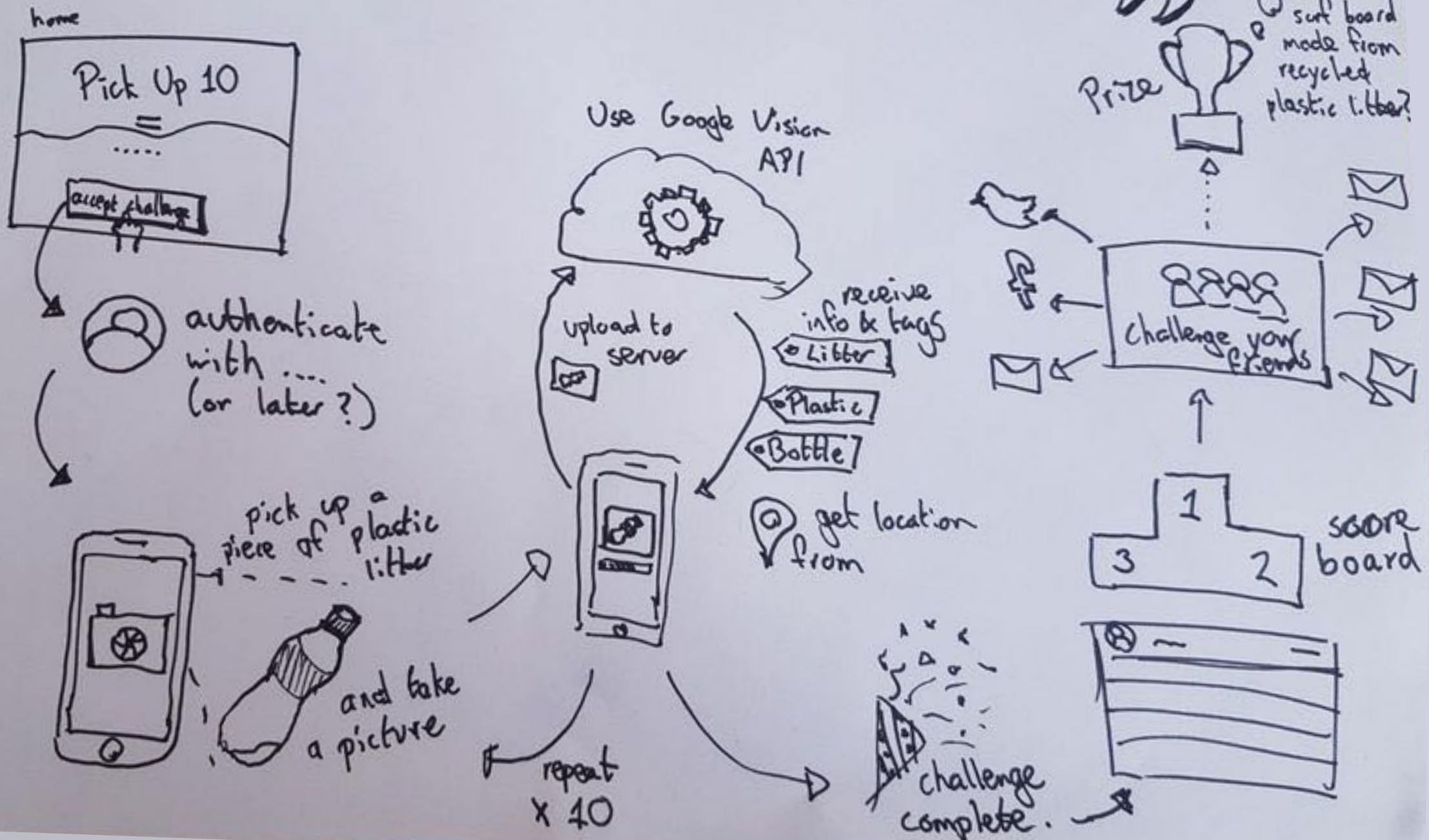
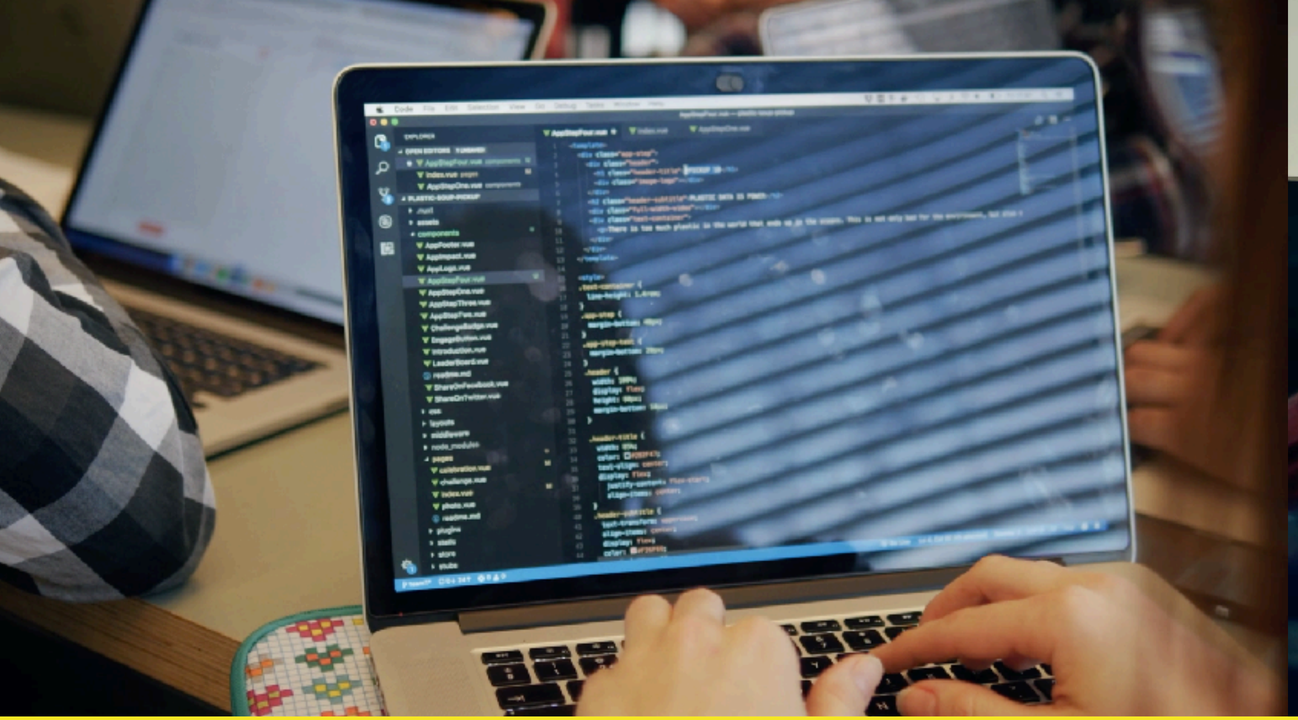
PICK UP 10
SERVERLESS STRUGGLES

