



# Managing High Load on Low Budget

Máté Marjai - 2018, Moscow  
Big Data Days



# What's the reason?

Building a product from the ground up

Totally bootstrapped

Product is about payments, helping teams to do their integrations quicker and easier



# Payments?

Payments is broken

Payments suck at good & useful user messaging

A single integration takes weeks for a team

Payments integrations are never finished

And to top it, all API sandboxes are...





# What's wrong with Sandboxes?

They are...

not available.

not the same as the real API.

restricted with trivial test cases.

What can one do about all this?





# Fix it!

Started with in-house mocks & sims

Prototyped it in small environments

Developed it into a product - Testing Pays



# Step 1: Build the prototype

Limited resources and time

We chose Erlang due to familiarity & endurance

Covered our basic feature set

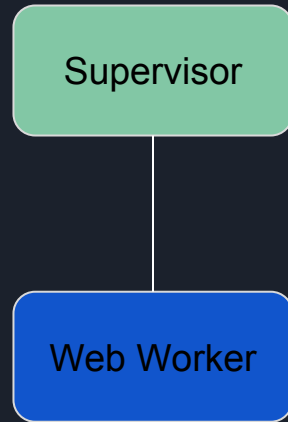
Can handle up to 5 requests per second

Average response time > 200ms





# Sim API Node





# Success!

We proved we can do it

It can run on hobby heroku, AWS micro instance, etc.

It can run on a Raspberry PI (spoiler alert - super slow)

Time to improve!



## Step 2: Improve

Introduced more workers and revised supervision tree

Added customer facing logs & reports

Goals:

- Better performance, with an eye on load testing
- Show live information to our users
- Show reports about test activity

# The Goal

Q SEARCH SIMS TestingPays CURRENT TEAM My Lovely Horse MY ACCOUNT Máté Z Logout

reallex payments **4** REQUESTS FOR THIS SIM **2.08%** RESPONSE COVERAGE

/auth (v1)

1. INTEGRATE **2. DEBUG** 3. TEST 4. REPORTS SETTINGS LOAD TEST 3D SECURE FLOW ?

**Debug your integration**  
View your requests as they are processed by the sim - easily debug the data you send in your test transactions.

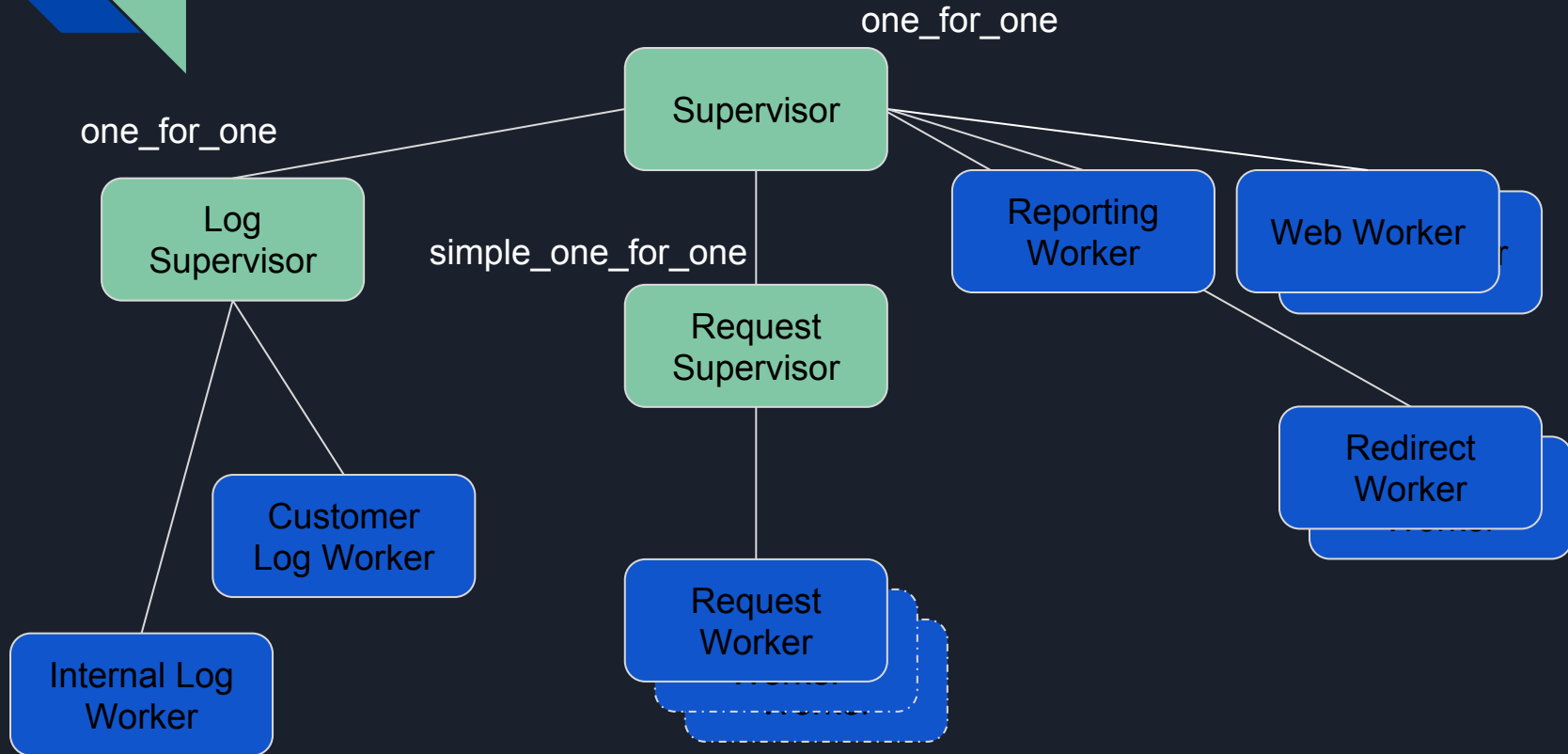
Start Live Tail to view real-time logging of your test transaction. Download log files to view older test transactions

