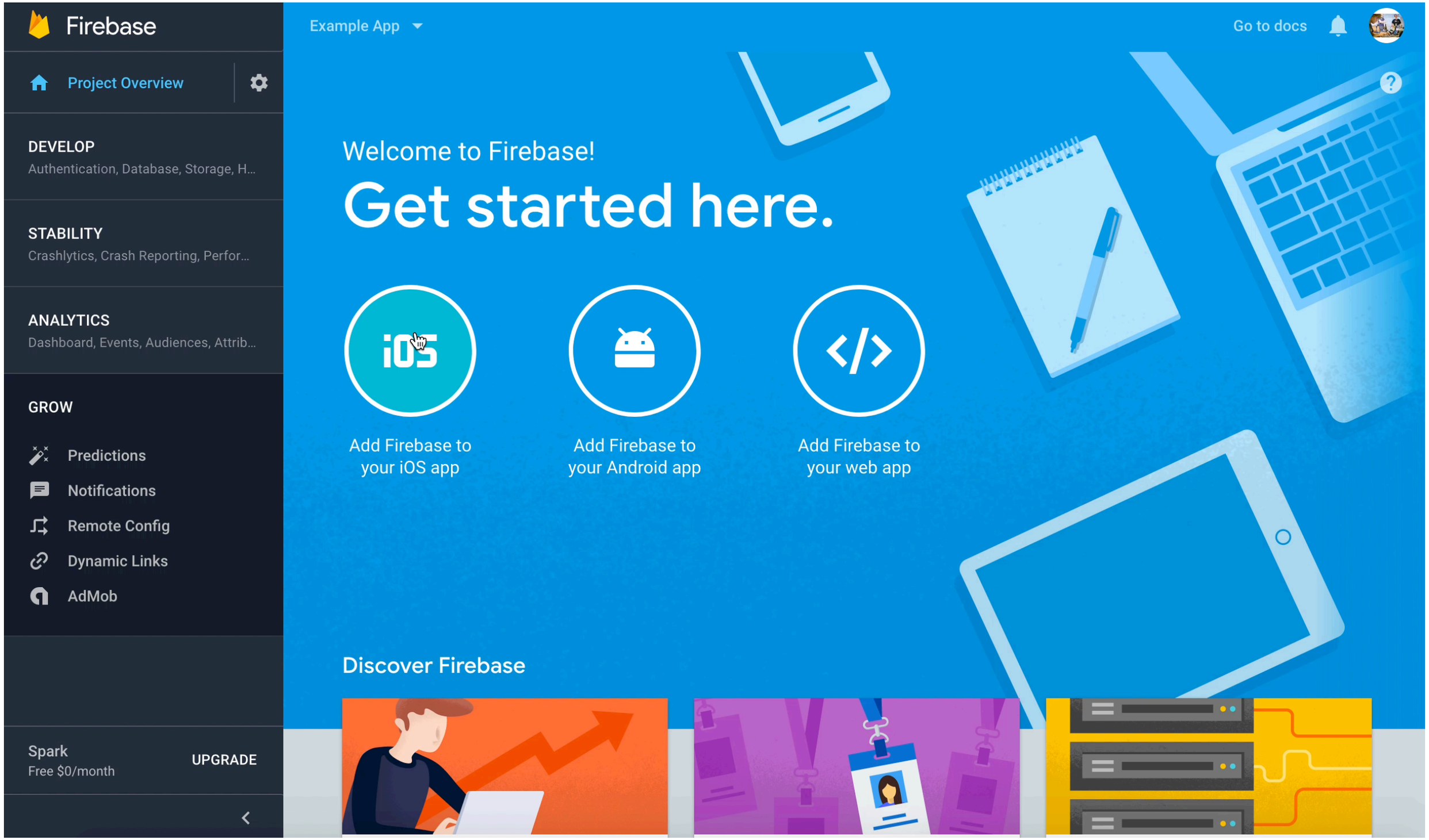
Source: https://www.drupalproject.org/project/c2dm