DE VOORHOEDE
front-end developers

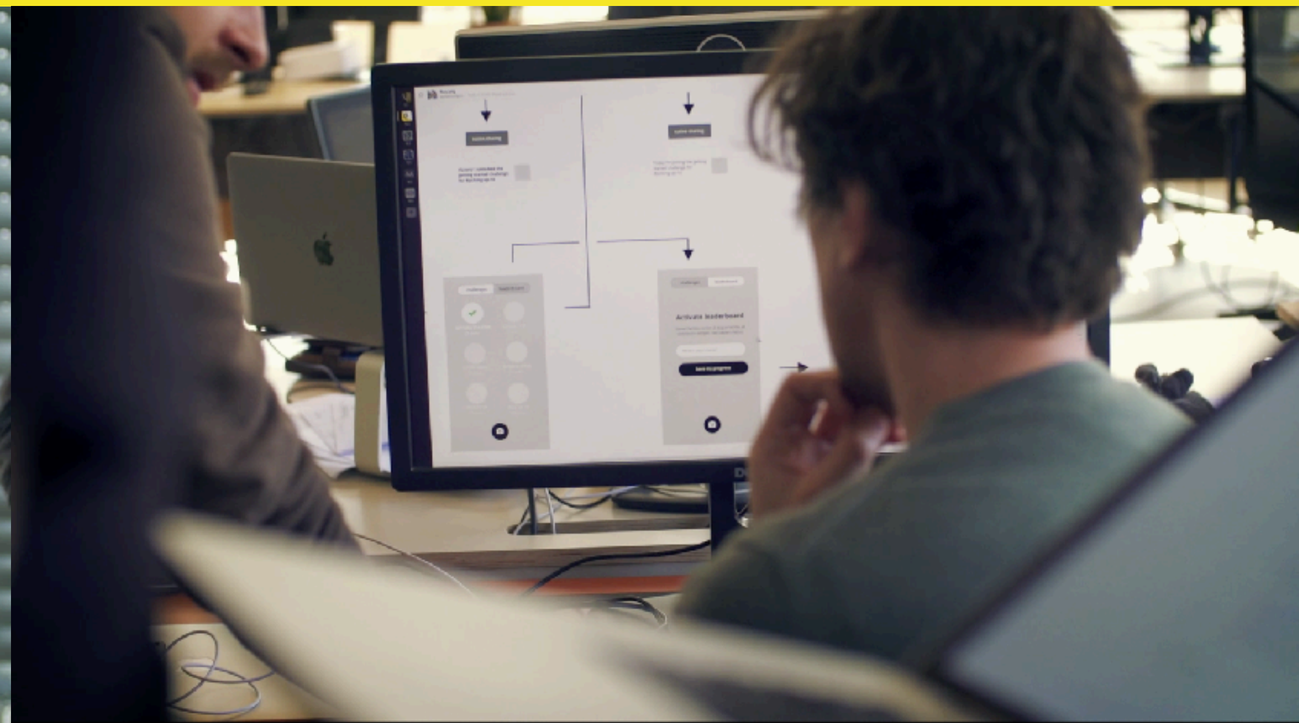
JASPER MOELKER
@jbmoelker

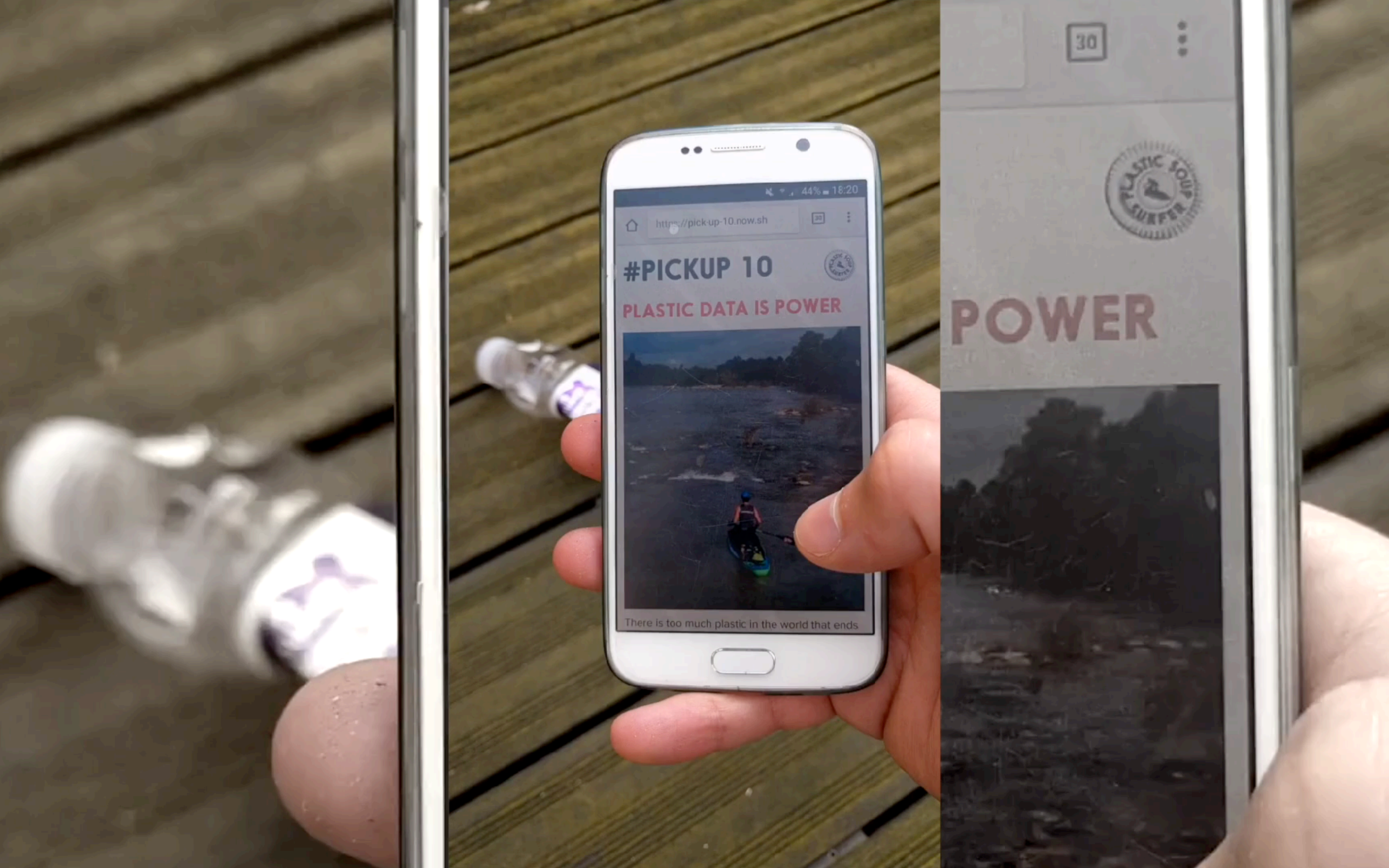
Pick Up 10


home
Pick Up 10
accept challenge
authenticate with ... (or later?)
pick up a piece of plastic litter
and take a picture
Use Google Vision API
upload to server
receive info & tags
Litter
Plastic
Bottle
get location from
repeat x 10
challenge complete.
Prize
surf board made from recycled plastic litter?
challenge your friends
1
3
2
score board



HACK-A-THON




44% 18:20
https://pick-up-10.now.sh
#PICKUP 10
PLASTIC DATA IS POWER
There is too much plastic in the world that ends
30
PLASTIC SOUP SURFER
POWER

GOING SERVERLESS

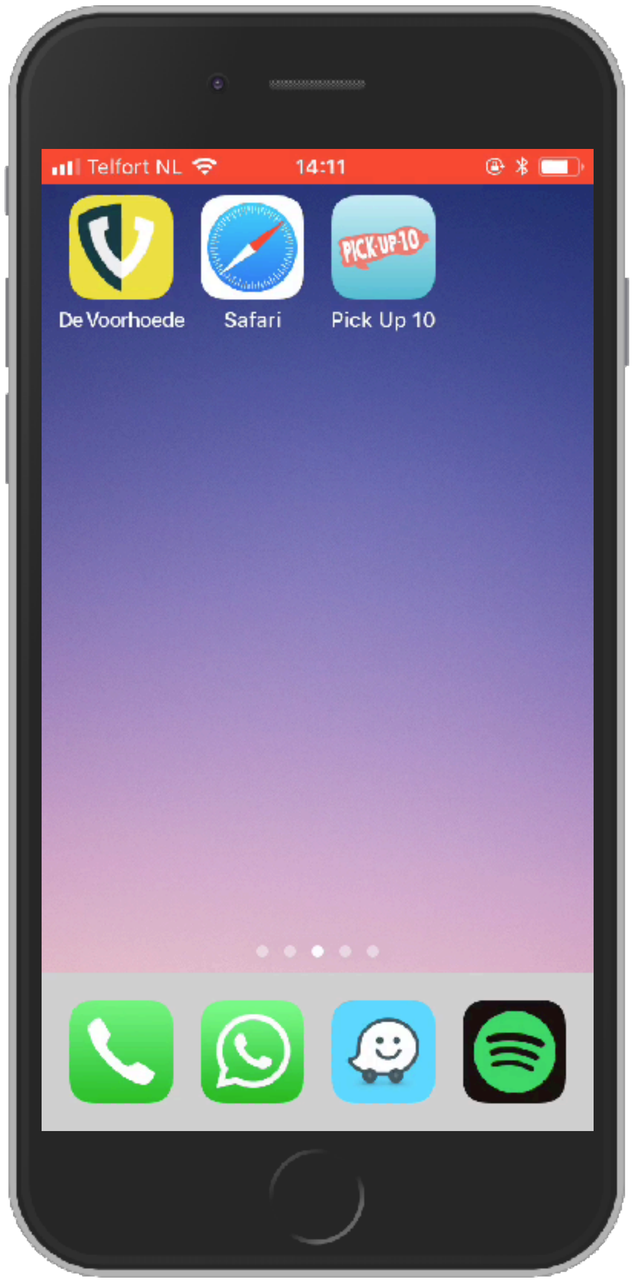
why serverless?

“focus on your app,

not your infrastructure”

