Productive | > Reliable | > Fast

#### Phoenix framework



# Why another language/framework?

Ruby is Slow

# Why another language/framework?

- \* Ruby is Slow
- \* Rails Performance

# Why another language/framework?

- \* Ruby is Slow
- \* Rails Performance
- People are still complaining!

#### O.K. What are the alternatives?

#### O.K. What are the alternatives?

\* Go

#### Go?

#### O.K. What are the alternatives?

- \* Go
- \* Scala/Clojure

# Scala/Clojure?



#### O.K. What are the alternative?

- \* Go
- \* Scala/Clojure

## Why do we love Ruby?

- \* Community
- \* The joy of programming
- \* Productivity

# You might still say it's not performing as you want!

Seriously!?

#### What if?

#### What if?

\* You can have all the benefits of Ruby

#### What if?

- \* You can have all the benefits of Ruby
- \* Without sacrificing performance.

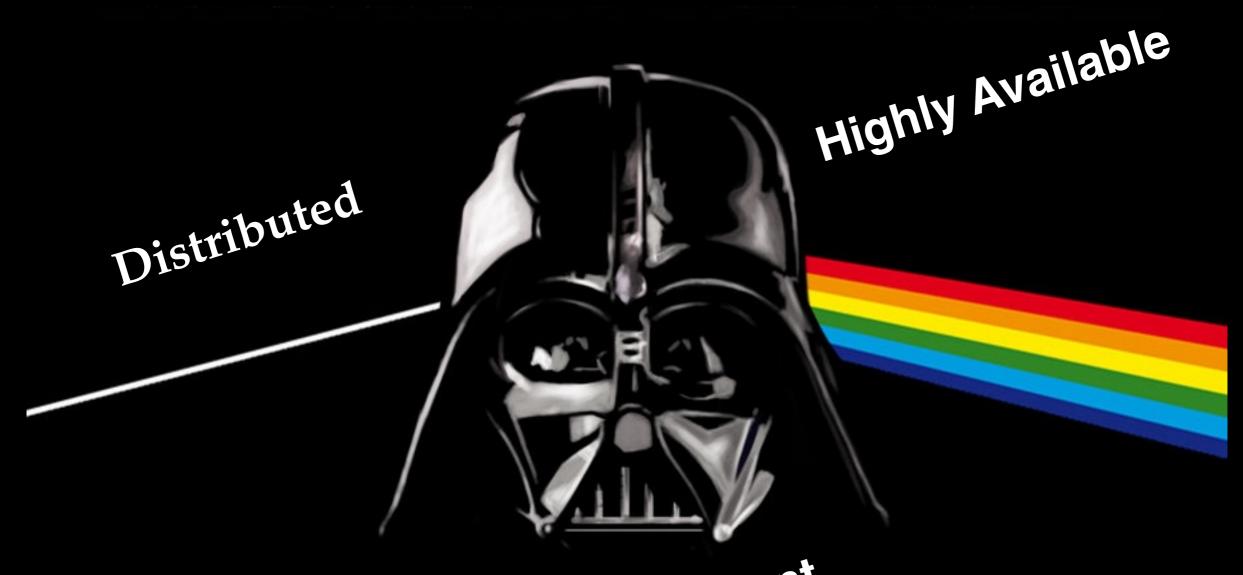


#### Meet Elixir

### Why Elixir/Phoenix?

\* Fast

# Erlang VM



Fault-tolerant

#### Benchmark

Framework	Throughput (req/s)	Latency (ms)
Phoenix 0.13.1	179685.94	0.61ms
Gin (GO)	176156.41	0.65ms
Play (Java/Scala)	171236.03	1.89ms
Rails	11903.48	8.50ms

https://gist.github.com/omnibs/e5e72b31e6bd25caf39a

### Why Elixir/Phoenix?

- \* Fast
- \* Reliable

#### Reliable

- \* Process Monitoring/Supervision
- \* Recover Failure

# Why Elixir/Phoenix?

- \* Fast
- \* Reliable
- \* Productive

#### Productive

```
defmodule Etel.Router do
     use Etel.Web, :router
     pipeline :browser do
       plug :accepts, ["html"]
       plug :fetch_session
       plug :fetch_flash
       plug :protect_from_forgery
       plug :put_secure_browser_headers
 9
10
     end
11
12
     pipeline :api do
13
       plug :accepts, ["json"]
14
     end
15
16
     scope "/", Etel do
17
       pipe_through :browser # Use the default browser stack
18
19
       get "/", PageController, :index
20
     end
21
22
     # Other scopes may use custom stacks.
     scope "/api", Etel do
23
       pipe_through :api
24
25
     end
26 end
```

# There is no need to trade beautiful, maintainable code to gain performance

#### Demo