[Play Test Request](#)

[Fullscreen](#) [Pause](#) [Clear](#) [Stop Live Tail](#)

```
[2018-03-07 14:26:40.132303] [8d125cc6-2213-11e8-8542-0ad772b8af92] API Sim Enabled for Processor reallex auth
[2018-03-07 14:26:40.156445] [8d125cc6-2213-11e8-8542-0ad772b8af92] API Sim Enabled for Processor reallex auth
[2018-03-07 14:26:40.156536] [8d125cc6-2213-11e8-8542-0ad772b8af92] Sim Limits Check called
[2018-03-07 14:26:40.157624] [8d125cc6-2213-11e8-8542-0ad772b8af92] Sim request limits: 300; Current count in billing period 3 auth
[2018-03-07 14:26:40.159130] [8d125cc6-2213-11e8-8542-0ad772b8af92] Request Validation called
[2018-03-07 14:26:40.159212] [8d125cc6-2213-11e8-8542-0ad772b8af92] Validation result: passed
[2018-03-07 14:26:40.159892] [8d125cc6-2213-11e8-8542-0ad772b8af92] Response mapping key found: 00
[2018-03-07 14:26:40.161472] [8d125cc6-2213-11e8-8542-0ad772b8af92] Assemble Response called with response code: 200:00_success:0
[2018-03-07 14:26:40.177597] [8d125cc6-2213-11e8-8542-0ad772b8af92] Rendering Realex template with parameters {"timestamp":"20180307142640","issuer_bank_name":"Bank of Testing Pays","issuer_bank_country":"Ireland","issuer_bank_country_code":"IE","issuer_bank_region":"EUR","account":"Internet","time_taken":"1","auth_time_taken":"1","post_code_check":"U","address_che
2213-11e8-8542-0ad772b8af92","authcode":"","00","batchid":"","1","cvv":"","M","message":"Authorized by
TP","result_code":"","00","pasref":"abcd1234123412","three_d_url":"https://api.testingpays.com/e013a6702cc366f69a5954bf625b97ea/3d_secure/realex/auth/3ds_render","sha1hash":"","e91c2ec00e6cc00
```

# Sim API Node





# More Success!

95% of requests run < 20ms (with 70% < 15ms)