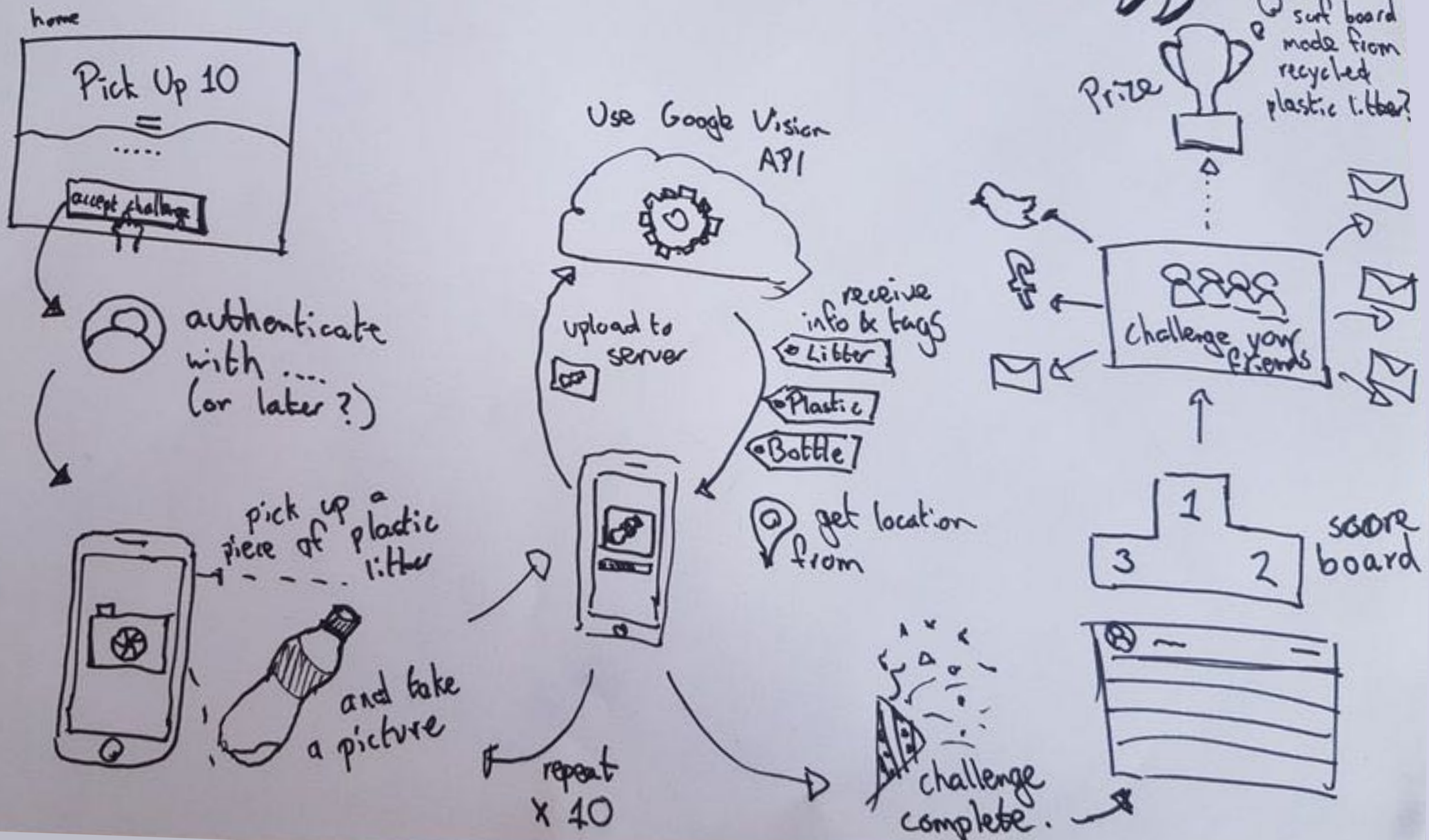
serverless.com

FOCUS ON YOUR APP



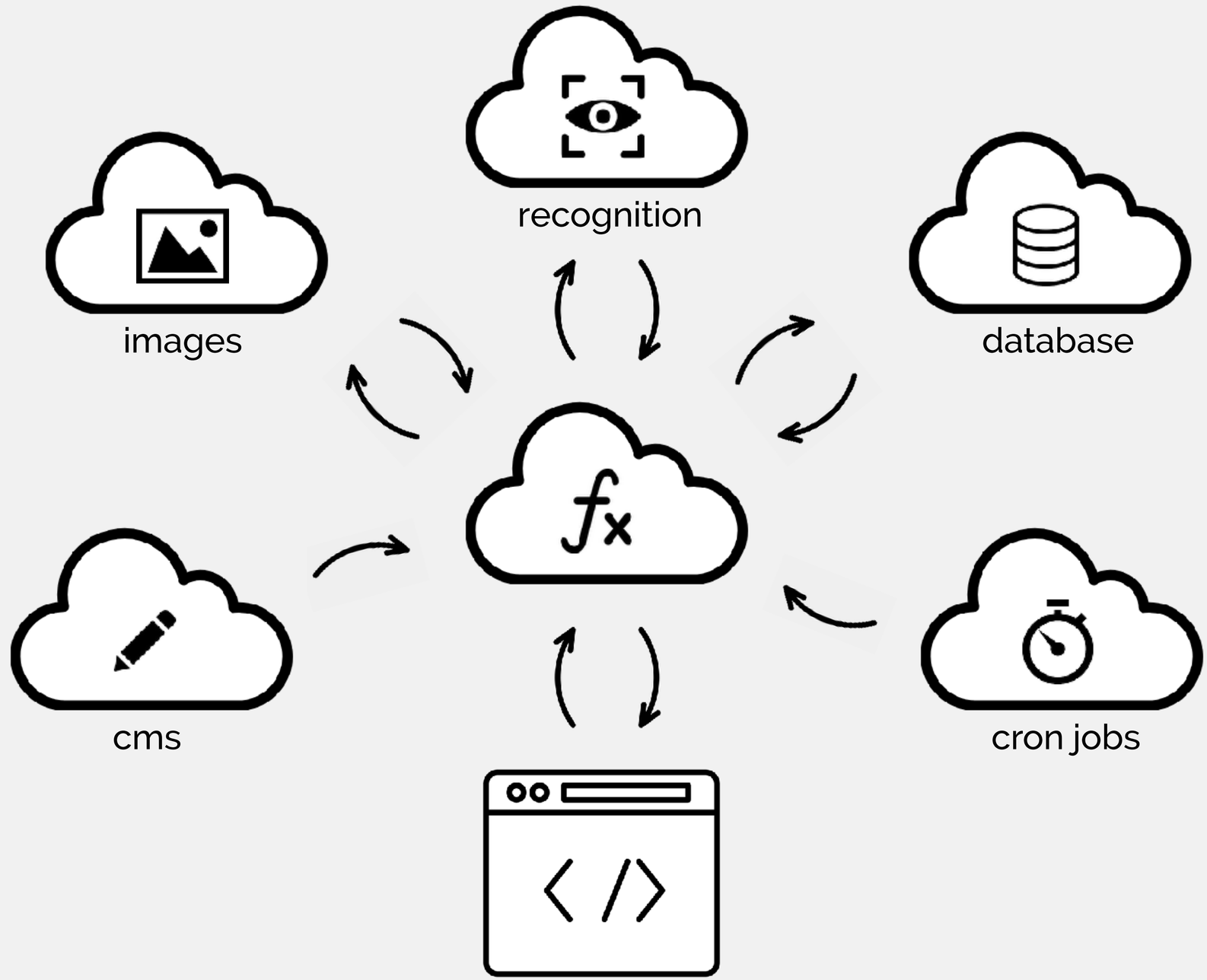
PWA

Pick Up 10


home
Pick Up 10
accept challenge
authenticate with ... (or later?)
pick up a piece of plastic litter
and take a picture
Use Google Vision API
upload to server
receive info & tags
Litter
Plastic
Bottle
get location from
repeat x 10
challenge complete.
Prize
surf board made from recycled plastic litter?
challenge your friends
1
3
2
score board

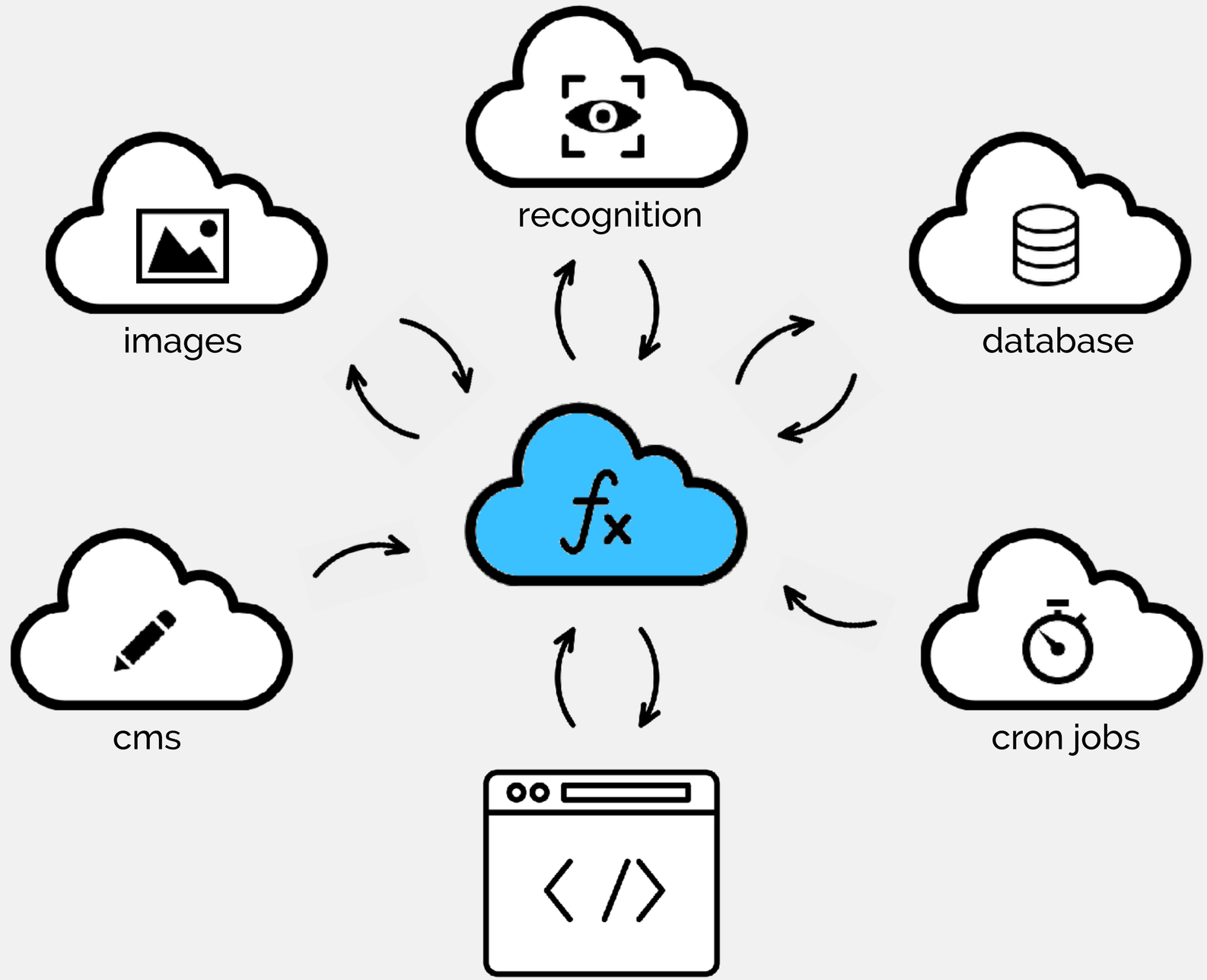
SIMPLIFY INFRASTRUCTURE

SERVERLESS #PICK UP 10



SERVERLESS CODE

SERVERLESS CODE



www.netlify.com

NETLIFY FUNCTION PATTERN

```
/**
 * filename: src/functions/hello.js
 * dev url : http://localhost:9000/hello?name=Meetup
 * prod url: https://my-project.netlify.com
 *                /.netlify/functions/hello?name=Meetup
 */

export const handler = (event, context, callback) => {
  const { name = 'World' } = event.queryStringParameters

  callback(null, {
    statusCode: 200,
    body: `Hello, ${name}`,
  })
}
```

www.netlify.com/docs/functions/

NETLIFY CONFIGURATION

```
// package.json:

"scripts": {
  "build": "NODE_ENV=production run-s build:*",
  "build:client": "nuxt generate",
  "build:server": "netlify-lambda build src/functions/",
  "dev": "run-p dev:*",
  "dev:client": "nuxt",
  "dev:server": "netlify-lambda serve src/functions/"
},

// netlify.toml:

[build]
  command = "npm run build"
  publish = "dist/client/"
  functions = "dist/functions/"
```

Secure https://app.netlify.com/sites/pick-up-10/functions

Jasper Moelker › pick-up-10

Overview Deploys Functions Identity Forms Split Testing Settings

Functions

Functions Free

6 Lambda functions actively running in **production**.

Search by branch or Deploy Preview

annotate-image.js
Created on Jun 12 (a month ago)

delete-pick.js
Created on Jun 11 (a month ago)

get-latest-stats.js
Created on Jun 11 (a month ago)

get-user.js
Created on Jun 11 (a month ago)

save-user.js
Created on Jun 11 (a month ago)

update-stats.js
Created on Jun 11 (a month ago)

Secure | https://app.netlify.com/sites/pick-up-10/functions/get-user

Jasper Moelker › pick-up-10

‹ Functions

Function get-user

Running in production.

