

How the cloud got its groove back

Orion Henry and Blake Mizerany
Erlang Factory, London '09



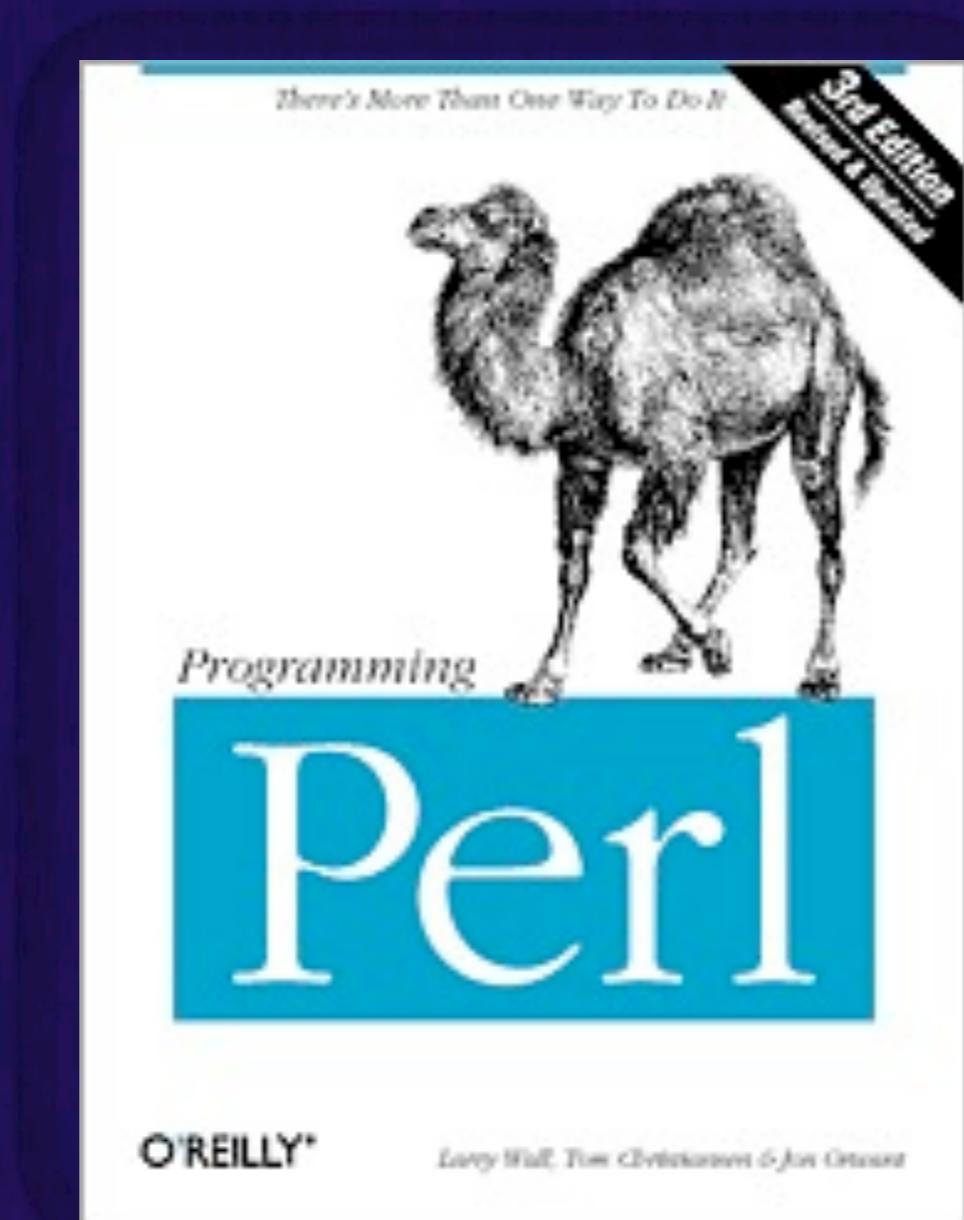