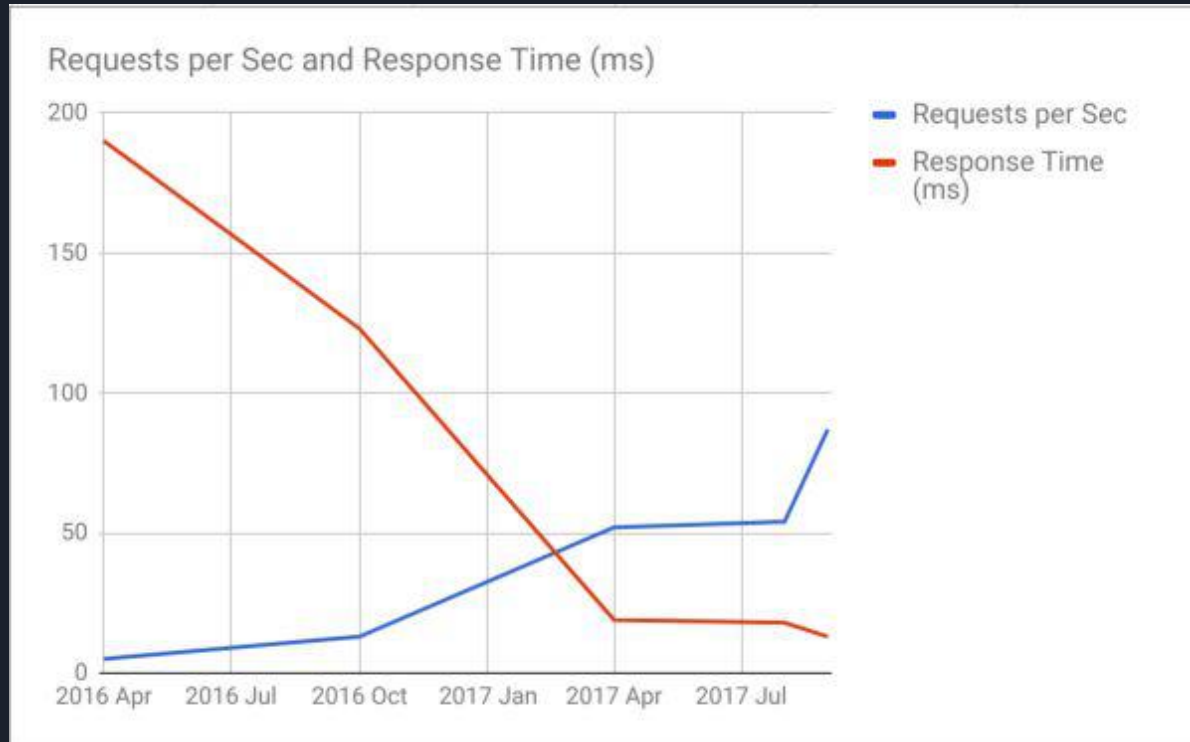
We can reach 100 requests per second

Can handle over 70 requests per second under load

Still running on the smallest possible instances\*

\*more about that later

# Graphs!





# More Success!

Runs better on the RPI than our initial version\*

- 20 requests per second
- Average response time < 60ms

\* casing can get hot though!





# What about Elixir?

We use Elixir for messaging and reporting:

- Live Log
- Request Counters
- Show stats and usage



# What about Elixir?

It's a simple Phoenix app with channels

Connects to a Redis cluster

Super visors?!?!?





# Supervisors!

Supervisors are supposed to be processes which do nothing but make sure their children are restarted when they die\*

\*Unless the desired behaviour is to just let them disappear.



# Sup Flags

- Strategy
  - one\_for\_one
  - one\_for\_all
  - rest\_for\_one
  - simple\_one\_for\_one
- Intensity
- Period