Endpoint: https://www.pickup10.org/.netlify/functions/get-user

Preview deploy →

Function log

Copy to clipboard | Scroll to bottom

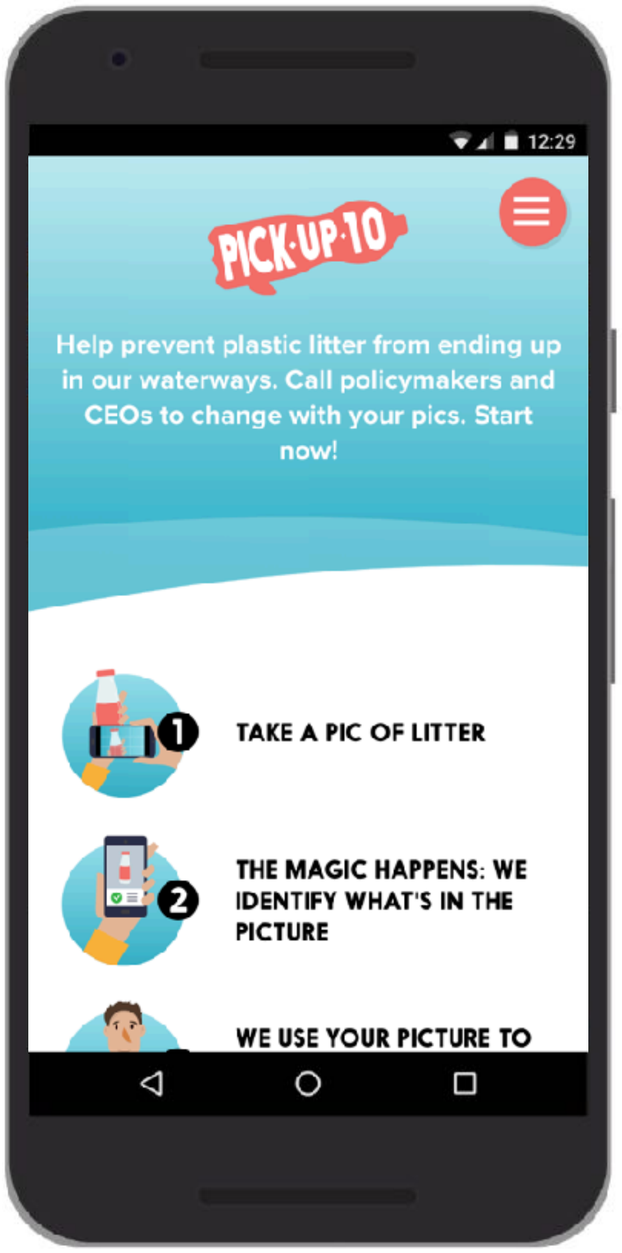
It's very **easy to get**

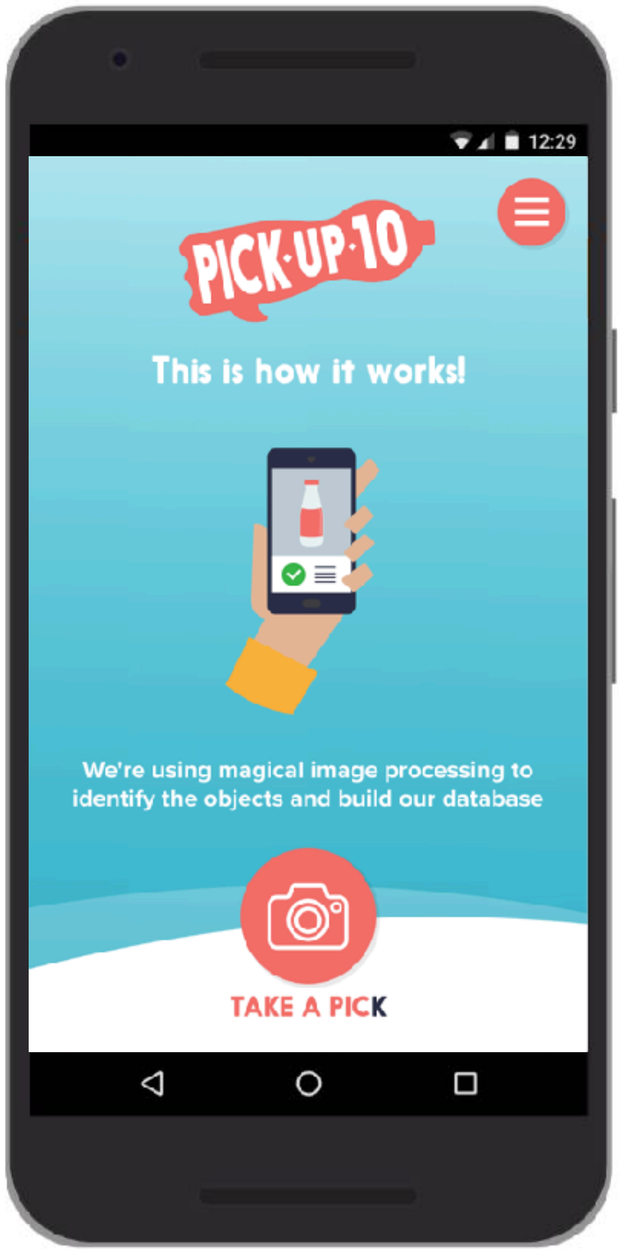
started with serverless

static sites & cloud functions

SERVERLESS IMAGES

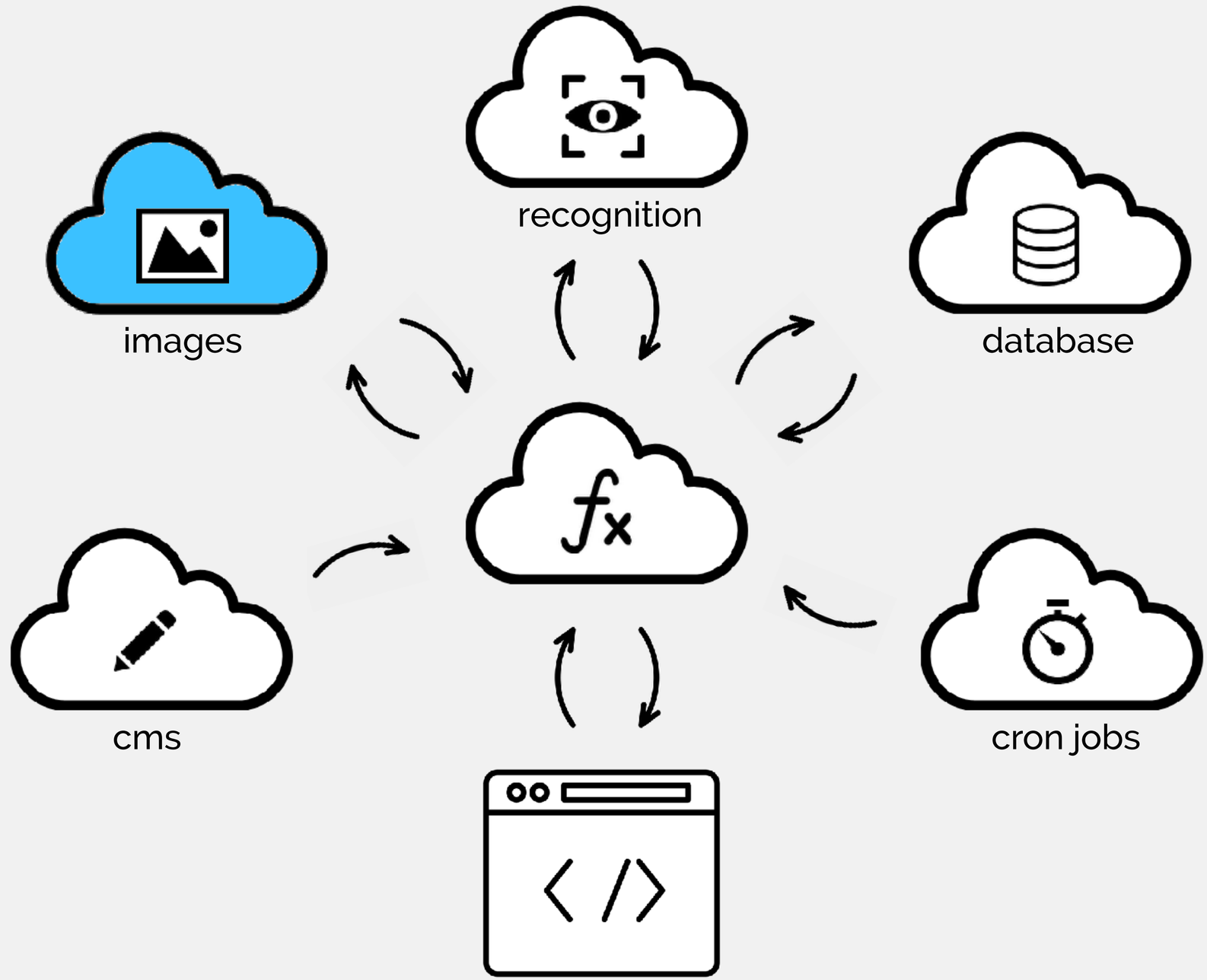
TAKE A PICK


12:29
PICK·UP·10
Help prevent plastic litter from ending up in our waterways. Call policymakers and CEOs to change with your pics. Start now!
1
TAKE A PIC OF LITTER
2
THE MAGIC HAPPENS: WE IDENTIFY WHAT'S IN THE PICTURE
WE USE YOUR PICTURE TO


12:29
PICK·UP·10
This is how it works!
We're using magical image processing to identify the objects and build our database
TAKE A PICK


A

SERVERLESS IMAGES



aws.amazon.com/s3

GET SIGNED IMAGE URL

```
const s3 = new S3({
  endpoint: 'https://s3.eu-central-1.amazonaws.com',
  signatureVersion: 'v4',
  region: AWS_DEFAULT_REGION,
  accessKeyId: AWS_ACCESS_KEY_ID,
  secretAccessKey: AWS_SECRET_ACCESS_KEY,
})
```

```
export const handler = (event, context, callback) => {
  const { ext, type } = JSON.parse(event.body)
  const key = `${uuid()}.${ext}`
  const signedUrl = s3.getSignedUrl('putObject', { Bucket:
'pick-up-10', Key: key, ContentType: type, Expires: 60 })

  callback(null, {
    statusCode: 200, body: JSON.stringify({ signedUrl, key })
  })
}
```

Failed to execute deploy:
[PUT /deploys/{deploy_id}/functions/{name}][422]
uploadDeployFunction default &{Code:0 Message:}

GET SIGNED IMAGE URL

```
const s3 = new S3({
  endpoint: 'https://s3.eu-central-1.amazonaws.com',
  signatureVersion: 'v4',
  region: AWS_DEFAULT_REGION,
  accessKeyId: AWS_ACCESS_KEY_ID,
  secretAccessKey: AWS_SECRET_ACCESS_KEY,
})
```

```
export const handler = (event, context, callback) => {
  const { ext, type } = JSON.parse(event.body)
  const key = `${uuid()}.${ext}`
  const signedUrl = s3.getSignedUrl('putObject', { Bucket:
'pick-up-10', Key: key, ContentType: type, Expires: 60 })

  callback(null, {
    statusCode: 200, body: JSON.stringify({ signedUrl, key })
  })
}
```

