At Scale With Style

Erlang <3 <3 <3 Ruby

Martin Rehfeld, @klickmich

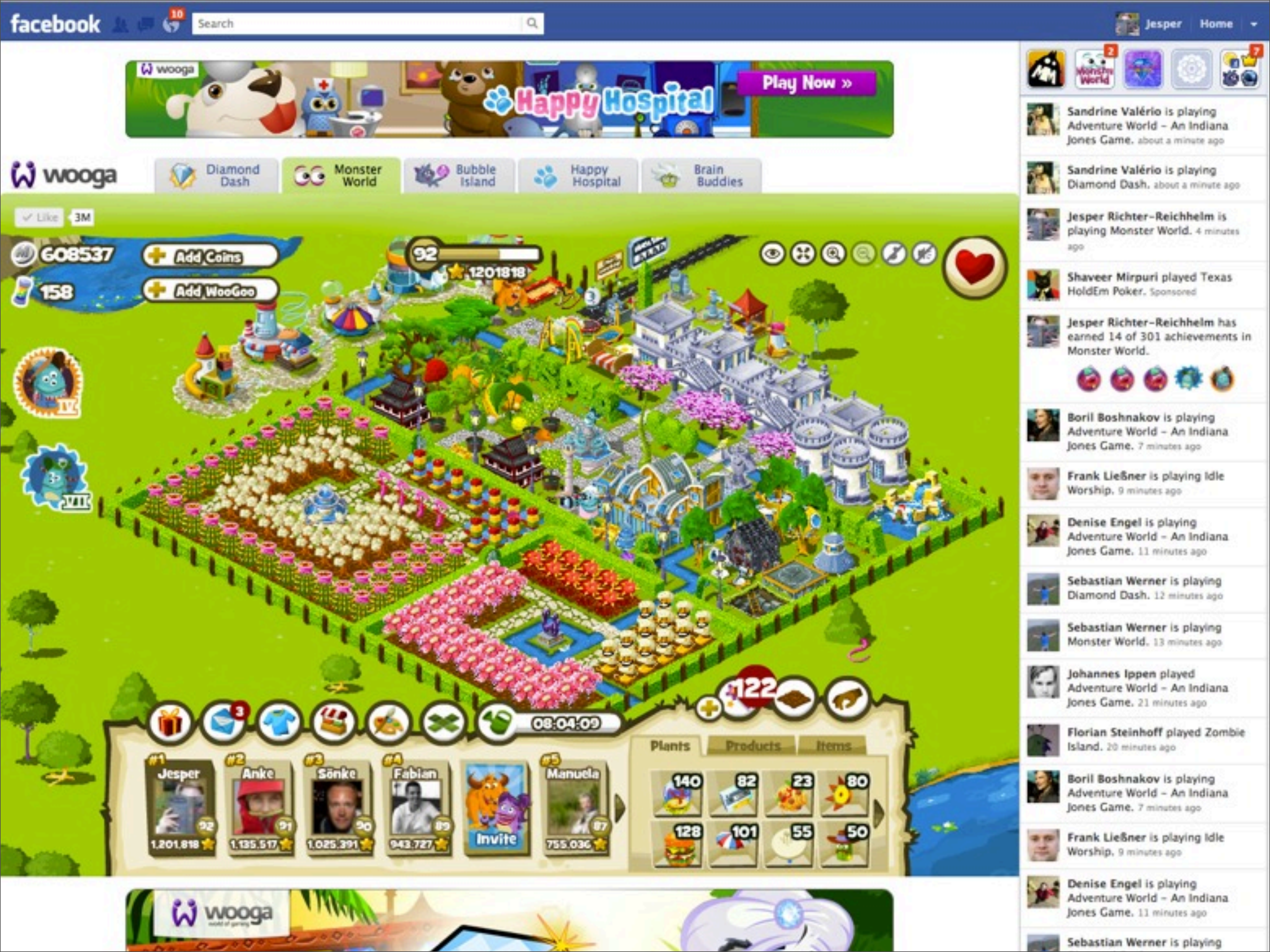
At Scale With Style

How we roll

Server architectures

How to innovate

Mind the limits



Sandrine Valério is playing Adventure World – An Indiana Jones Game. about a minute ago

Sandrine Valério is playing Diamond Dash. about a minute ago

Jesper Richter-Reichhelm is playing Monster World. 4 minutes ago

Shaveer Mirpuri played Texas HoldEm Poker. Sponsored

Jesper Richter-Reichhelm has earned 14 of 301 achievements in Monster World.

Boril Boshnakov is playing Adventure World – An Indiana Jones Game. 7 minutes ago

Frank Ließner is playing Idle Worship. 9 minutes ago

Denise Engel is playing Adventure World – An Indiana Jones Game. 11 minutes ago

Sebastian Werner is playing Diamond Dash. 12 minutes ago

Sebastian Werner is playing Monster World. 13 minutes ago

Johannes Ippen played Adventure World – An Indiana Jones Game. 21 minutes ago

Florian Steinhoff played Zombie Island. 20 minutes ago

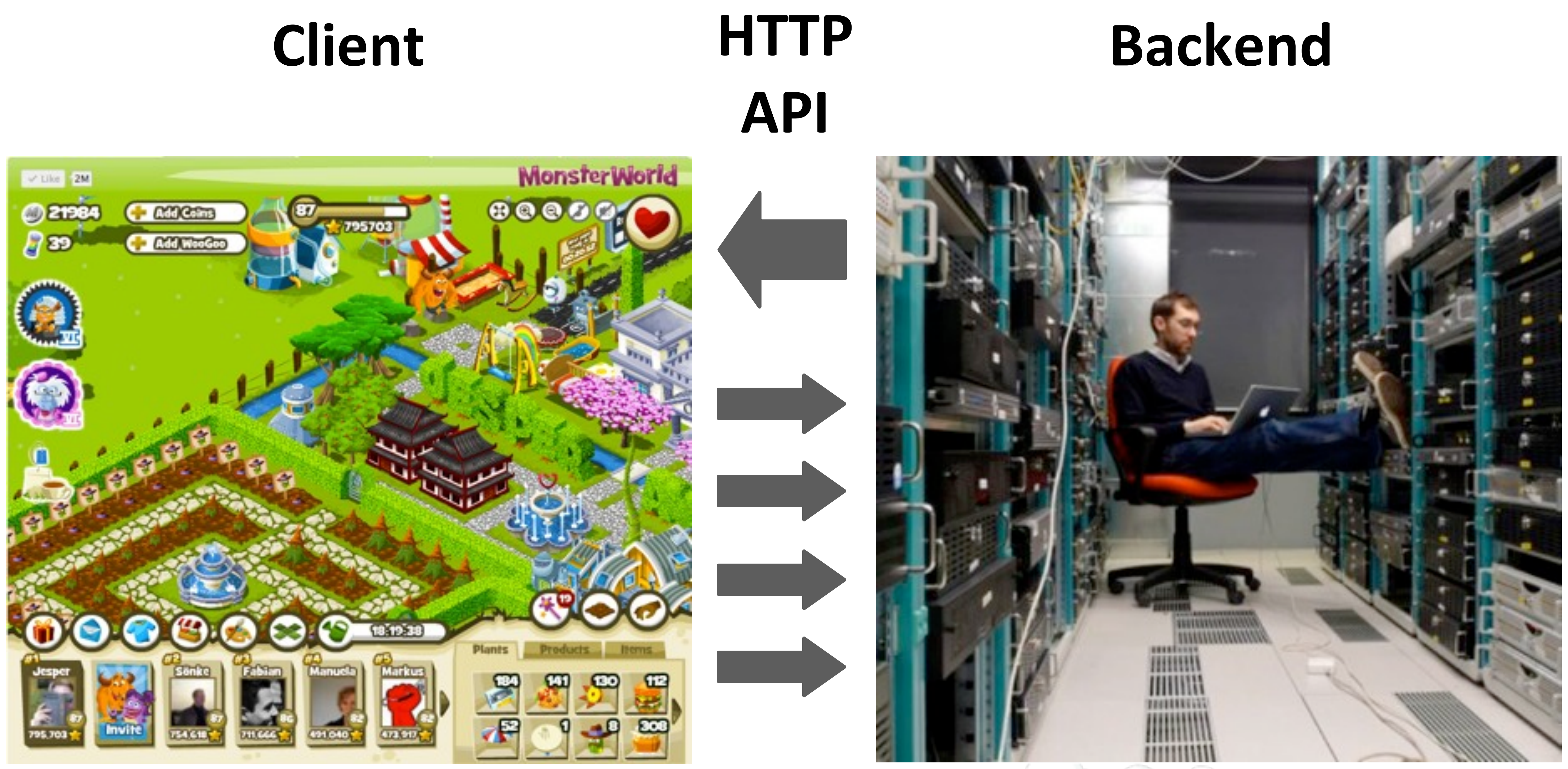
Boril Boshnakov is playing Adventure World – An Indiana Jones Game. 7 minutes ago