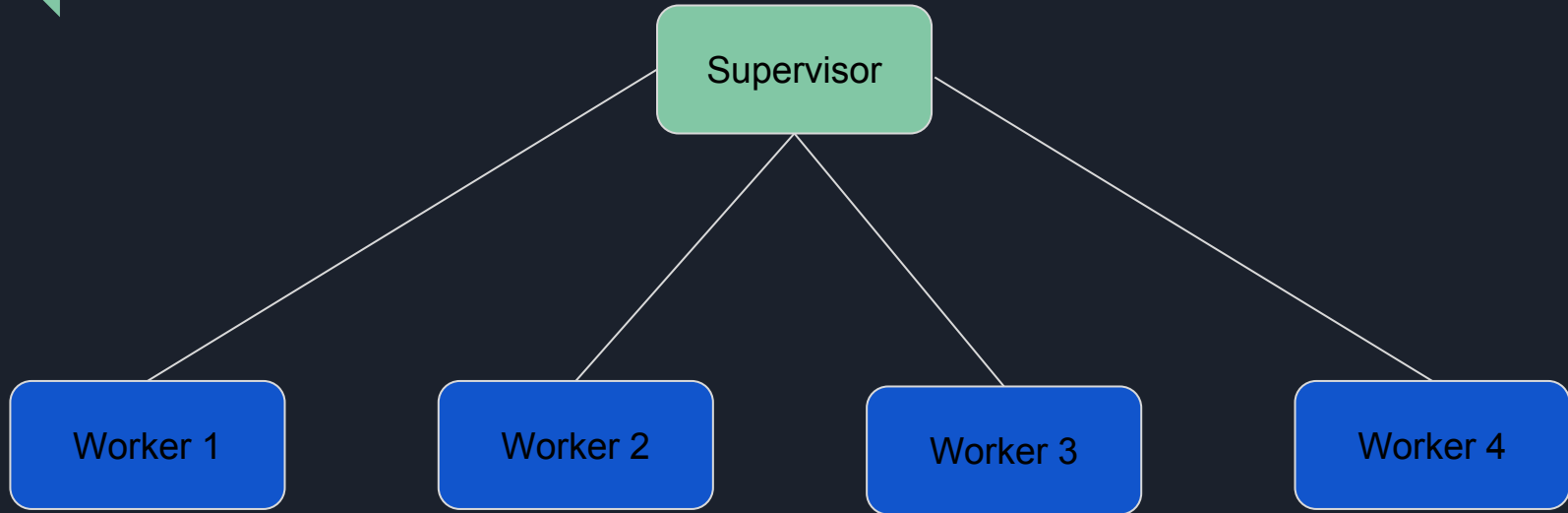


# Child Specs

- Id
- Start
- Restart
- Shutdown
- Type
- Modules

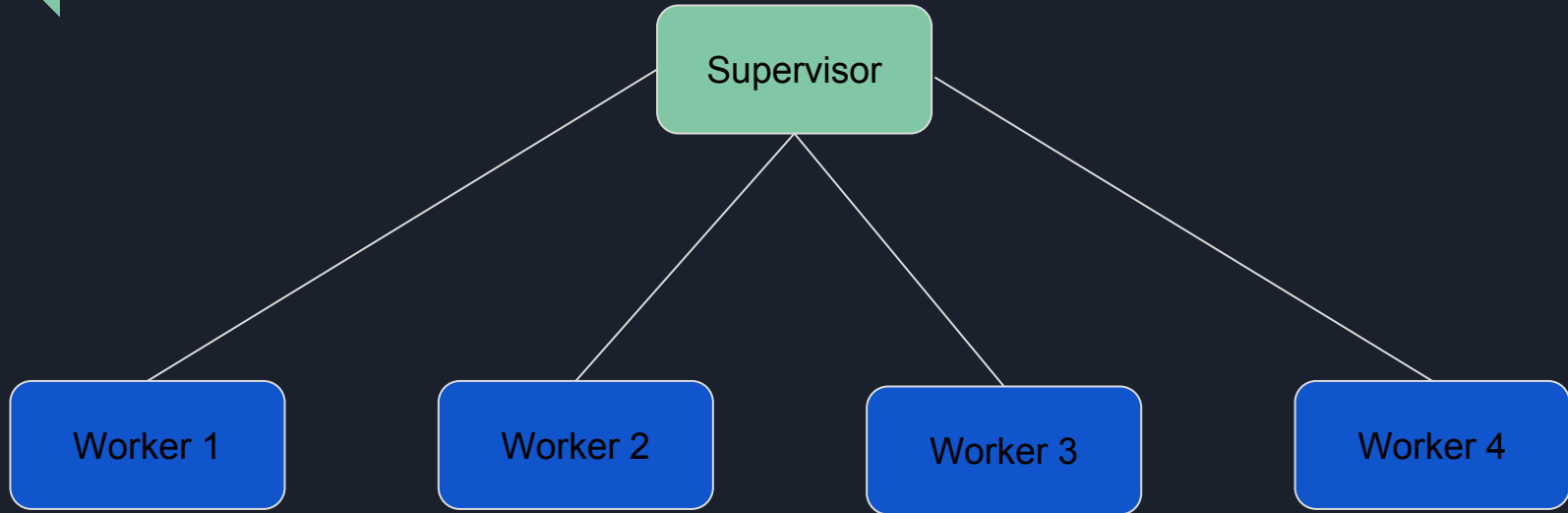


one\_for\_one