PREFIX AWS ENV VARIABLES

```
const s3 = new S3({
  endpoint: 'https://s3.eu-central-1.amazonaws.com',
  signatureVersion: 'v4',
  region: PU10_AWS_DEFAULT_REGION,
  accessKeyId: PU10_AWS_ACCESS_KEY_ID,
  secretAccessKey: PU10_AWS_SECRET_ACCESS_KEY,
})
```

```
export const handler = (event, context, callback) => {
  const { ext, type } = JSON.parse(event.body)
  const key = `${uuid()}.${ext}`
  const signedUrl = s3.getSignedUrl('putObject', { Bucket:
'pick-up-10', Key: key, ContentType: type, Expires: 60 })

  callback(null, {
    statusCode: 200, body: JSON.stringify({ signedUrl, key })
  })
}
```

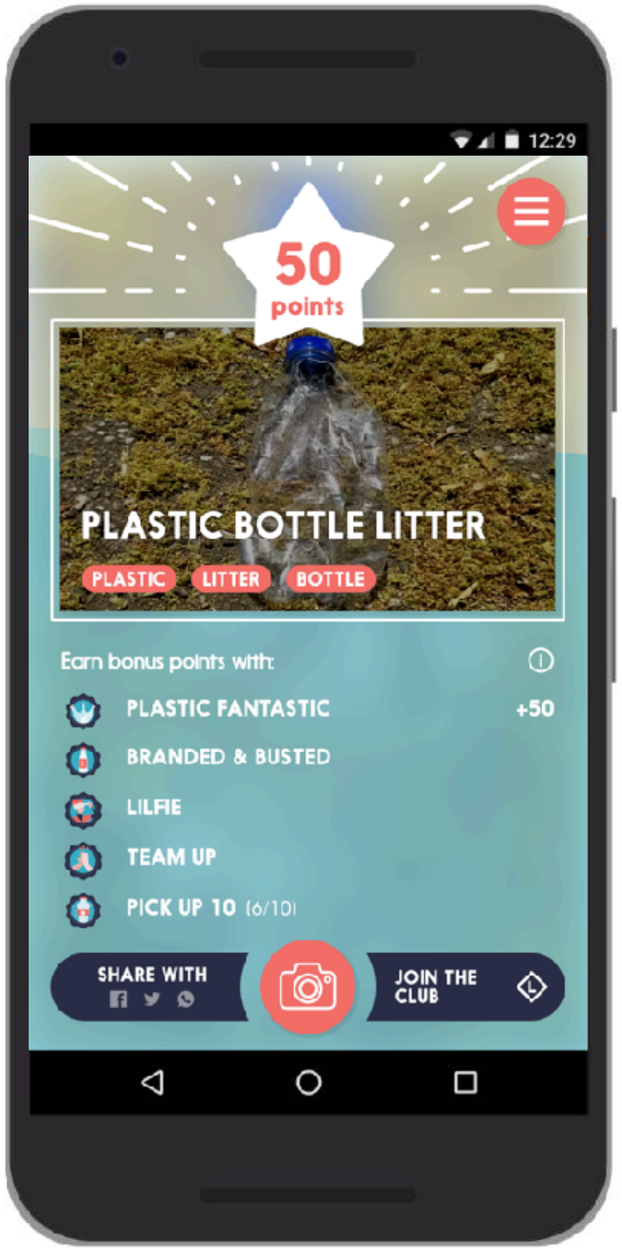
Netlify functions
are an **abstraction**
on top of AWS Lambda

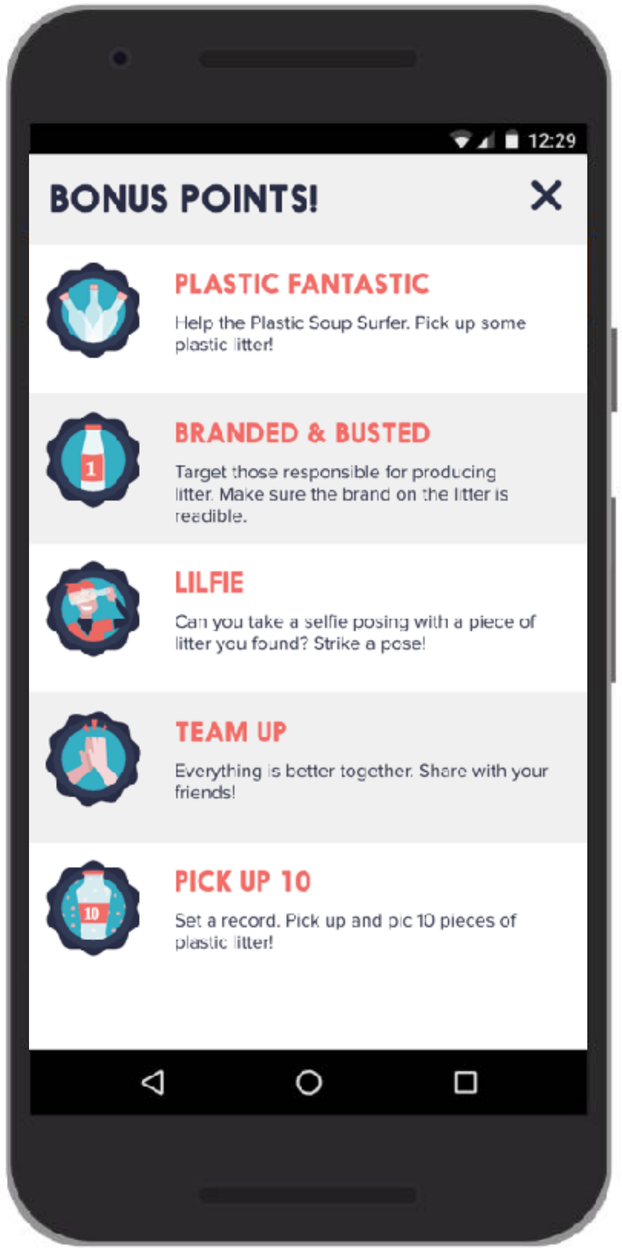
Cloudinary

cloudinary.com

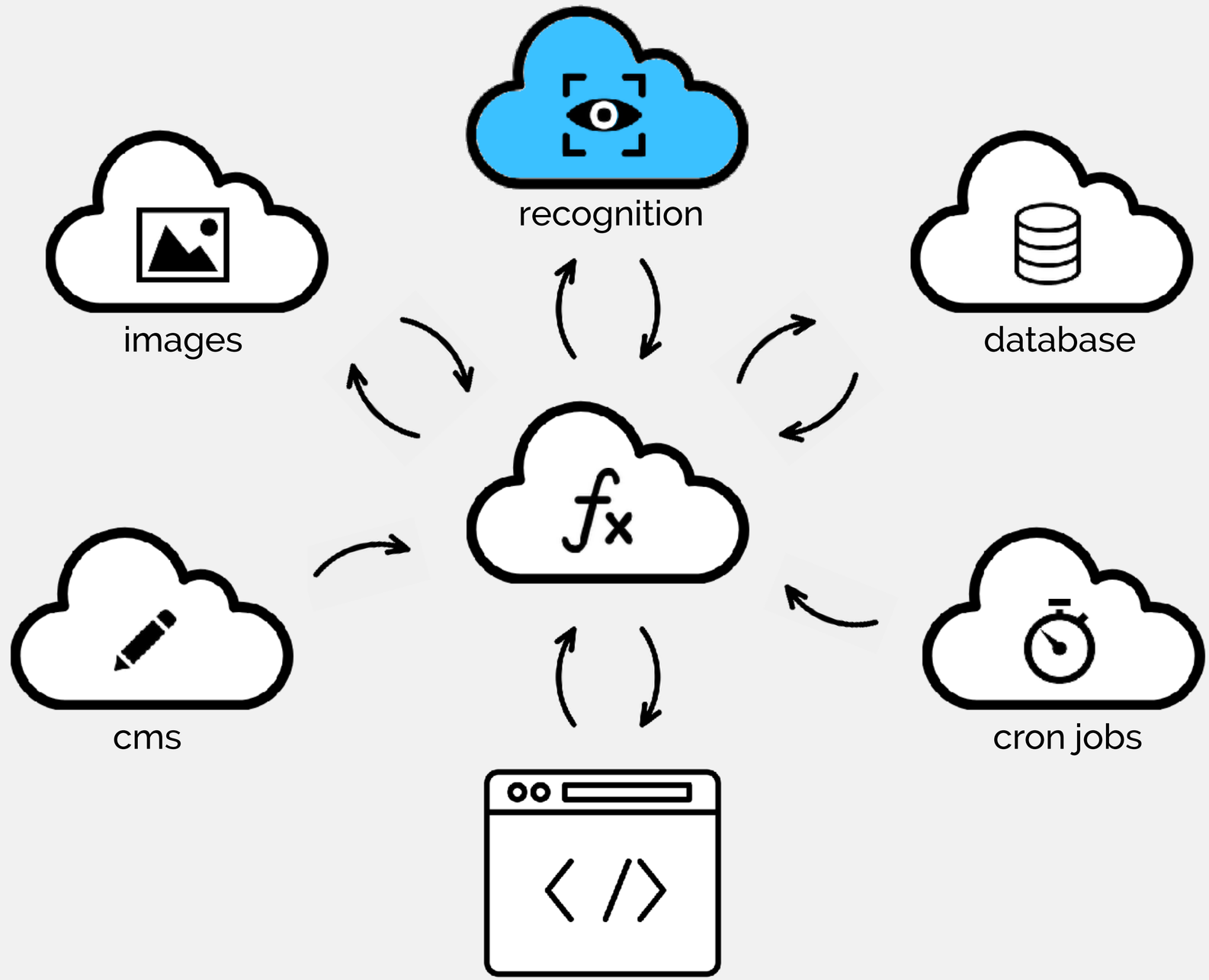
SERVERLESS
LITTER RECOGNITION

SCORE POINTS





SERVERLESS LITTER DETECTION



Cloud Vision API

cloud.google.com/vision

NODE.JS CLIENT

```
import vision from '@google-cloud/vision'

export const handler = (event, context, callback) => {
  const { imageUri } = event.queryStringParameters
  const client = new vision.ImageAnnotatorClient()

  Promise.all([
    client.faceDetection(imageUri),
    client.logoDetection(imageUri),
    client.webDetection(imageUri),
  ])
  .then(annotations => doMagic(annotations))
  .then(resData => callback(null, {
    statusCode: 200,
    body: JSON.stringify(resData)
  }))
}
```

