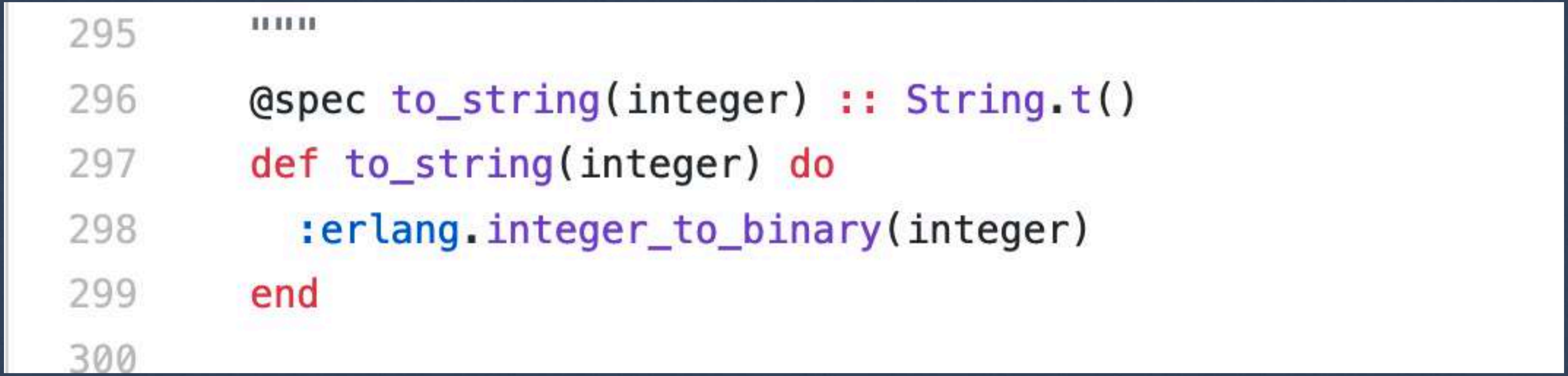
It’s Time to Embrace Erlang

Todd Resudek

- $ Software Engineer at SimpleBet
- Hex Core Team
- @supersimple
- sprsmpl

“Elixir is the solution to Erlang syntax”

Elixir Integer Module



```
"""
@spec to_string(integer) :: String.t()
def to_string(integer) do
  :erlang.integer_to_binary(integer)
end

```

NYPD
ERLANG

Erlang Code

```erlang
%% OTP 21+
-ifdef (OTP_RELEASE).
compose_query(Pairs) ->
    uri_string:compose_query(Pairs).
-else.
compose_query(Pairs) ->
    String = join("&", lists:map(fun ({K, V}) -> K ++ "=" ++ V end, Pairs)),
    http_uri:encode(String).
-endif.
```

Problem 1

The approach

The approach

1. Attracts those that tried Erlang but did not love it.
2. Offends and alienates those already invested in Erlang.

The approach

1. Those with the most knowledge of the BEAM weren’t interested.
2. 2 separate groups working in parallel and not together.
3. Community growth was limited.

Problem 2

We go out of our way to avoid even a minimal amount of Erlang

…Elixir discourages simply wrapping Erlang libraries in favor of directly interfacing with Erlang code.

Do you have a minute to talk about HTTPoison?

HTTP Request

```
HTTPoison.request(
  method,
  url,
  body,
  headers,
  options
)
```