heroku

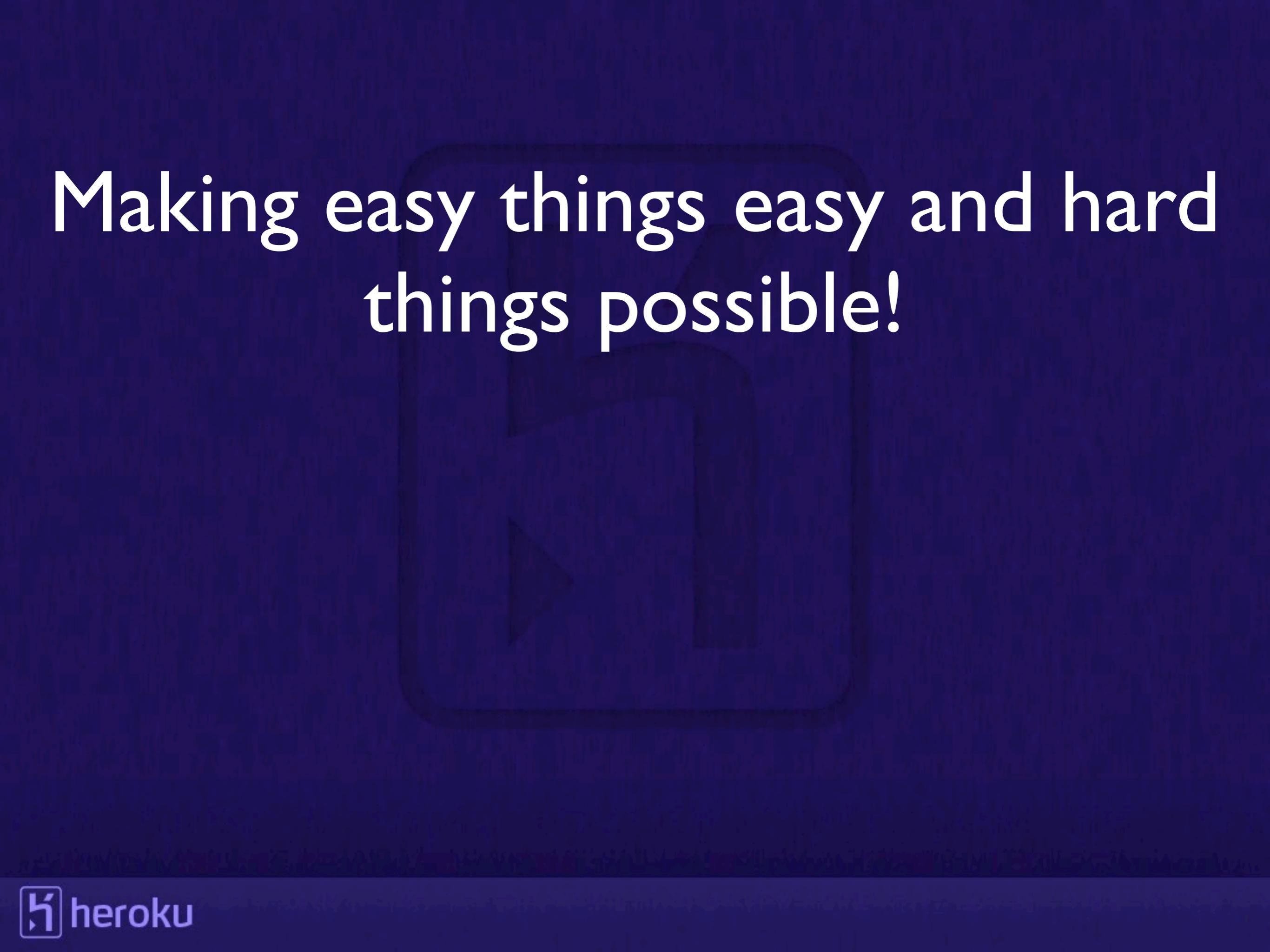
frictionless ruby platform

Why are you at our Erlang Conference?!