Frank Ließner is playing Idle Worship. 9 minutes ago

Denise Engel is playing Adventure World – An Indiana Jones Game. 11 minutes ago

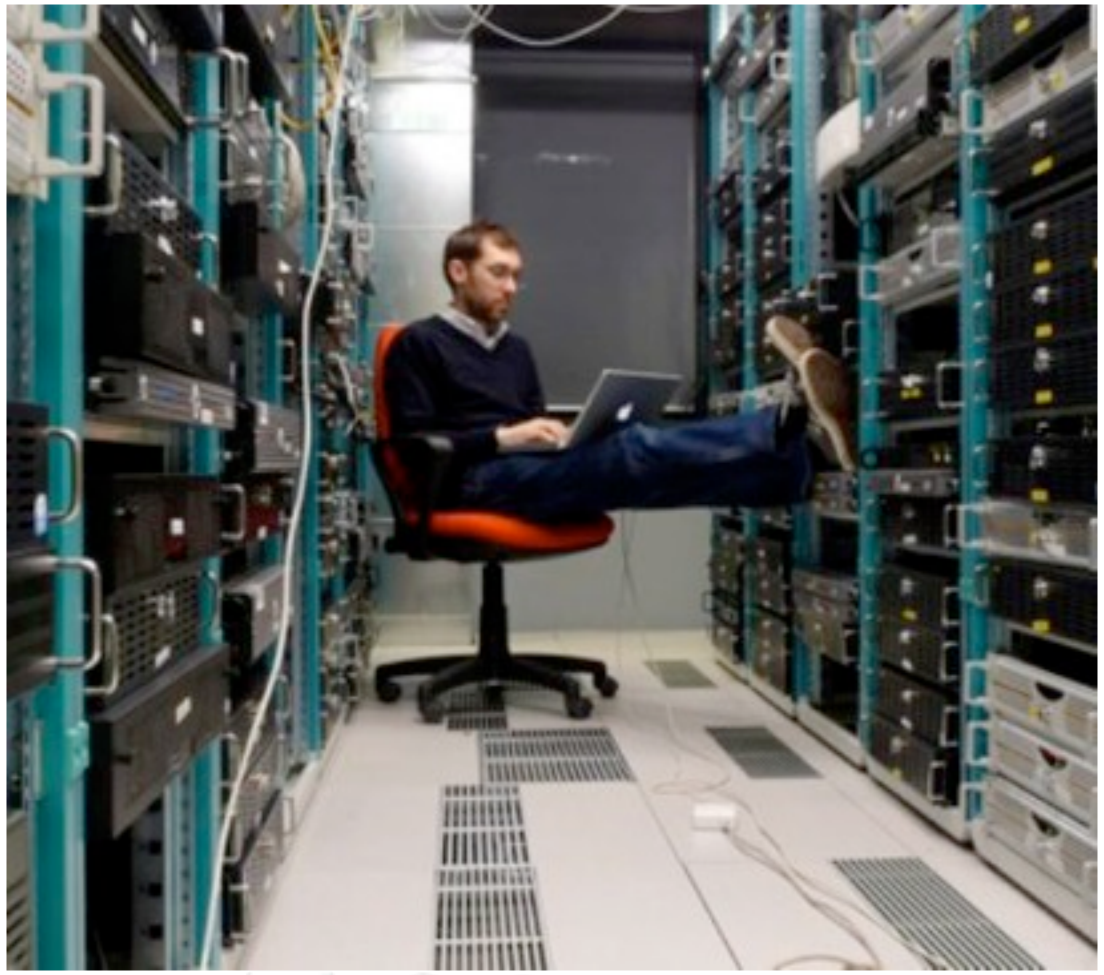
Sebastian Werner is playing

Typical game architecture



Typical game architecture

Backend



State Changes

Validation

Persistence

The scale is interesting

14 billion requests / month

>100,000 DB operations / second

>50,000 DB updates / second

Wooga’s approach to development

Small independent teams for each game

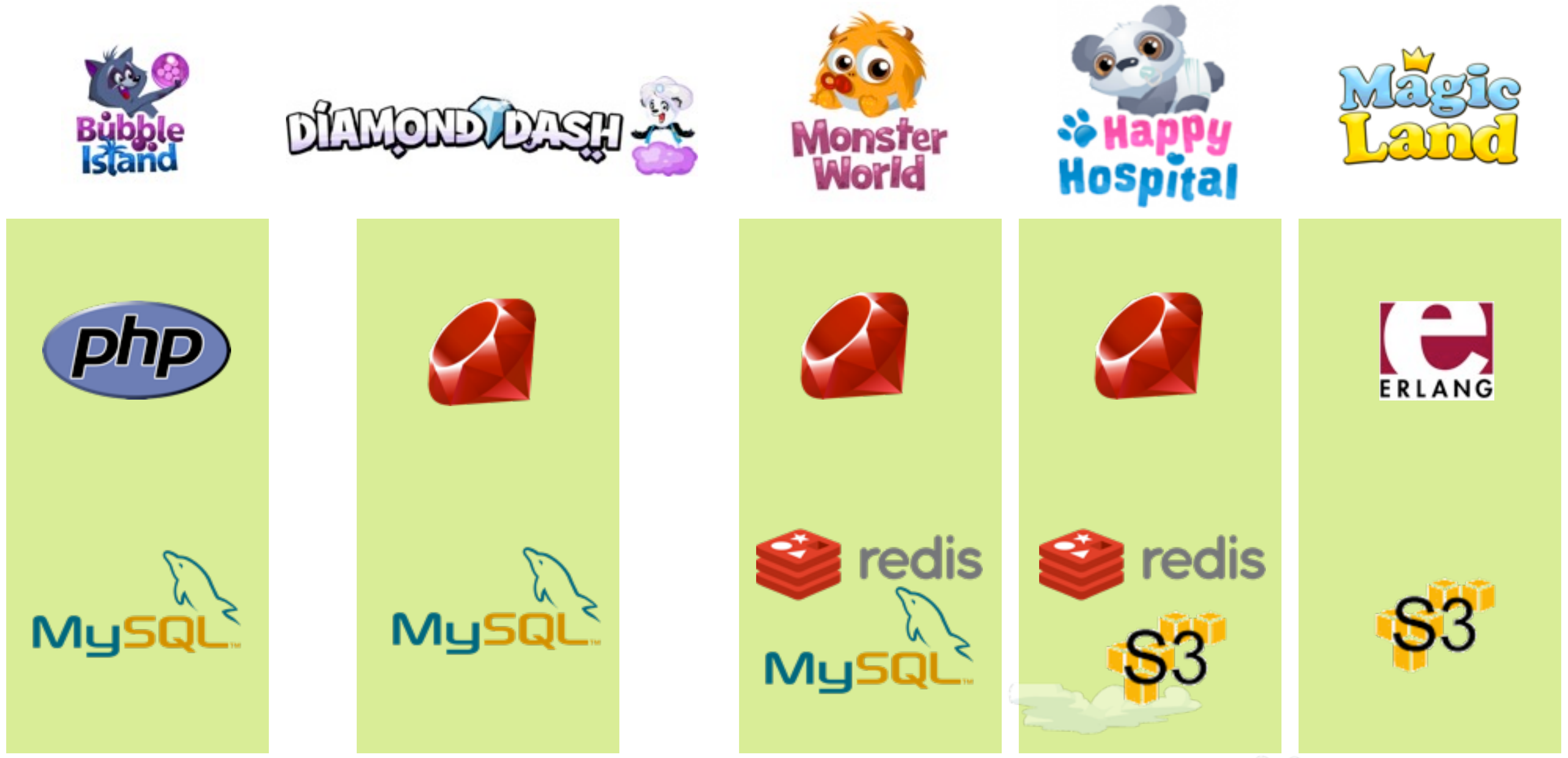
Team gets to choose tools and technologies

Same team also does ops after going live

Culture of sharing

Look around what is there, pick/adapt existing solutions, but take ownership for what you include