HTTP Request

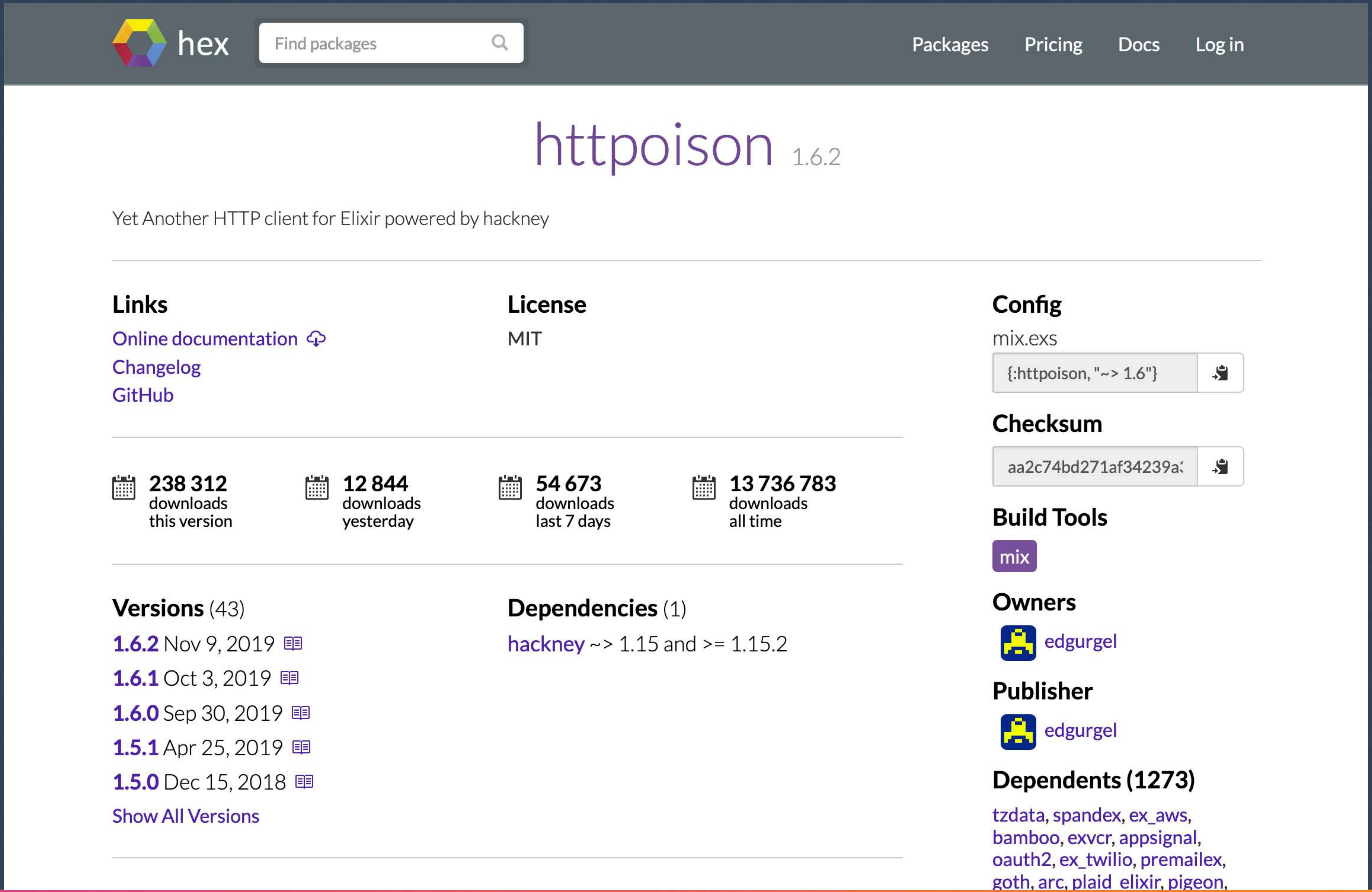
```
:hackney:request(
  method,
  url,
  headers,
  payload,
  options
)
```

HTTP Request

```elixir
defp do_request(request, hn_options) do
  :hackney.request(
     request.method,
     request.url,
     request.headers,
     request.body,
     hn_options
   )
end
```

ERLANG

HTTPoison



httpoison 1.6.2

Yet Another HTTP client for Elixir powered by hackney

Links

Online documentation

Changelog

GitHub

License

MIT

238 312 downloads this version

12 844 downloads yesterday

54 673 downloads last 7 days

13 736 783 downloads all time

Versions (43)

- **1.6.2** Nov 9, 2019
- **1.6.1** Oct 3, 2019
- **1.6.0** Sep 30, 2019
- **1.5.1** Apr 25, 2019
- **1.5.0** Dec 15, 2018

Show All Versions

Dependencies (1)