A quick overview of languages

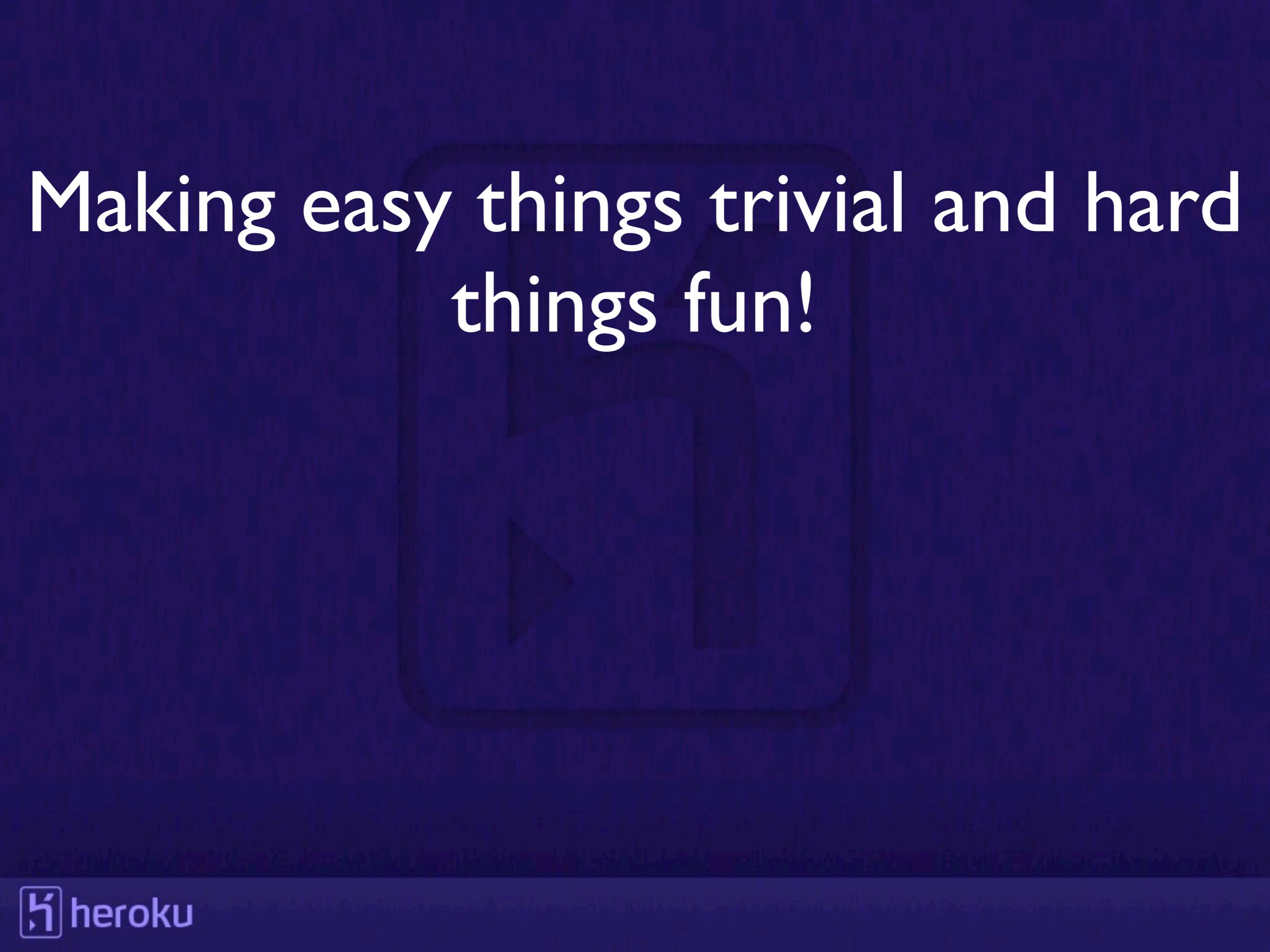




Making easy things easy and hard things possible!

Ruby





Making easy things trivial and hard
things fun!