VISION REST API

```
import request from ‘request-promise-native'

export const handler = ({ body }, context, callback) => {
  const { imageUri } = JSON.parse(body)
  request(`${VISION_API_URL}?key=${VISION_API_KEY}`, {
    method: 'POST', json: true, body: { requests: [{
        image: { source: { imageUri }, },
        features: [
          { type: 'FACE_DETECTION' },
          { type: 'LOGO_DETECTION' },
          { type: 'WEB_DETECTION' },
        ]
  }]})
  .then(annotations => doMagic(annotations))
  .then(resData => callback(null, {
    statusCode: 200, body: JSON.stringify(resData)
  }))
}
```

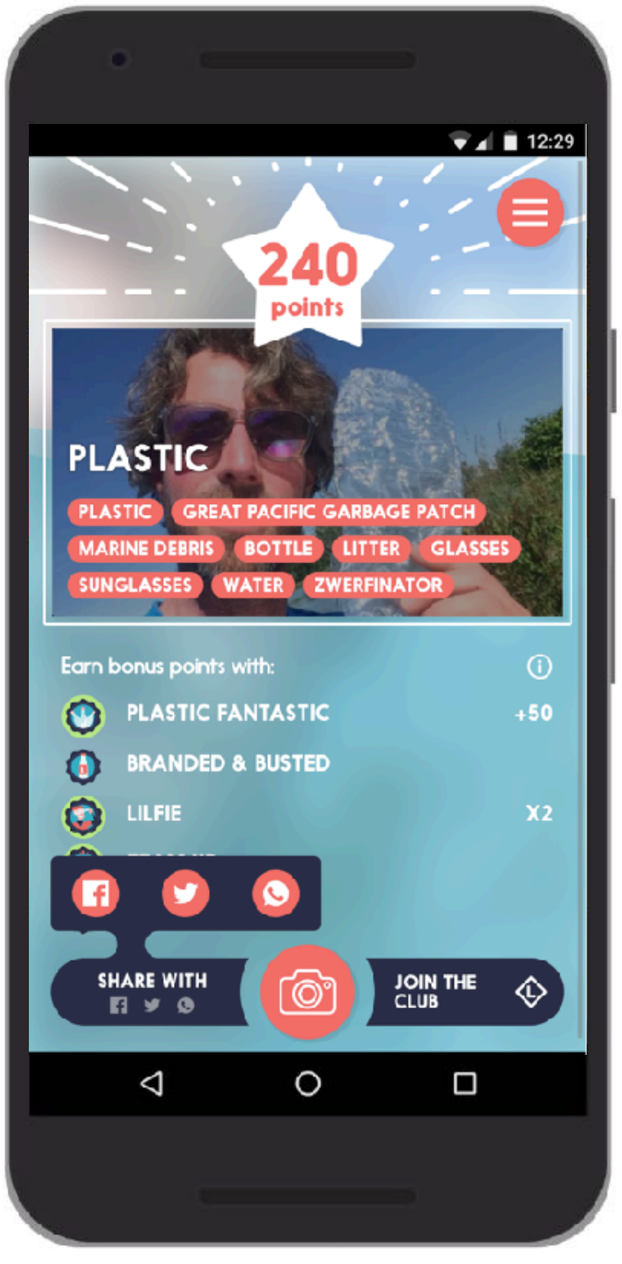
inside cloud functions

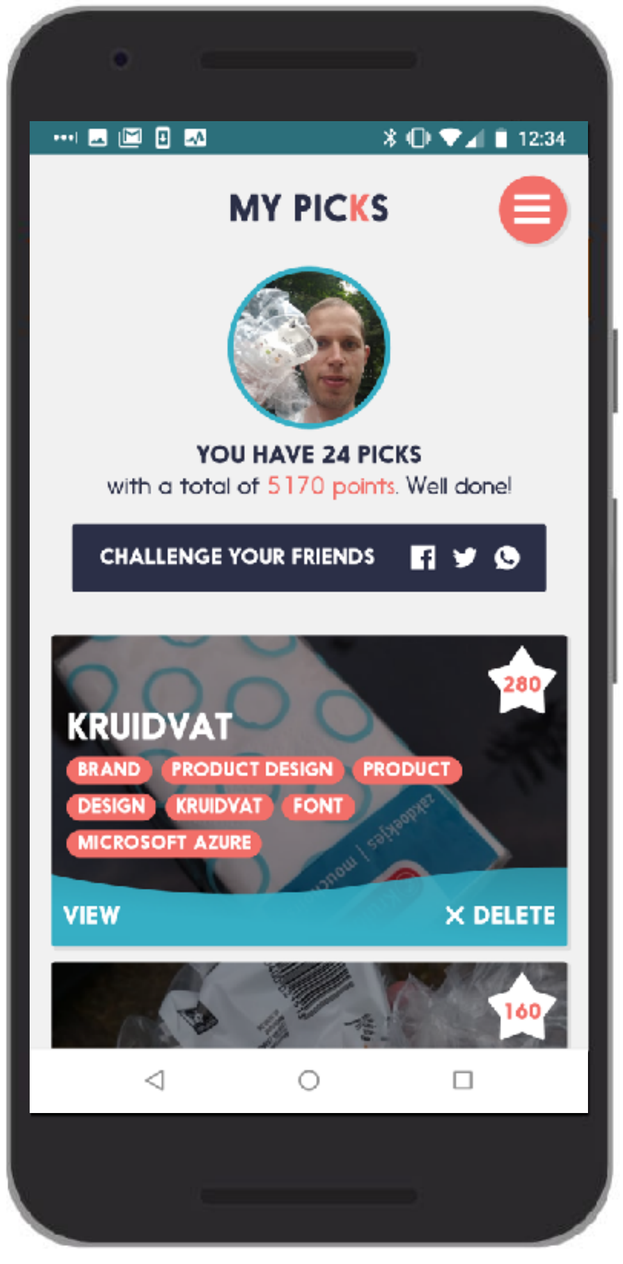
avoid native modules

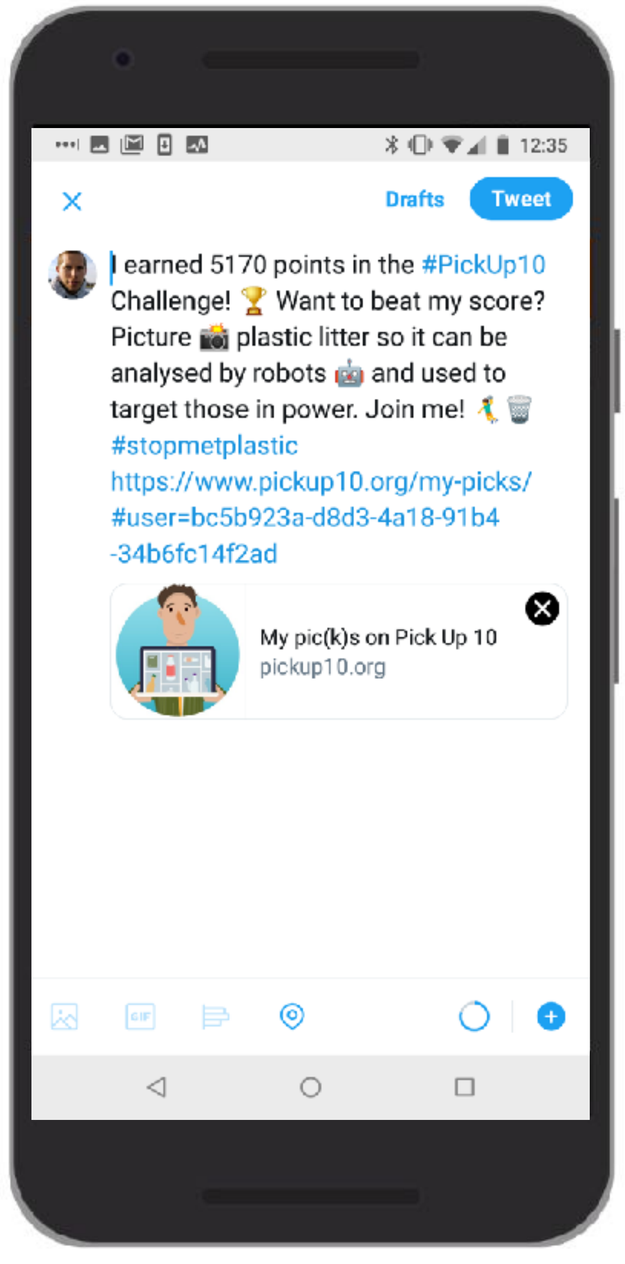
HTTP APIs are often easier

SERVERLESS DATABASE

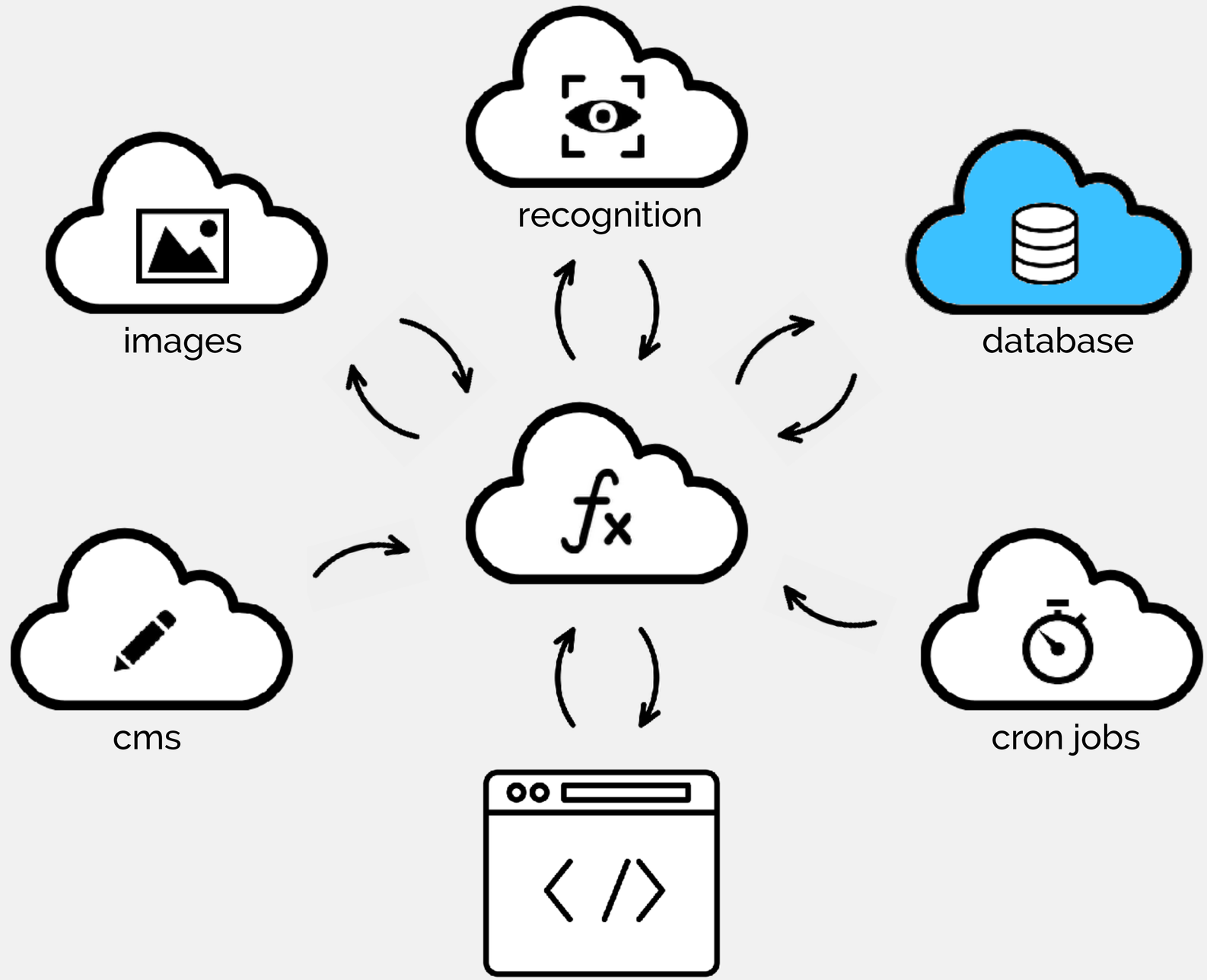
MY PICKS


12:29
240
points
PLASTIC
PLASTIC
GREAT PACIFIC GARBAGE PATCH
MARINE DEBRIS
BOTTLE
LITTER
GLASSES
SUNGLASSES
WATER
ZWERFINATOR
Earn bonus points with:
PLASTIC FANTASTIC
+50
BRANDED & BUSTED
LILFIE
X2
SHARE WITH
JOIN THE CLUB


12:34
MY PICKS
YOU HAVE 24 PICKS
with a total of 5170 points. Well done!
CHALLENGE YOUR FRIENDS
280
KRUIDVAT
BRAND
PRODUCT DESIGN
PRODUCT
DESIGN
KRUIDVAT
FONT
MICROSOFT AZURE
VIEW
DELETE
160


12:35
Drafts
Tweet
I earned 5170 points in the #PickUp10 Challenge! 🏆 Want to beat my score? Picture 📸 plastic litter so it can be analysed by robots 🤖 and used to target those in power. Join me! 🏌️ 🗑️
#stopmetplastic
https://www.pickup10.org/my-picks/#user=bc5b923a-d8d3-4a18-91b4-34b6fc14f2ad
My pic(k)s on Pick Up 10
pickup10.org

SERVERLESS #PICK UP 10



mlab.com

USE MONGODB DRIVER

```
import mongoose from ‘mongoose'

mongoose.connect(`mongodb://${MLAB_USER}:${MLAB_PASSWORD}
@ds318769.mlab.com:47479/pick-up-10`)

const Users = mongoose.model('Users', mongoose.Schema({
  publicKey: String, privateKey: String, picks: []
}))

export const handler = (event, context, callback) => {
  const { publicKey } = event.queryStringParameters
  Users.findOne({ publicKey })
    .then(result => result.toObject())
    .then(({ picks }) => callback(null, {
      statusCode: 200,
      body: JSON.stringify({ publicKey, picks })
    }))
}
```

Secure | https://app.netlify.com/sites/pick-up-10/functions/get-user

Jasper Moelker › pick-up-10

‹ Functions

Function get-user

Running in production.

Endpoint: https://www.pickup10.org/.netlify/functions/get-user

Preview deploy →

Function log

Copy to clipboard | Scroll to bottom

mLab Data API

mLab databases can be accessed by applications in two ways.

The first method—the one we **strongly** recommend whenever possible for added performance and functionality—is to connect using one of the available MongoDB drivers. You do not need to use our API if you use the driver.

The second method, documented in this article, is to connect via mLab's RESTful Data API. **Use this method only if you cannot connect using a MongoDB driver.**