Existing backends – Technology landscape



At Scale With Style

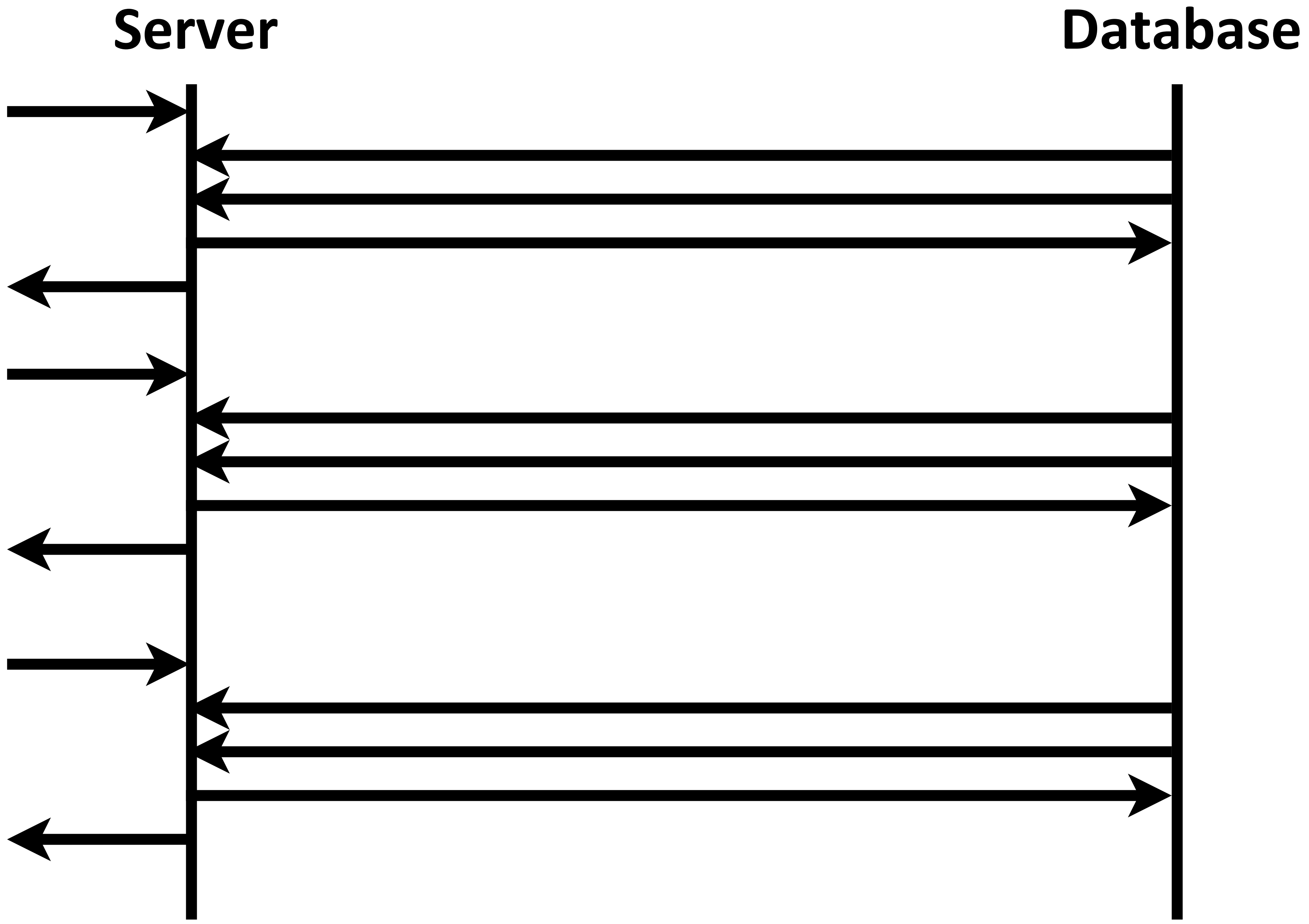
How we roll

Server architectures

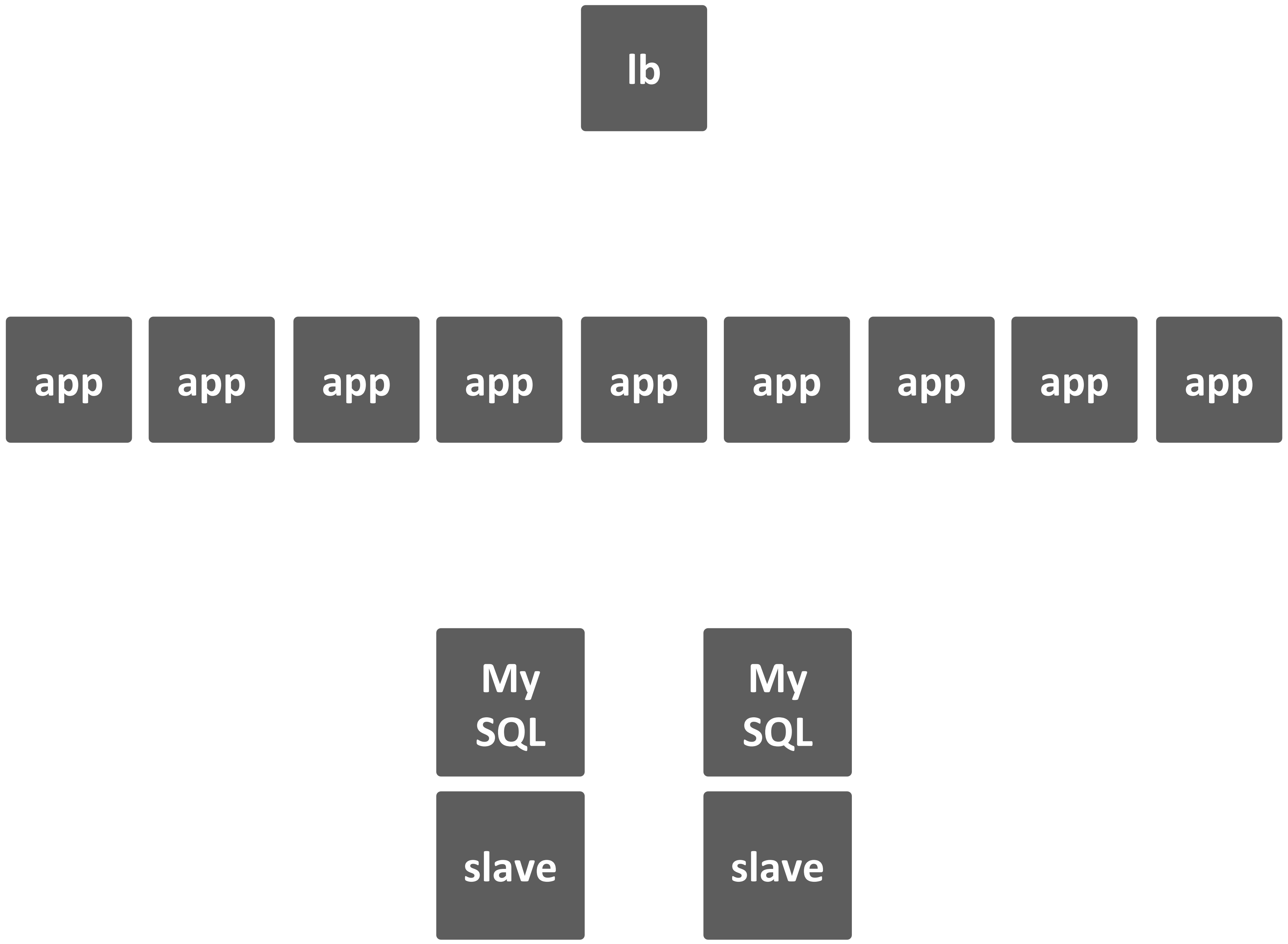
How to innovate

Mind the limits

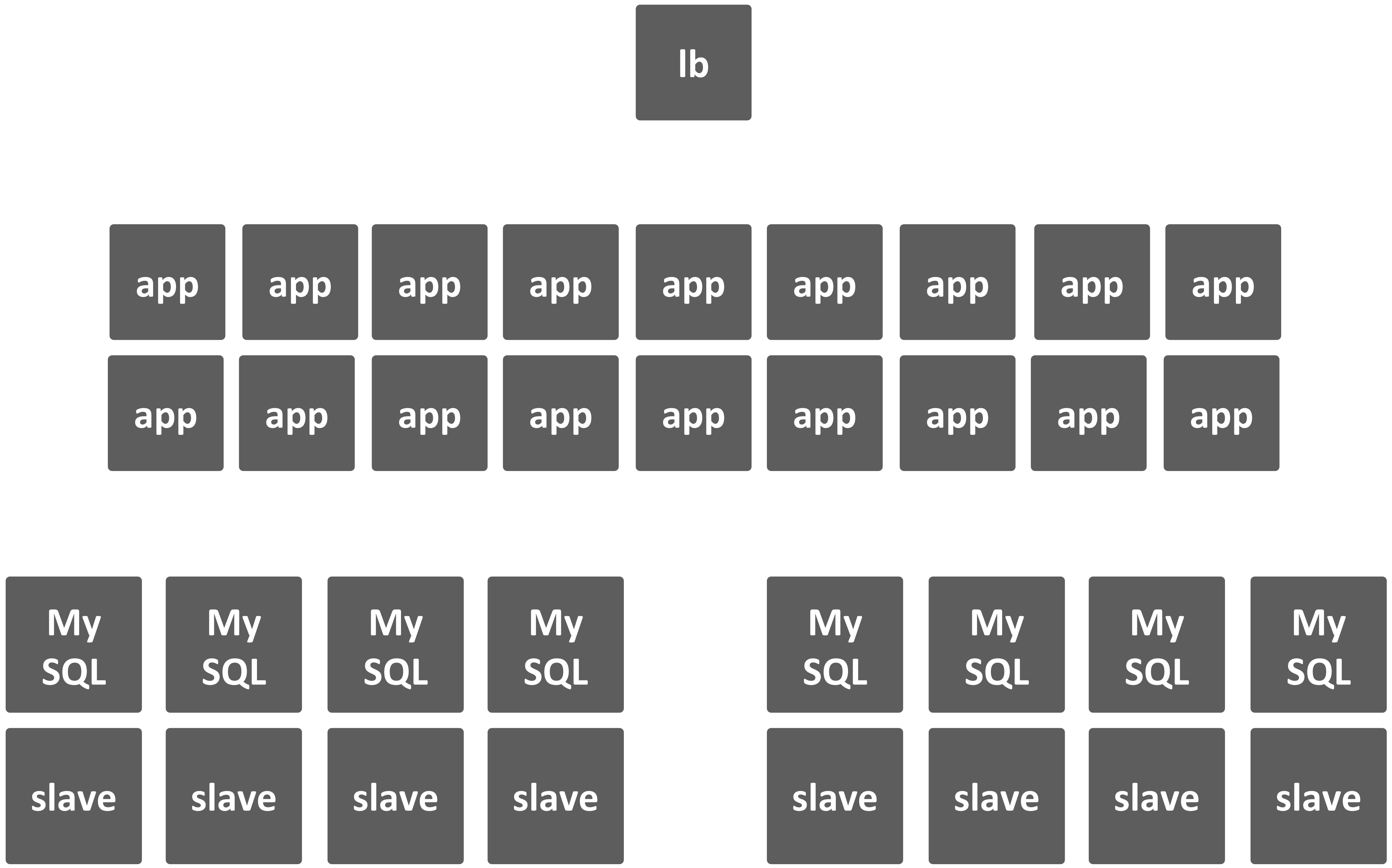
Most games use stateless application servers