A Concurrent Approach to Software Development

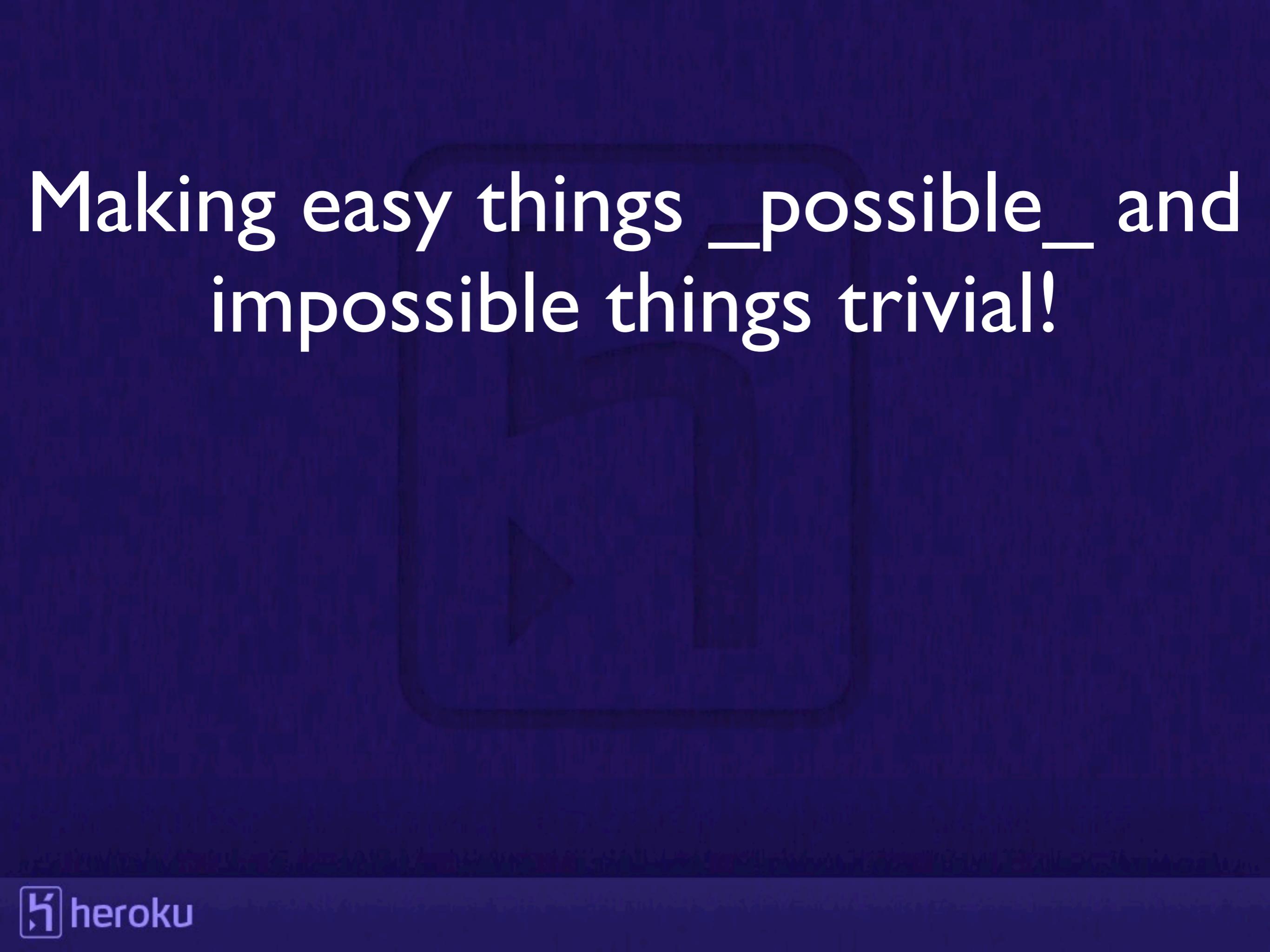
Erlang

Programming



O'REILLY®

*Francesco Gesarini
& Simon Thompson*



Making easy things possible and
impossible things trivial!