hackney ~> 1.15 and >= 1.15.2

Config

mix.exs

```
{:httpoison, "~> 1.6"}
```

Checksum

```
aa2c74bd271af34239a
```

Build Tools

mix

Owners

edgurgel

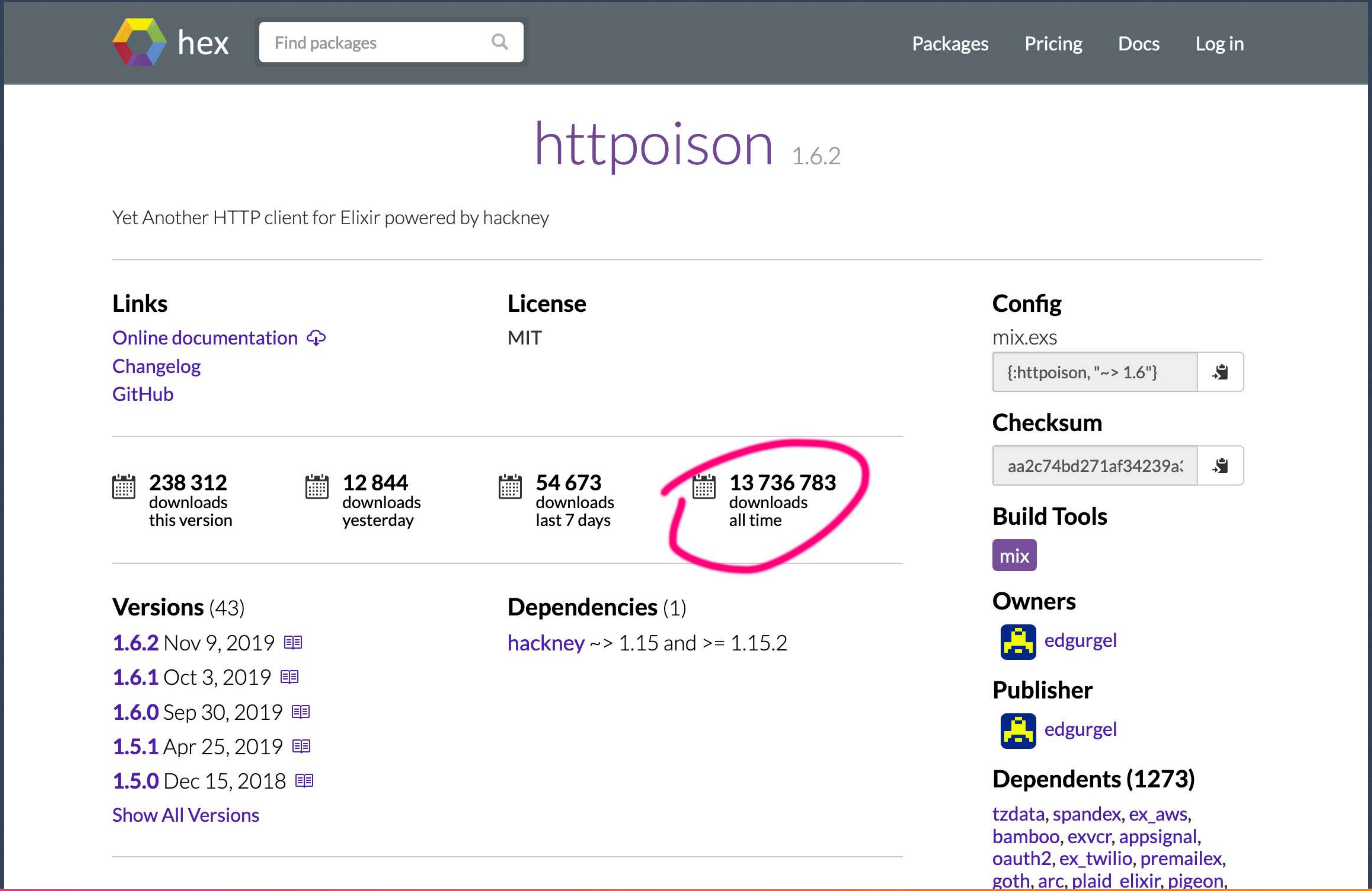
Publisher

edgurgel

Dependents (1273)

tzdata, spandex, ex_aws, bamboo, exvcr, appsignal, oauth2, ex_twilio, premailex, goth, arc, plaid_elixir, pigeon,

HTTPoison



httpoison 1.6.2

Yet Another HTTP client for Elixir powered by hackney

Links
Online documentation
Changelog
GitHub

License
MIT

238 312 downloads this version
12 844 downloads yesterday
54 673 downloads last 7 days
13 736 783 downloads all time

Versions (43)
1.6.2 Nov 9, 2019
1.6.1 Oct 3, 2019
1.6.0 Sep 30, 2019
1.5.1 Apr 25, 2019
1.5.0 Dec 15, 2018
Show All Versions