WARNING

Your API key will give full access to all data within the databases belonging to your mLab account. If you distribute it to untrusted individuals, they can gain access to your account and your data.

We have seen customers distribute mobile and AJAX-based web applications to their end users. Doing this will expose your account to attack. Such clients are appropriate only when you can control their distribution to just mLab account users.

Here's an example of a complete Resource URL:

```
https://api.mlab.com/api/1/databases?apiKey=2E81PUmPFI84t7UIc_5YdldAp1ruUPKye
```

docs.mlab.com/data-api/

mLab DOCS + SUPPORT

mLab Data API

Insert multiple documents

To add multiple documents to the specified collection, specify a list of documents in the data payload:

jQuery reference

```
POST /databases/{database}/collections/{collection}
Content-Type: application/json
Body: <JSON data>

Example (using jQuery):

$.ajax( { url: "https://api.mlab.com/api/1/databases/my-db/collections/my-coll?apiKey=myAPIKey",
                  data: JSON.stringify( [ { "x" : 1 }, { "x" : 2 }, { "x" : 3 } ] ),
                  type: "POST",
                  contentType: "application/json" } );
```

Your API key, including any clients from which your API key can be recovered, should not be distributed to non-administrators.

docs.mlab.com/data-api/#insert-document

Build environment variables

CLOUDINARY_API_KEY	243423892894234
CLOUDINARY_API_SECRET	2CRDT4EagiaMxb1amfyC9uQc...
CLOUDINARY_API_URL	https://api.cloudinary.com/v1_1/...
DATO_API_TOKEN	ymjuxwsjx6ml0at7en2hqby959...
MLAB_API_KEY	gOunGGEKFPk1RsOwDUqLR2v...
MLAB_API_URL	https://api.mlab.com/api/1/data...
MLAB_PASSWORD	lqvch7kthzeglq3b9
MLAB_USER	pu10
NODE_ENV	production
PU10_AWS_ACCESS_KEY_ID	F9OI5BULUXXZW6PS3KI6
PU10_AWS_DEFAULT_REGION	eu-central-1
PU10_AWS_SECRET_ACCESS_KEY	1HGyq57Io3QsAc9DiS09PSX2l...
STATS_UPDATE_TOKEN	j0TQVTS0M7T9X4holPKSEqR4...
VISION_API_KEY	jUT5KDPZmxvNlGcM7yd6n1B8...
VISION_API_URL	https://vision.googleapis.com/v...

Edit variables

MLAB REST API

```
import request from 'request-promise-native'

export const handler = (event, context, callback) => {
  const { publicKey } = event.queryStringParameters
  const queryString = JSON.stringify({ publicKey })

  const uri = `${MLAB_API_URL}/collections/users
    ?fo=true&q=${queryString}&apiKey=${MLAB_API_KEY}`

  request({ uri, json: true })
    .then(({ picks }) => callback(null, {
      statusCode: 200,
      body: JSON.stringify({ publicKey, picks })
    }))
}
```

cloud functions **keep**

sensitive data server-side

SERVERLESS CRON JOBS

STATS & DATA

Loading stats...

0
PLASTICS PICKED UP

0
BRANDS IDENTIFIED

0
LILFIES TAKEN

We last counted today

Goals

Your picks are data, proof, numbers on (plastic) litter to confront policymakers and company CEOs. These are our goals:

1 **10.000 PLASTIC PICKS**
we'll demand anti-litter campaigns from multionationals

www.pickup10.org/stats-and-data

STATS & DATA

Loading stats...