Traditional infrastructure



Traditional infrastructure

- fixed physical resources
- brittle configurations
- difficult to change
- difficult to scale

Cloud infrastructure



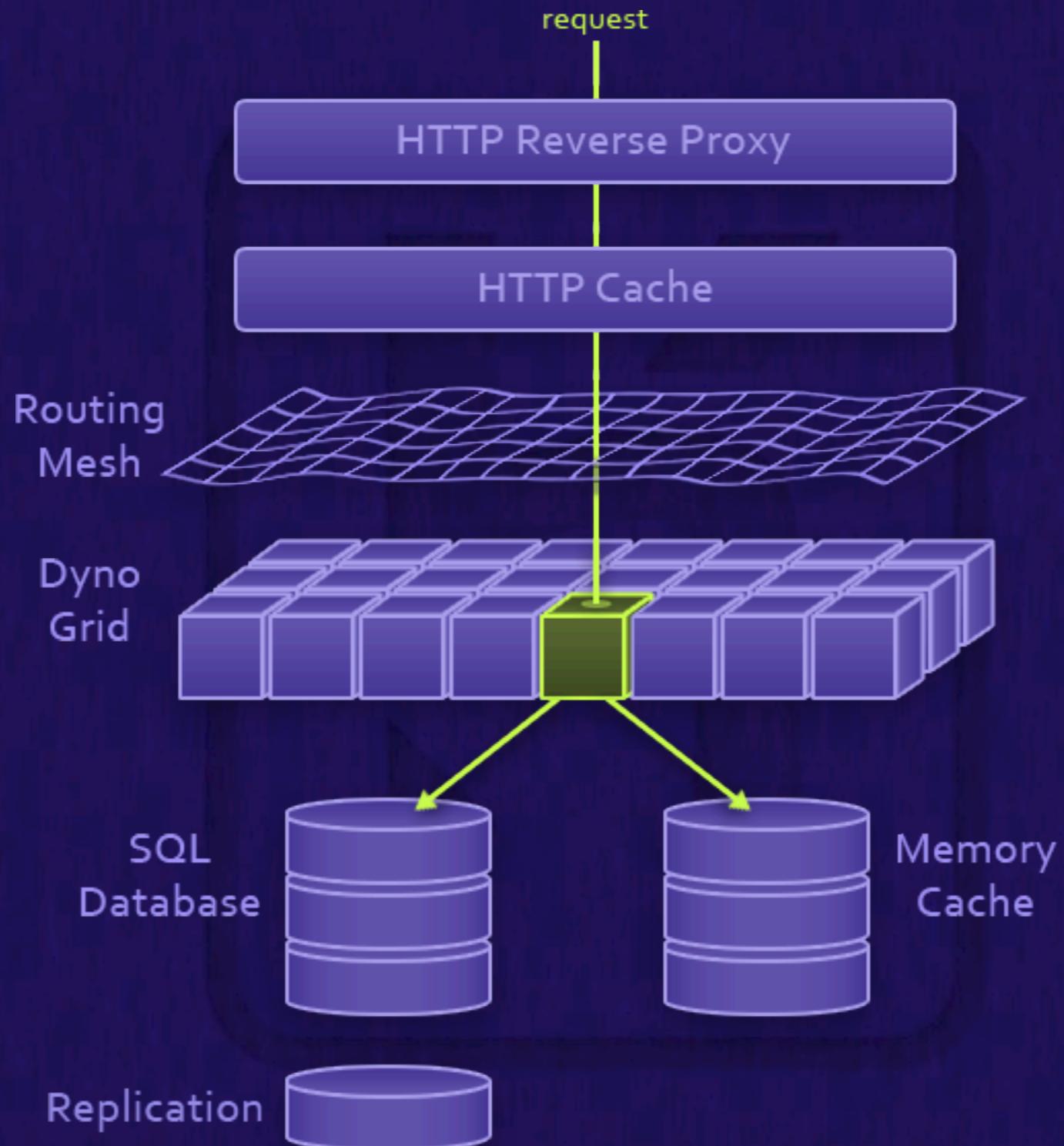
Cloud infrastructure

- transient virtualized resources
- easily scalable
- multi-tenant
- dynamic configurations
- on-demand resources
- pay only for what you use



heroku

DEMO!