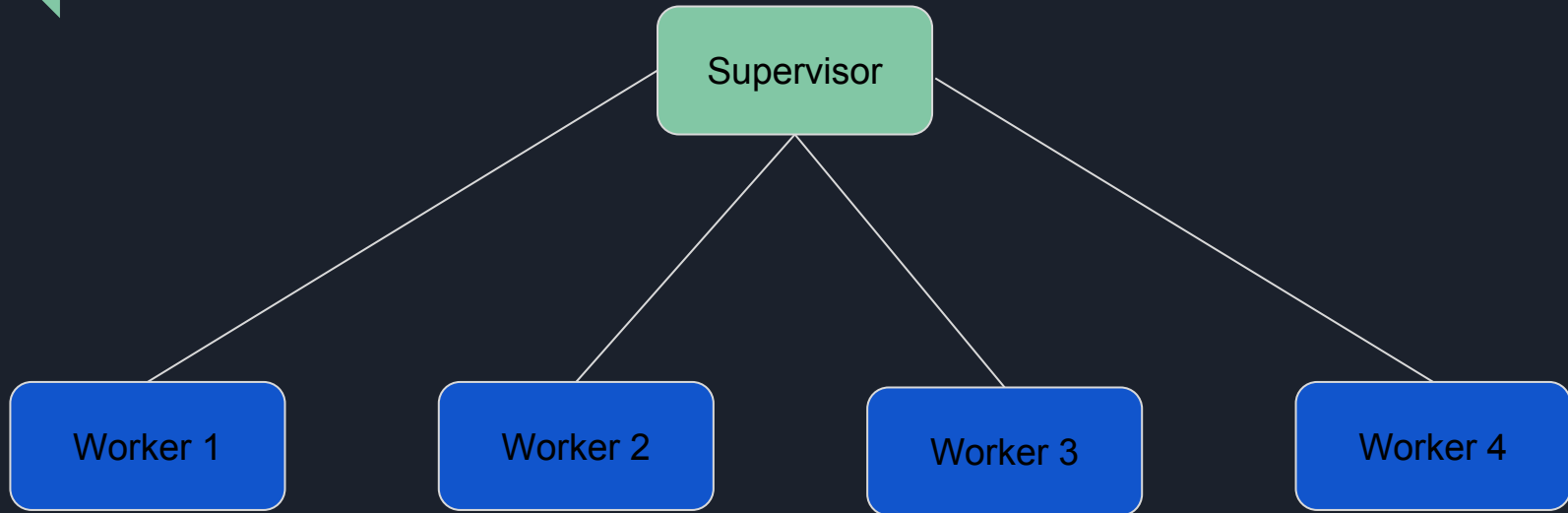
one\_for\_all



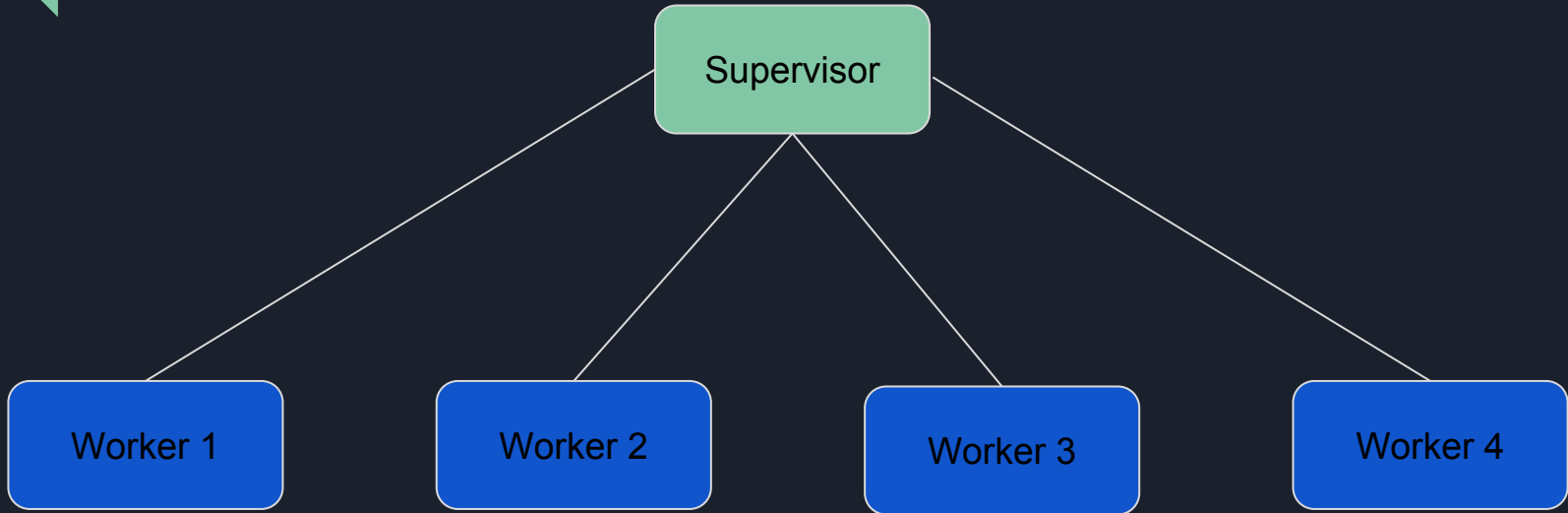




# rest\_for\_one



# simple\_one\_for\_one



This is great, what about budgets?





Thank You!