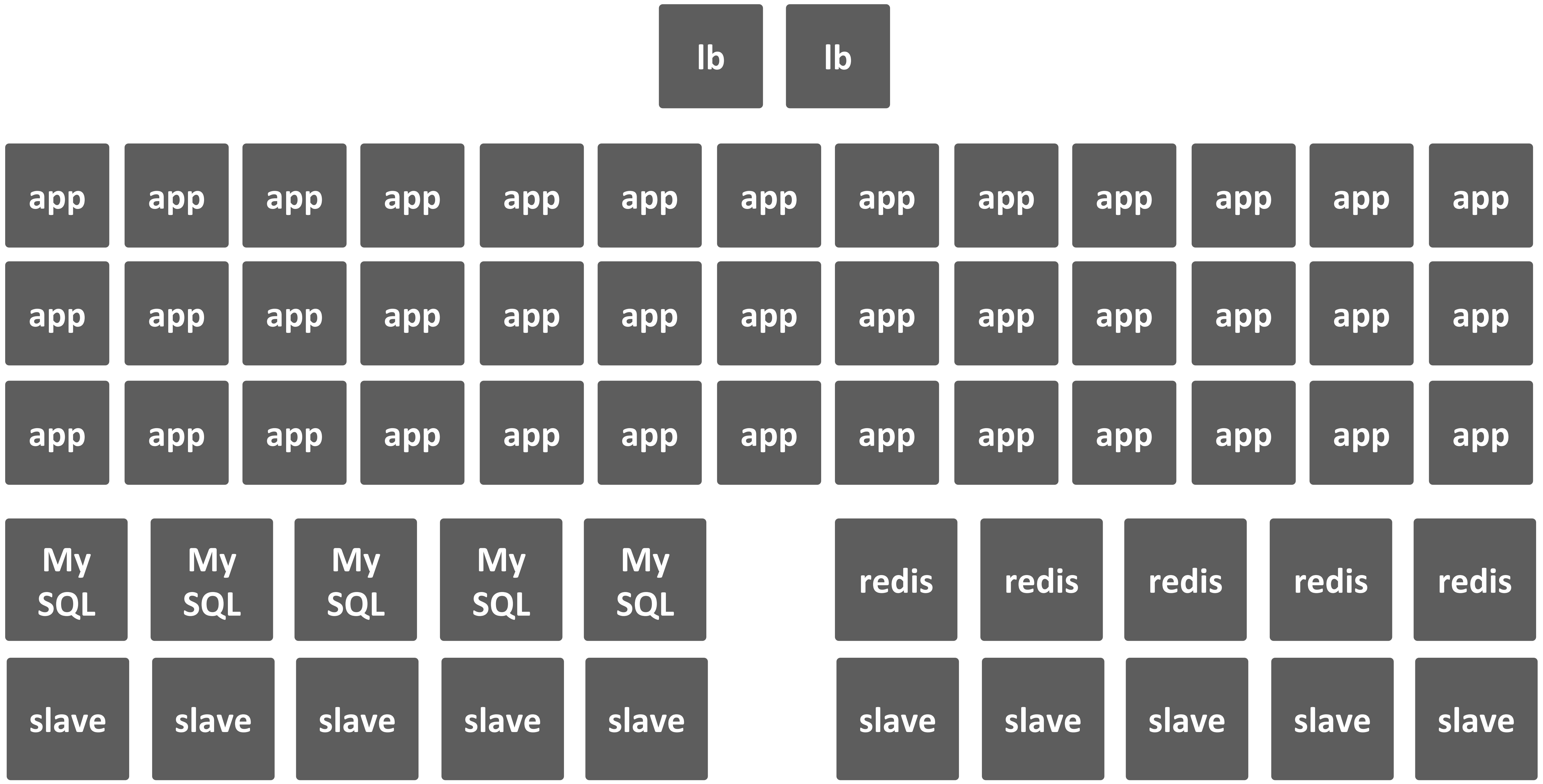
And then there’s sharding



More app servers, more sharding



Wait, seriously?!



Find the flaw

“Stateless application servers guarantee one thing:

The data is never where you need it!”