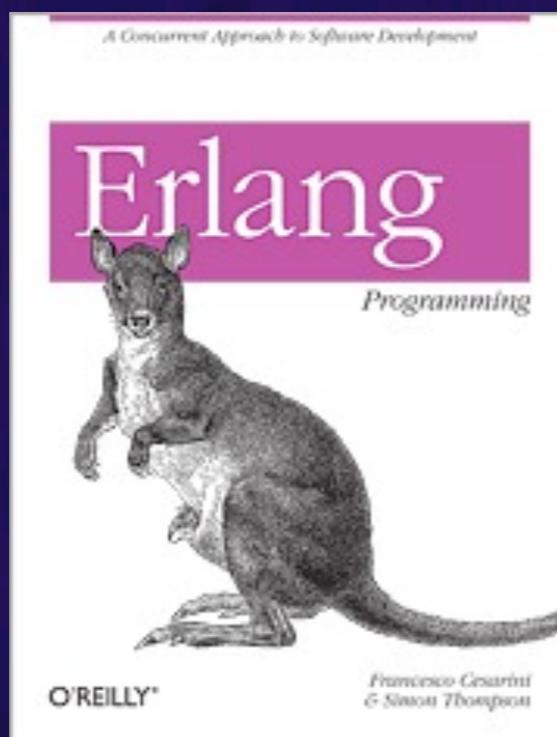


this is the future!





this is the future!



the enabler

The transient infrastructure

The Heroku routing mesh enables:

- tens of thousands of applications
- multi-tenant environment
- thousands of route changes per minute
- offline signifiers
- error signifiers

The natural tool for the job



Intelligent Proxies



The anatomy a proxy



The anatomy of our routing mesh





why not use C?

```
domain_to_dyno(Domain) of
  case get_dynos(Domain) of
    [] -> need_dyno;
    Dynos -> { route, pick_random(Dynos), Domain }
end.
```

```
domain_to_dyno(Domain) of
  case get_dynos(Domain) of
    [] -> need_dyno;
    Dynos -> { route, pick_dyno(Dynos), Domain }
  end.
```

```
pick_dyno(Dynos) ->
  MinLoad = get_min_load(Dynos),
  Unloaded = filter_with_load(Dynos, MinLoad),
  pick_random(Unloaded).
```

```
domain_to_dyno(Domain) of
  case get_dynos(Domain) of
    [] -> need_dyno;
    Dynos ->
      case pick_dyno(Dynos) of
        busy -> { busy, Domain };
        Dyno -> { route, Dyno, Domain }
      end
  end.
```

```
pick_dyno(Dynos) ->
  case filter_with_load(Dynos, 0) of
    [] -> busy;
    Unloaded -> pick_random(Unloaded)
  end.
```