how do I even

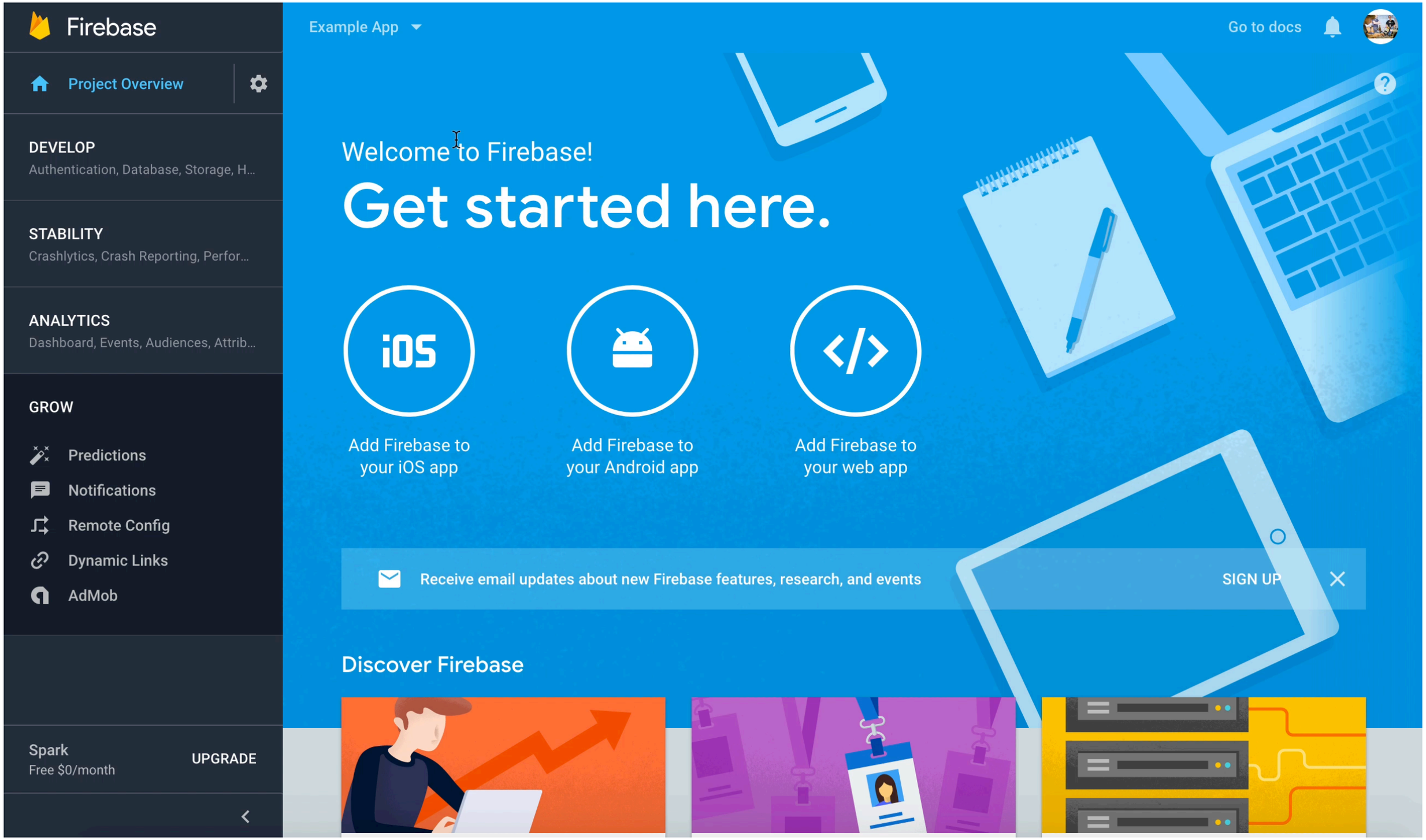
1. Register a project



2. Add your app and follow the guide



3. Get your API key



4. Define our message type

```go
type fcmMessage struct {
    To              string      `json:"to,omitempty"`
    RegistrationIDs []string    `json:"registration_ids,omitempty"`
    Data            interface{} `json:"data,omitempty"`
}
```

5. Fill it with data

Name it whatever you want

```go
m := fcmMessage{
    RegistrationIDs: ids,
    Data:            map[string]string{"message": message},
}
```

6. Create request

```
r, err := http.NewRequest("POST", fcmServerURL, bytes.NewReader(jd))
if err ≠ nil {
    log.Printf("Failed to create new HTTP request: %s", err.Error())
}
```

7. Attach headers and send

```
// Add the Authorization header with the API key
r.Header.Set("Authorization", fmt.Sprintf("key=%s", fcmKey))

r.Header.Set("Content-Type", "application/json")

client := &http.Client{}
resp, err := client.Do(r)
```

Let’s try the example

Any questions?

Find the code here:

`github.com/calmandniceperson/fcm_example_server`
`github.com/calmandniceperson/FCMExampleApp`