0
PLASTICS PICKED UP

0
BRANDS IDENTIFIED

0
LILFIES TAKEN

We last counted today

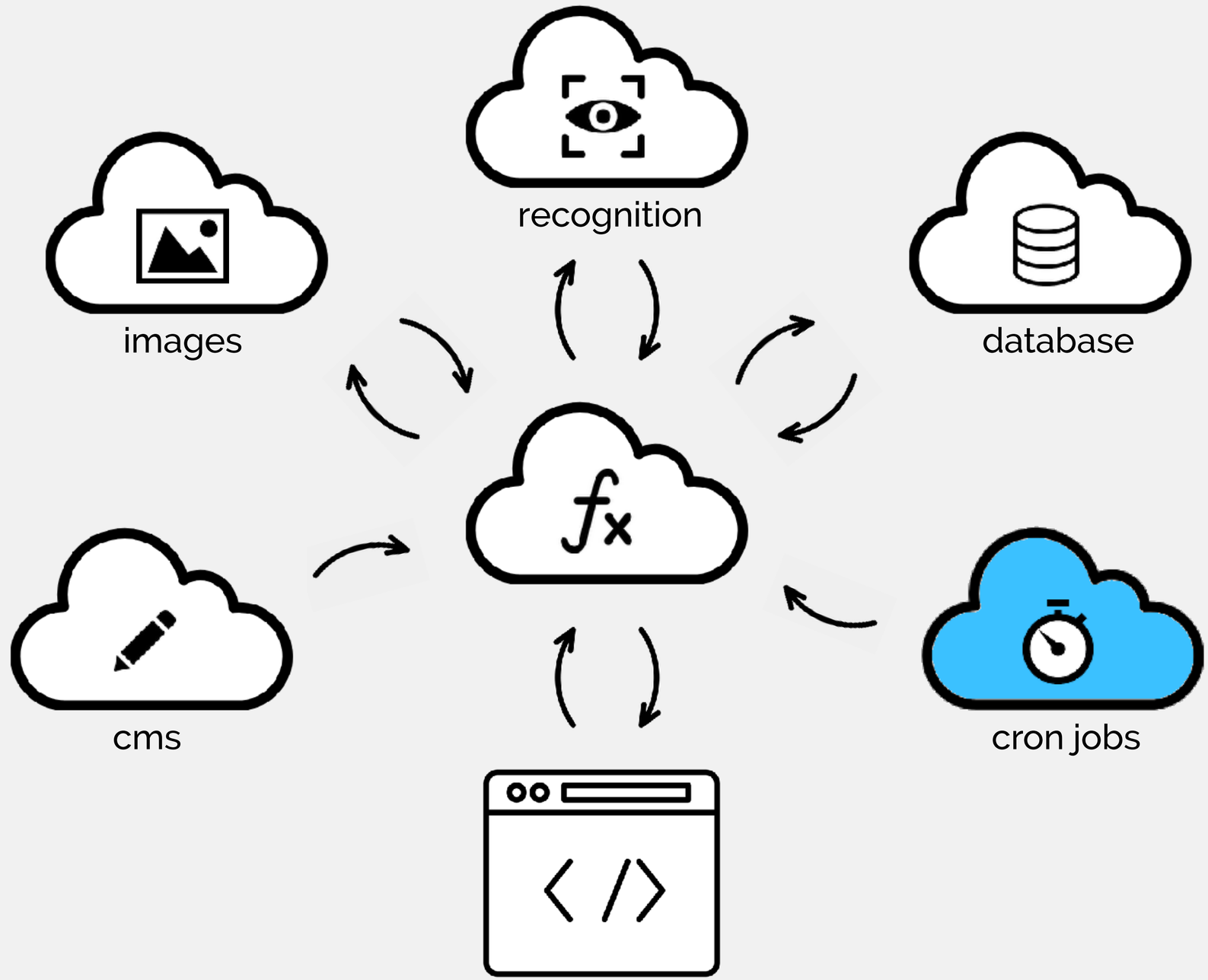
Goals

Your picks are data, proof, numbers on (plastic) litter to confront policymakers and company CEOs. These are our goals:

1 **10.000 PLASTIC PICKS**
we'll demand anti-litter campaigns from multionationals

www.pickup10.org/stats-and-data

SERVERLESS CRON JOBS



webtask.io

Secure https://webtask.io/make

webtask

Extend

Scheduler

Refresh | Remove schedule

This cron is active

Time Zone: Europe/Amsterdam

Next run will be today at 18:00:00

Run this every 10 mins

Write an advanced schedule

00	*/1	*	*	*
Minute	Hour	Day	Month	Weekday

Update schedule

Stats

Run count: 722

Error count: 8

Last Results

View All

Completed 42 minutes ago

pick-up-10-stats-updater

```
1  const request = require('request')
2
3  /**
4  * @param context {WebtaskContext}
5  */
6  module.exports = function(context, cb) {
7    const token = context.secrets.PU10_UPDATE_STATS_TOKEN
8    request({
9      method: 'POST',
10     uri: 'https://www.pickup10.org/.netlify/functions/update-stats',
11     json: true,
12     body: { token },
13   }, (err, res, body) => {
14     if (err) {
15       cb(err)
16     } else {
17       cb(null, body)
18     }
19   })
20 }
```

find the best service

for the job

SERVERLE$$

NETLIFY FUNCTIONS

Settings for pick-up-10

www.pickup10.org

Deploys from GitHub. Owned by Jasper Moelker

Last update on Jun 20 (20 days ago)

- General
- Build & deploy
- Domain management
- **Functions**
  - AWS Lambda
- Identity
- Forms
- Access control

AWS Lambda

Settings to build and deploy your AWS Lambda functions

Usage

Last update today at 8:35 AM **Functions Free**

Requests
Counts every time a function was invoked in the current billing period.
10,496/125,000
114,504 requests left

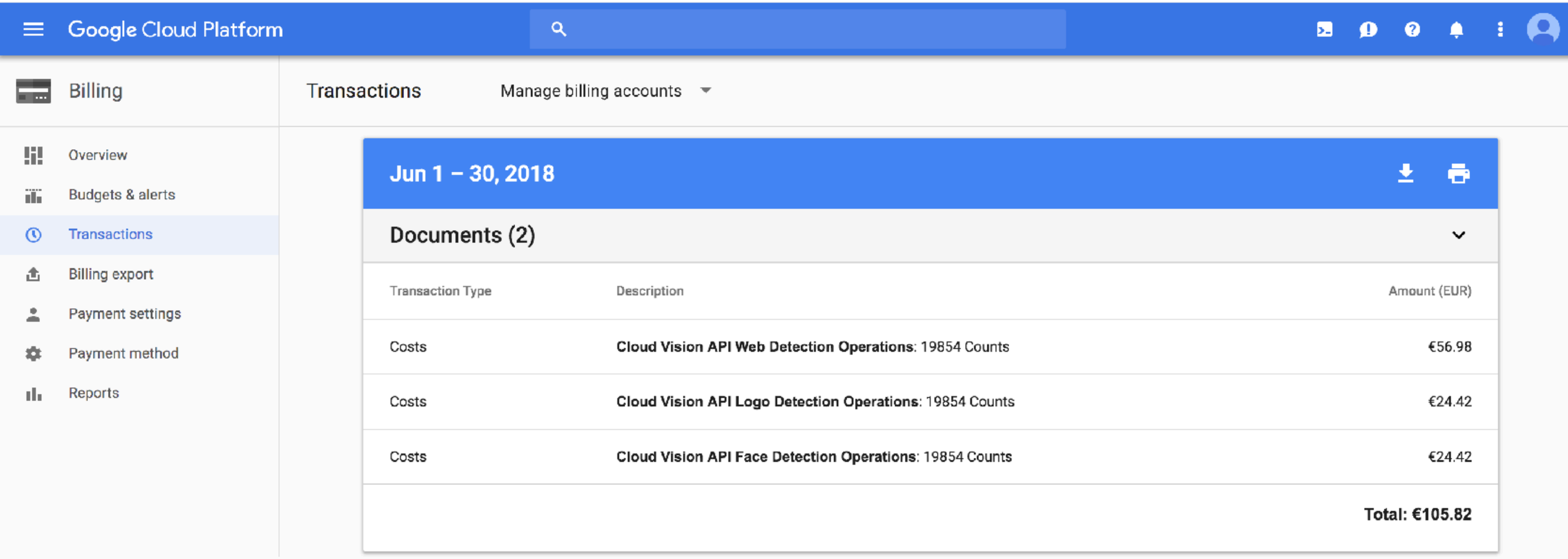
Run time
Combined run time of all function requests in the current billing period.
5 hours/100 hours
95 hours left

Scalable: Your plan will upgrade automatically to fit your usage.

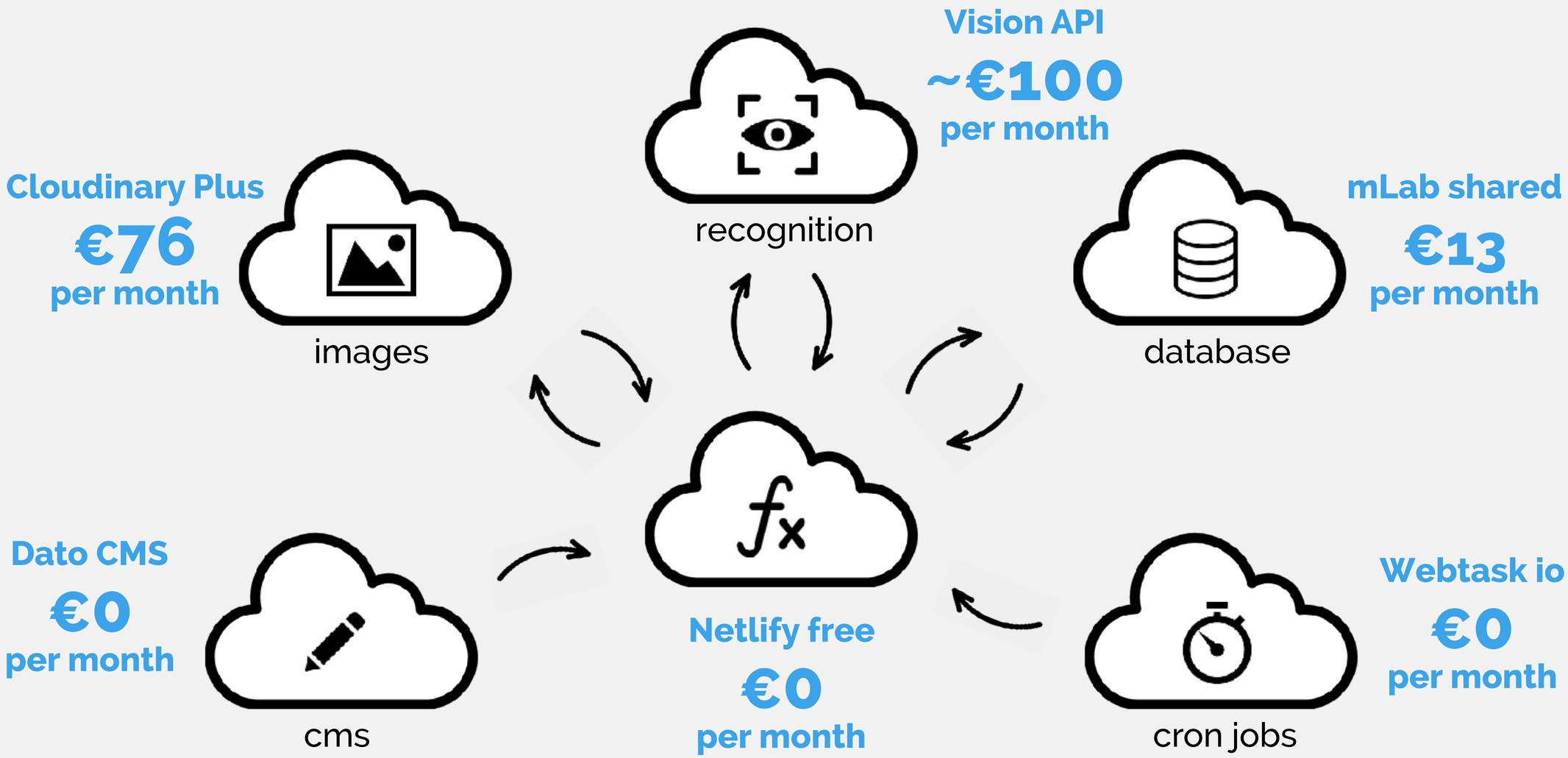
Learn more about pricing and usage →

Change plan | Billing and payment

GOOGLE VISION API



SERVERLE$$ #PICK UP 10



total
~€200
per month

costs are based on usage

cheaper than rolling your own

TAKEAWAYS

use the best cloud services

keep cloud functions **simple**

deploy, sit back, **profit**

THANK YOU

DE VOORHOEDE

front-end developers