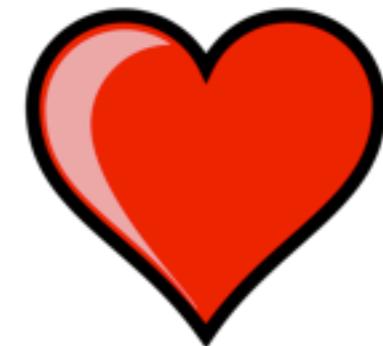
A dash of secret sauce

RabbitMQ

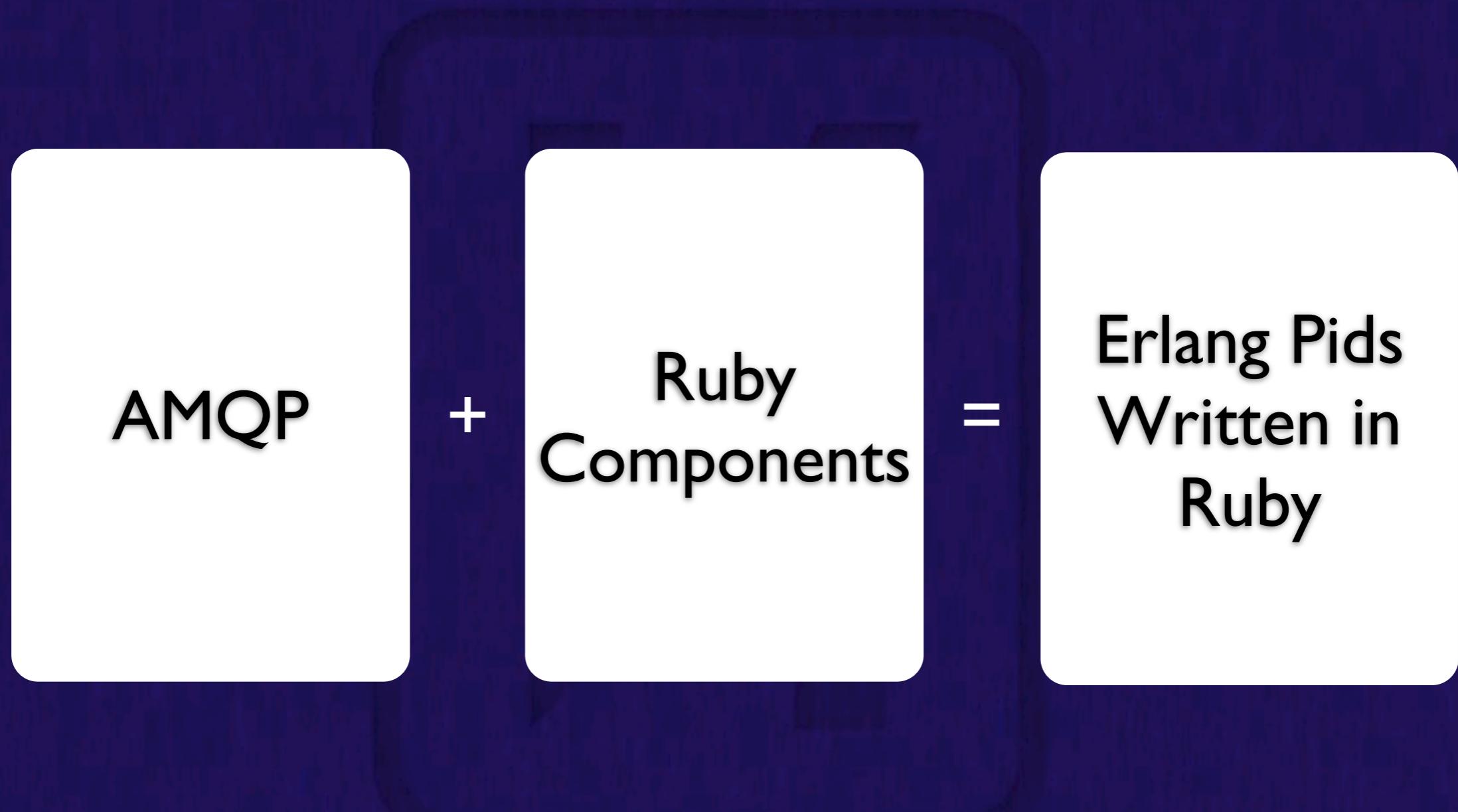
+

ETS/DETS

+



Multi-Lingual Actor Model



Nginx + Erlang Proxy

- frictionless routing
- fair load balancing
- connection queueing
- header introspection/editing
- custom traffic introspection
- custom authentication
- dynamic app scaling/idling

Memcached + Erlang Proxy

- authentication
- multi-tenant
- dynamic cache discovery
- automatic cache hashing
- transparent cache resizing
- protocol upgrade
- virtual infinite cache

PostgreSQL + Erlang Proxy

- live server migration
- connection pooling
- dynamic database discovery
- statement level replication
- load shaping
- business-specific constraints