Paolo Negri, Developer @ Wooga

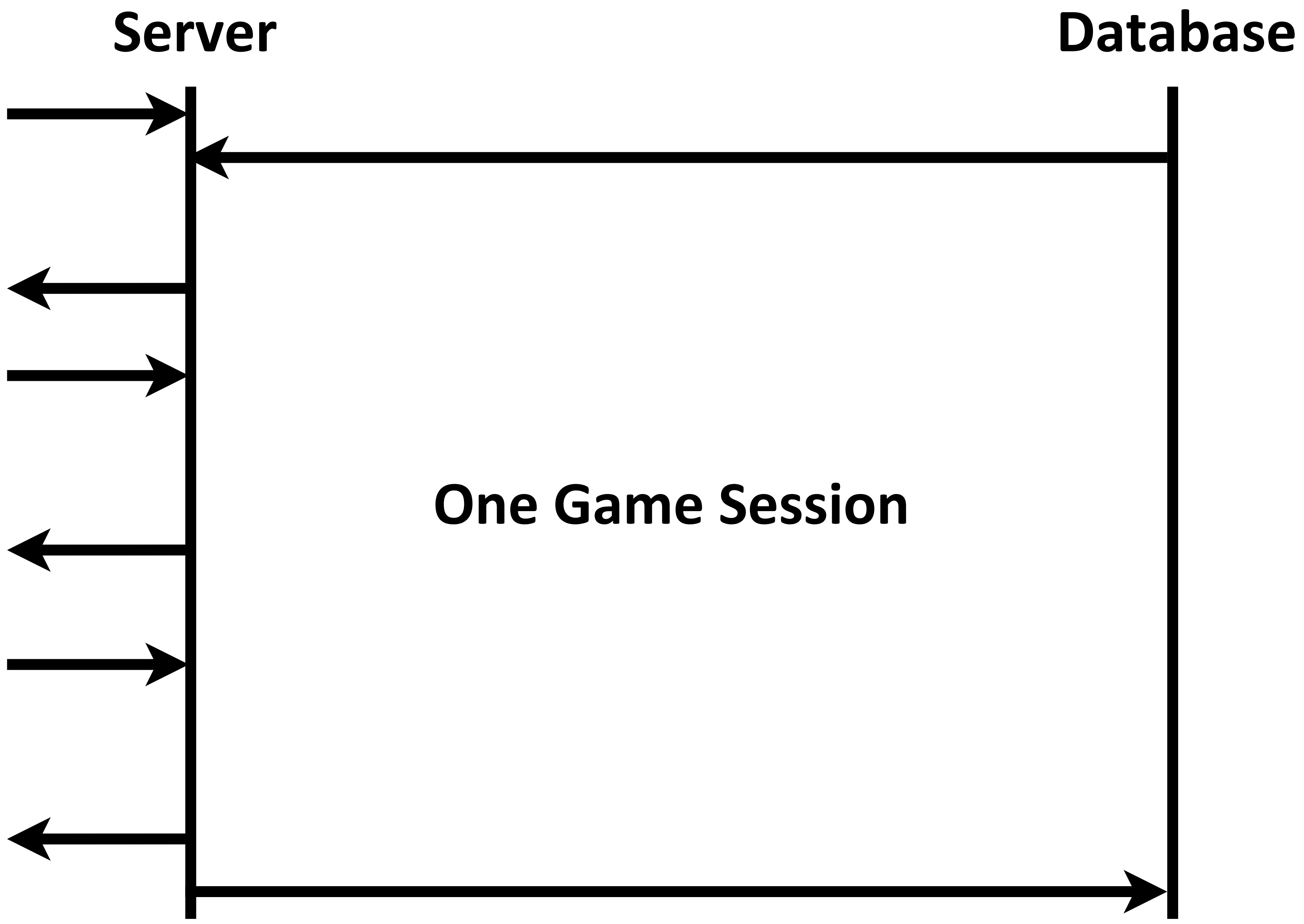
Strong session pattern

User starts playing

many transformations
of the same set of data

User stops playing

Stateful servers and DBs



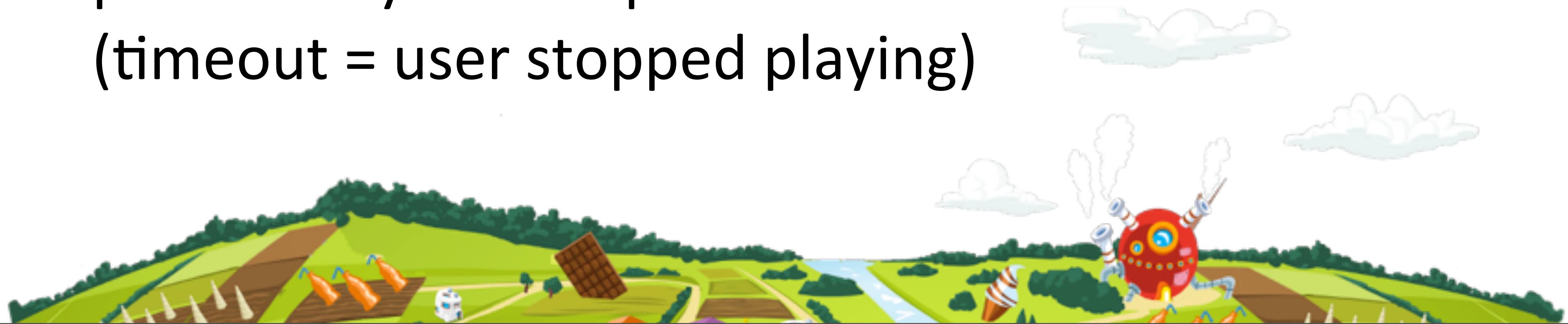
Stateful game server

One process per active user gaming session

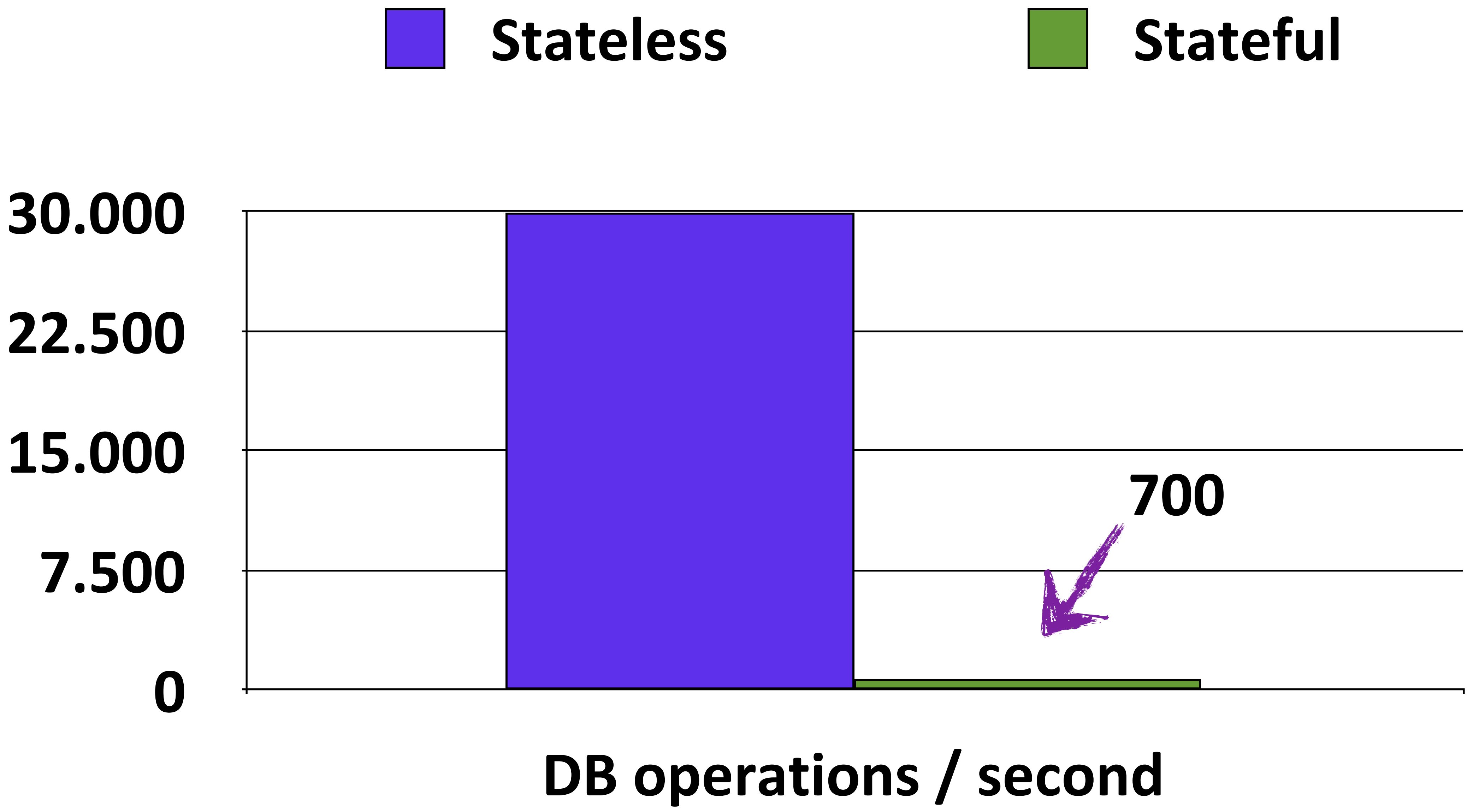
... holds the current state and is the only one that can modify it (**strong encapsulation**)

... handles all API calls for the given user one after the other (**concurrency control** through actor model)

... loads the game state from **storage** and writes it back periodically and on process termination (timeout = user stopped playing)



The DB is no longer the bottleneck



Magic Land uses Erlang

Details:

Awesome presentation on the Magic Land game server by @knutin & @hungryblank

Getting real with erlang

From the idea to a live system

Knut Nesheim @knutin
Paolo Negri @hungryblank

wooga
world of gaming

Thursday, November 3, 2011

http://www.slideshare.net/wooga/from-0-to-1000000-daily-users-with-erlang

At Scale With Style

How we roll

Server architectures

How to innovate

Mind the limits

CAN THERE BE
HAPPINESS
WITHOUT RUBY
memegenerator.net

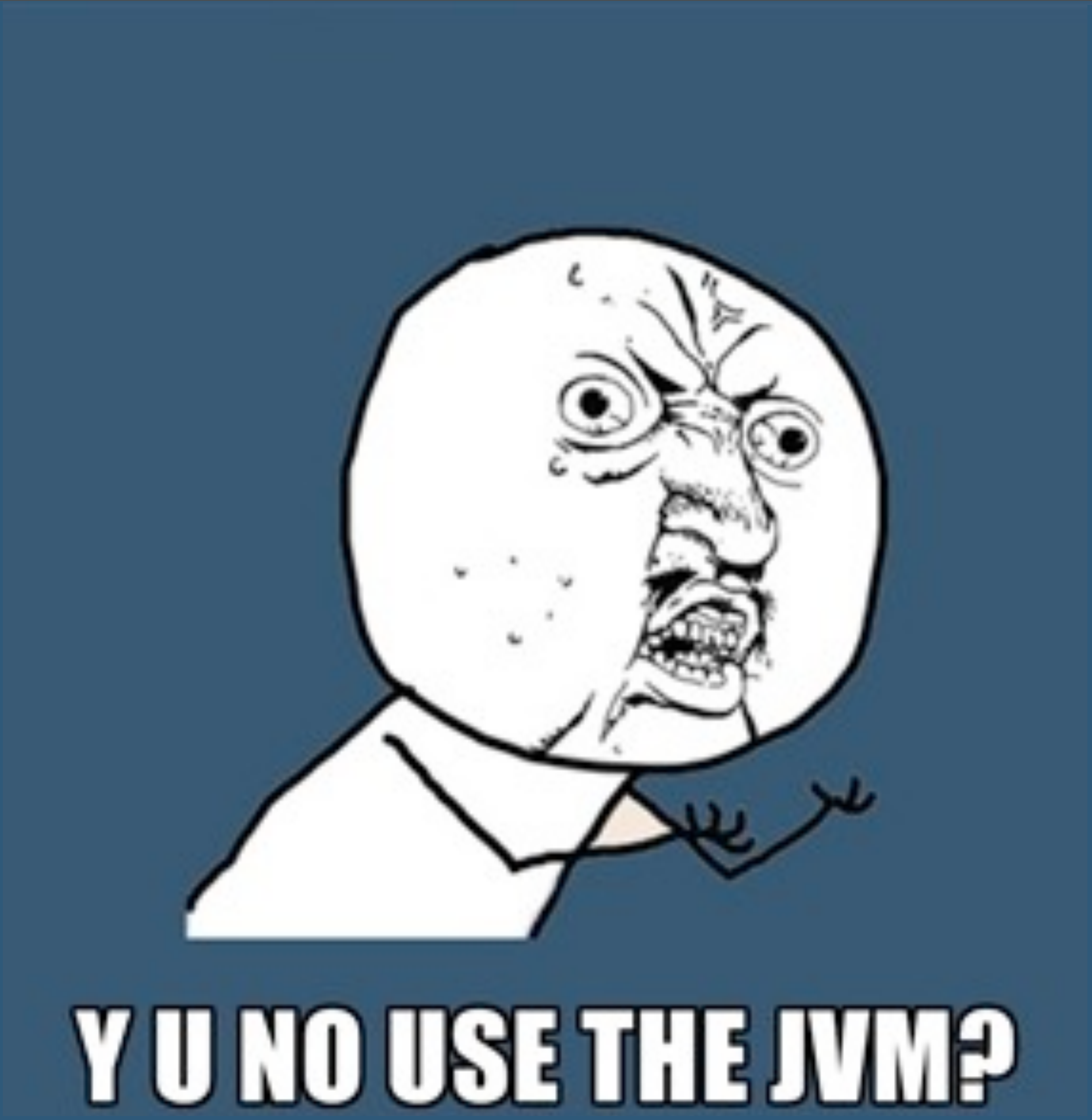
Erlang & Ruby

Erlang is great

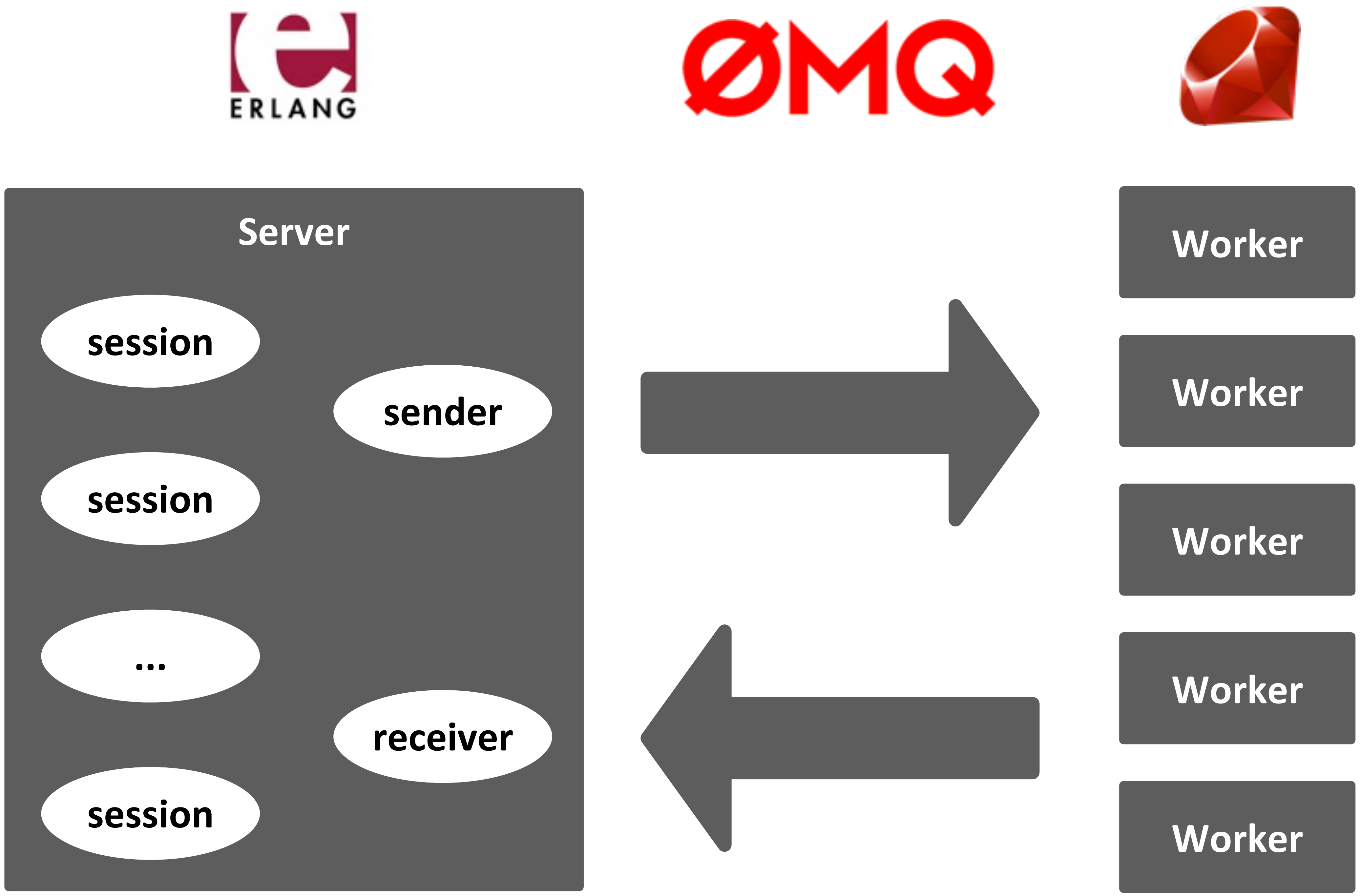
- Concurrent, robust
- Great for operation

Ruby is great

- Concise, expressive, flexible
- Great for development


Y U NO USE THE JVM?

Bringing two worlds together

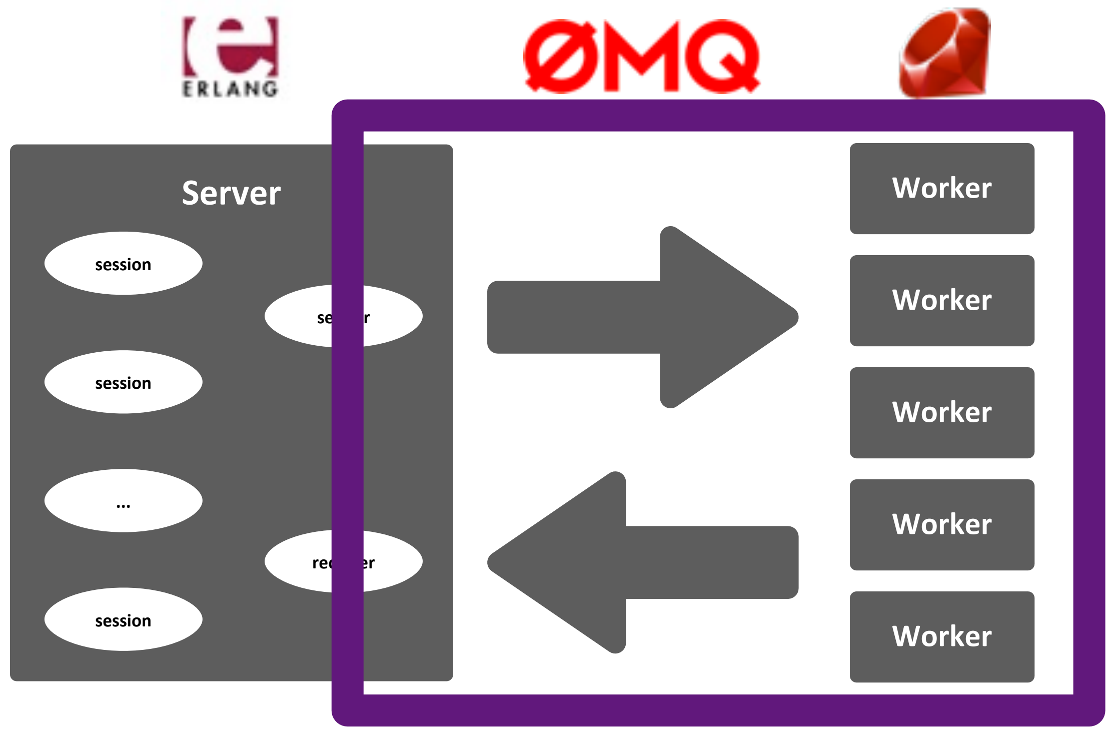


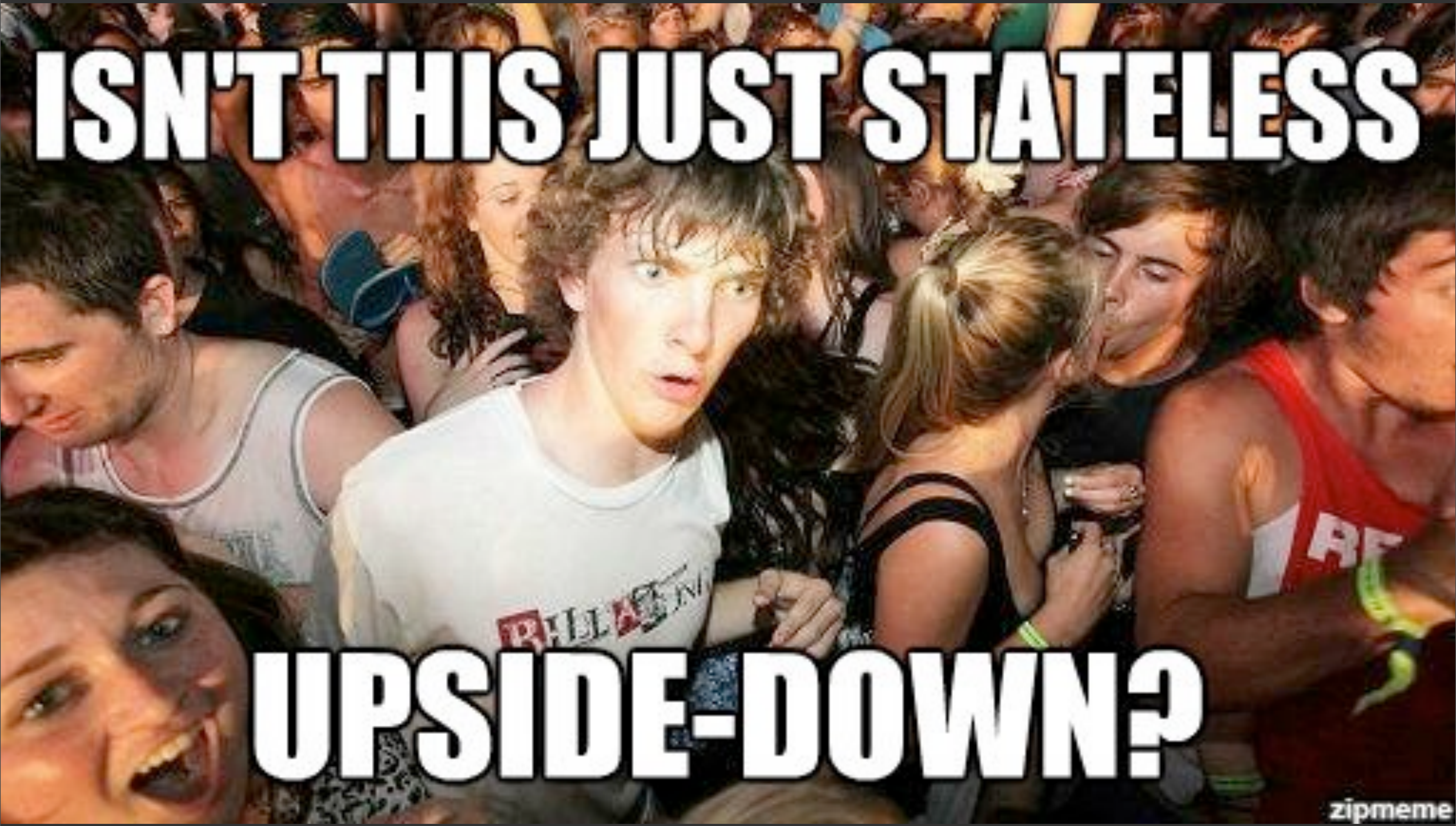
Do you know Mongrel2?

Mongrel2
The app is in control.

Web Server that hooks up applications via ØMQ

➡ **We chose the same queue setup & message format**




ISN'T THIS JUST STATELESS
UPSIDE-DOWN?
zipmeme

Connecting the dots

Mongrel2 http://mongrel2.org/ protocol & ZeroMQ setup

Erlang: https://github.com/hungryblank/emongrel2

Ruby

rack-mongrel2 fork https://github.com/khiltd/khi-rack-mongrel2
rack protocol http://rack.rubyforge.org
Sinatra http://www.sinatrarb.com/

➡ **essentially we are speaking HTTP over ZeroMQ and can hook up any Rack-based Ruby web framework**

Example controller in Ruby

```
  app.game_action '/:actor/fruit_tree/self/shake',
                  :observable => true,
                  :affects => [:fruit_trees, :user],
                  :params => [:x, :y] do

    x, y = params[:x], params[:y]
    fruit_trees[x, y].shake

  end
```

DSL-like definition of game action

Skinny as controllers should be

Example model in Ruby

```ruby
class FruitTree < Tree

  property :last_shake_time,           :type => Integer, :default => 0
  property :collectable_fruit_count, :type => Integer, :default => 0

  def shake
    raise G8::Error::Validation, "no fruit!" unless carries_fruit?

    session.user.xp += 1
    session.user.energy -= 1
    self.last_shake_time = game_time
    self.collectable_fruit_count = config.fruit_count
  end

end
```

Easily unit testable

Minimal amount of code

Game state

Game state is split in multiple parts

user, map, fruit_trees etc.

Erlang does not care about content

Serialized Ruby objects

Erlang does know mapping of state parts to URLs

Looking back at the game action

```
app.game_action '/:actor/fruit_tree/self/shake',
                :observable => true,
                :affects => [:fruit_trees, :user],
                :params => [:x, :y] do

  x, y = params[:x], params[:y]
  fruit_trees[x, y].shake
end
```

Mapping of state parts to game actions

- Worker knows mapping
- Worker pushes mapping to Erlang on startup
- Erlang can query mapping if needed

NICE!

http://www.flickr.com/photos/aigle_dore/

At Scale With Style

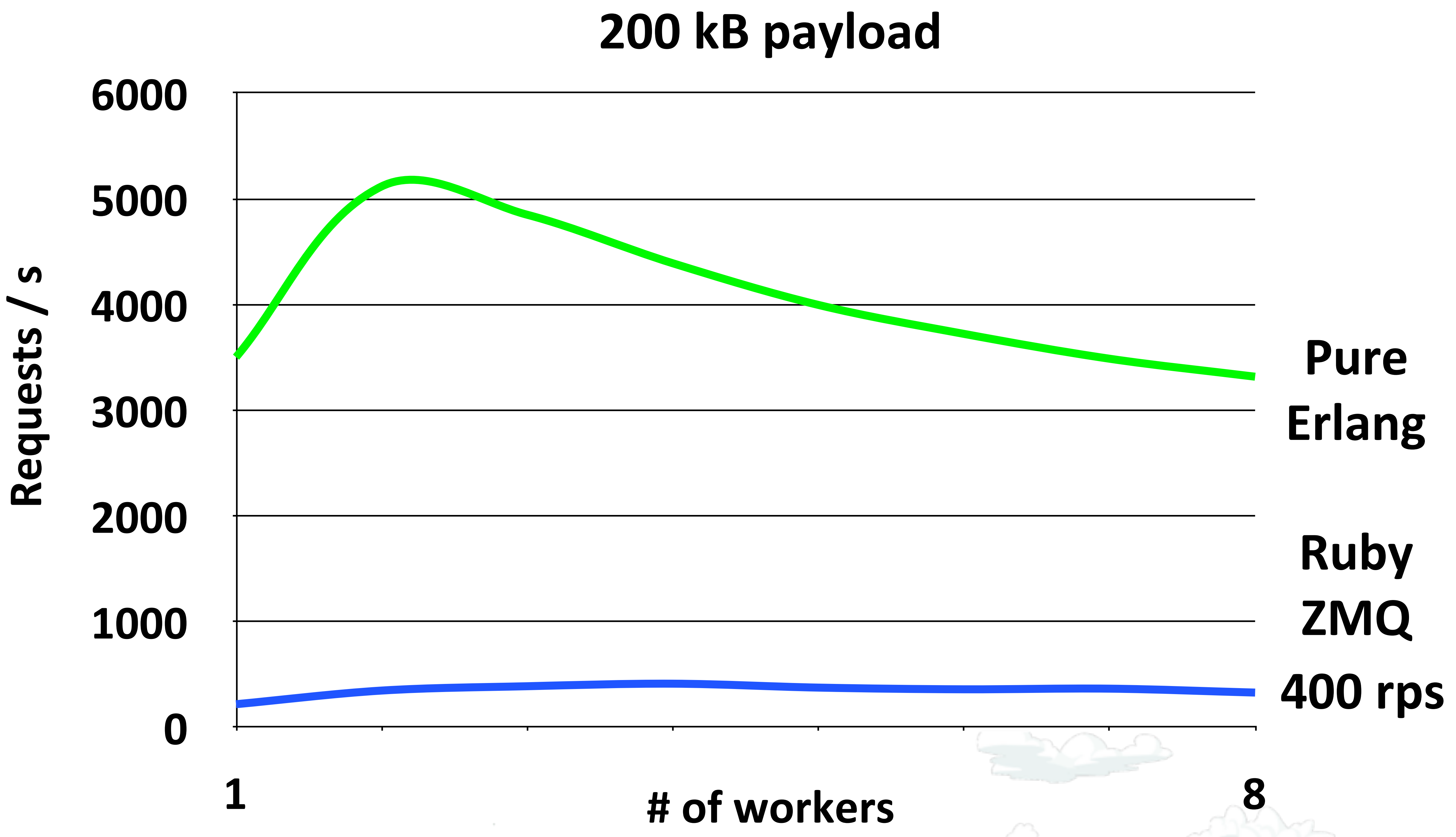
How we roll

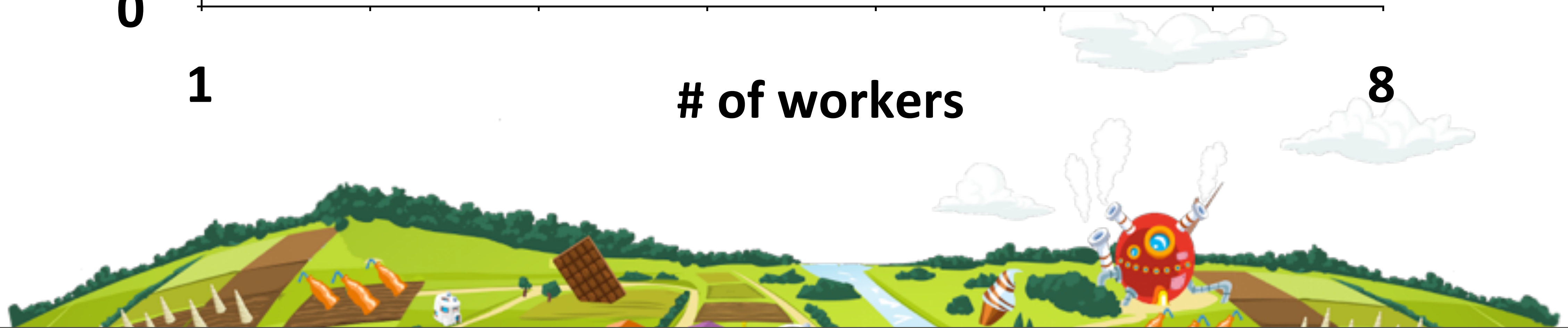
Server architectures

How to innovate

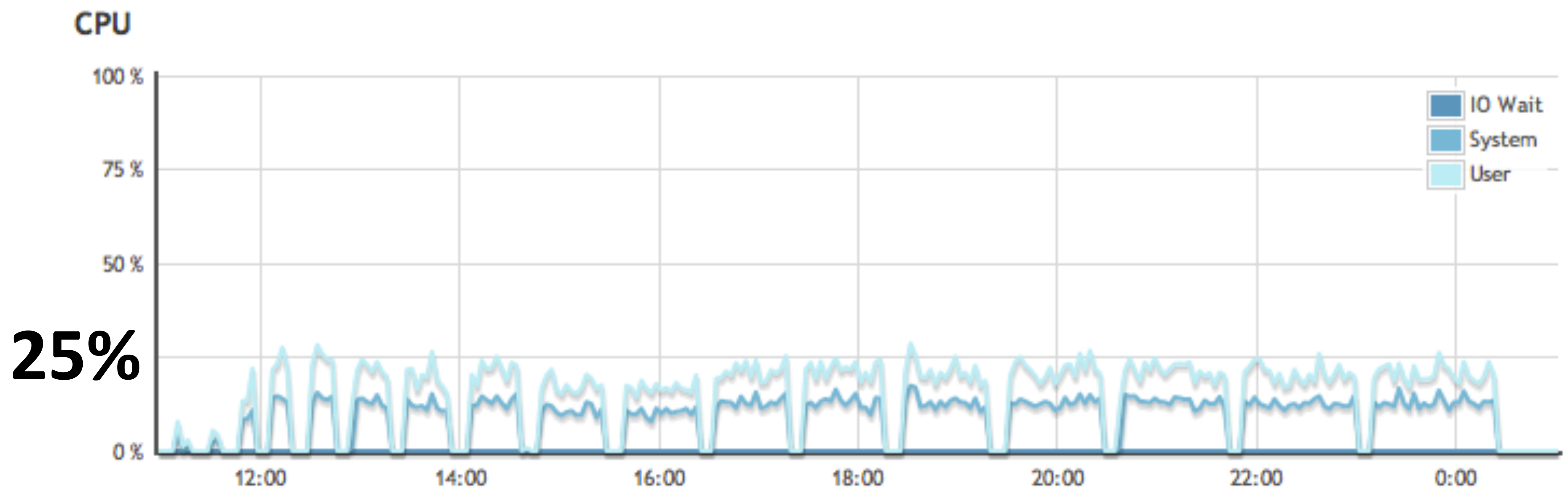
Mind the limits

Performance impact for large payloads



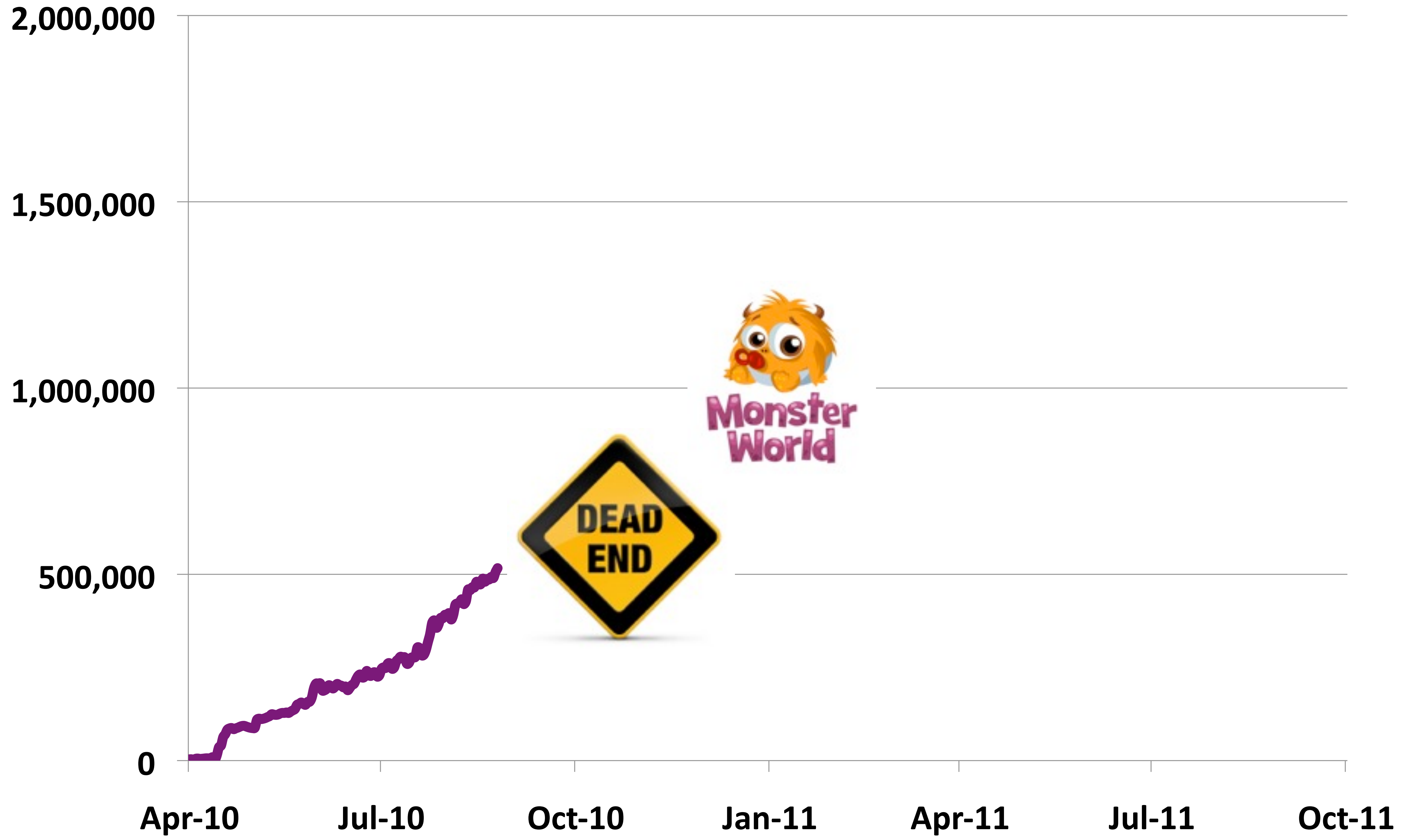


NOT CPU bound



Fixed bandwith limit @ 300 MB/s

This has happened before



HAPPINESS
LIES IN A WORKING PRODUCT
memegenerator.net

Example controller in Erlang

```
  shake(Args, State) ->
      Coords      = g8_map:coords(Args),
      Map         = g8_game:get_map(State),
      Tree        = g8_map:find(Coords, Map),

      NewTree     = g8_fruit_tree:shake(Tree),
      UserEffects = [incr_xp, decr_energy],
      NewMap      = g8_map:place(Coords, NewTree, Map),

      [observable,
       {state, g8_game:set_map(NewMap,
                g8_user:apply(UserEffects, State))}].
```

Example model in Erlang

```
shake(Tree) ->
    validate_carries_fruit(Tree),

    FruitCount = g8_fruit_tree_config:fruit_count(Tree),
    Tree#fruit_tree{last_shake_time = g8_game:gametime(),
                    collectable_fruit_count = FruitCount}.
```

CAN THERE BE HAPPINESS
WITHOUT RUBY
✓ YES
memegenerator.net

Questions?

Martin Rehfeld
@klickmich

slideshare.net/wooga
wooga.com/jobs

wooga
world of gaming