Dependencies (1)
hackney ~> 1.15 and >= 1.15.2

Config
mix.exs
{:httpoison, "~> 1.6"}

Checksum
aa2c74bd271af34239a

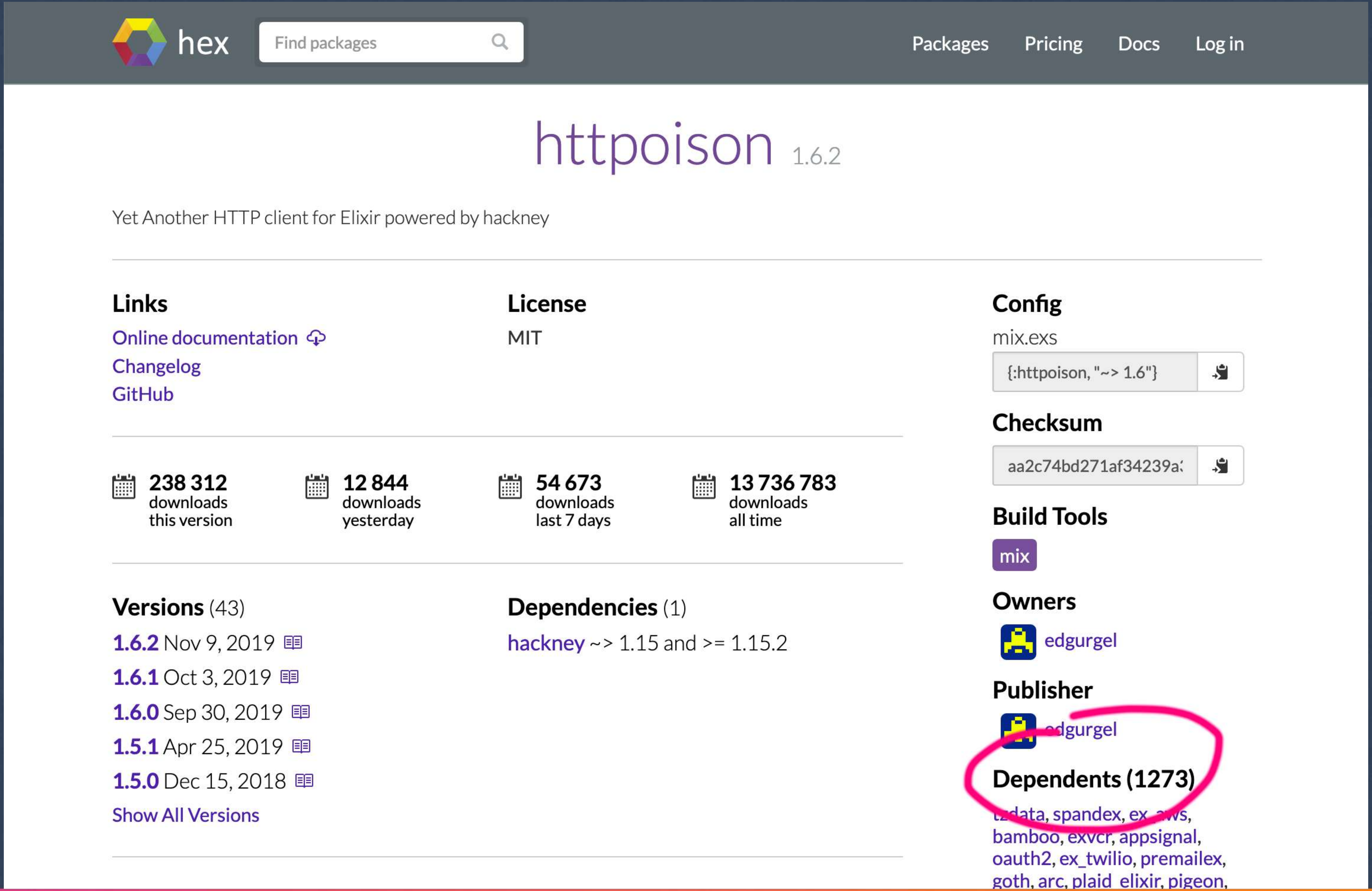
Build Tools
mix

Owners
edgurgel

Publisher
edgurgel

Dependents (1273)
tzdata, spandex, ex_aws, bamboo, exvcr, appsignal, oauth2, ex_twilio, premailex, goth, arc, plaid_elixir, pigeon,

HTTPoison



httpoison 1.6.2

Yet Another HTTP client for Elixir powered by hackney

Links

Online documentation
Changelog
GitHub

License

MIT

238 312 downloads this version

12 844 downloads yesterday

54 673 downloads last 7 days

13 736 783 downloads all time

Versions (43)

1.6.2 Nov 9, 2019
1.6.1 Oct 3, 2019
1.6.0 Sep 30, 2019
1.5.1 Apr 25, 2019
1.5.0 Dec 15, 2018
Show All Versions

Dependencies (1)

hackney ~> 1.15 and >= 1.15.2

Config

mix.exs

```
{:httpoison, "~> 1.6"}
```

Checksum

aa2c74bd271af34239a:

Build Tools

mix

Owners

edgurgel

Publisher

edgurgel

Dependents (1273)

tzdata, spandex, ex_aws, bamboo, exvcr, appsignal, oauth2, ex_twilio, premailex, goth, arc, plaid_elixir, pigeon,

Can we talk about crypto?

Dave
MILLION
Yiannimize
@YIANNIMIZE
@YIANNIMIZE1

Encrypting/Hashing

```
:sha256
|> :crypto.hash(“secret”)
|> Base.encode16()
```

ex_crypto

```
def sha256(data) do
  try do
    {:ok, :crypto.hash(:sha256, data)}
  rescue
    e in ArgumentError -> {:error, e}
end
```

ex_crypto

```
Error:notsup >> {:error, error}
```

An Elixir wrapper for an Erlang library should be opinionated.

Problem 3

To be great in Elixir, you must know at least some Erlang

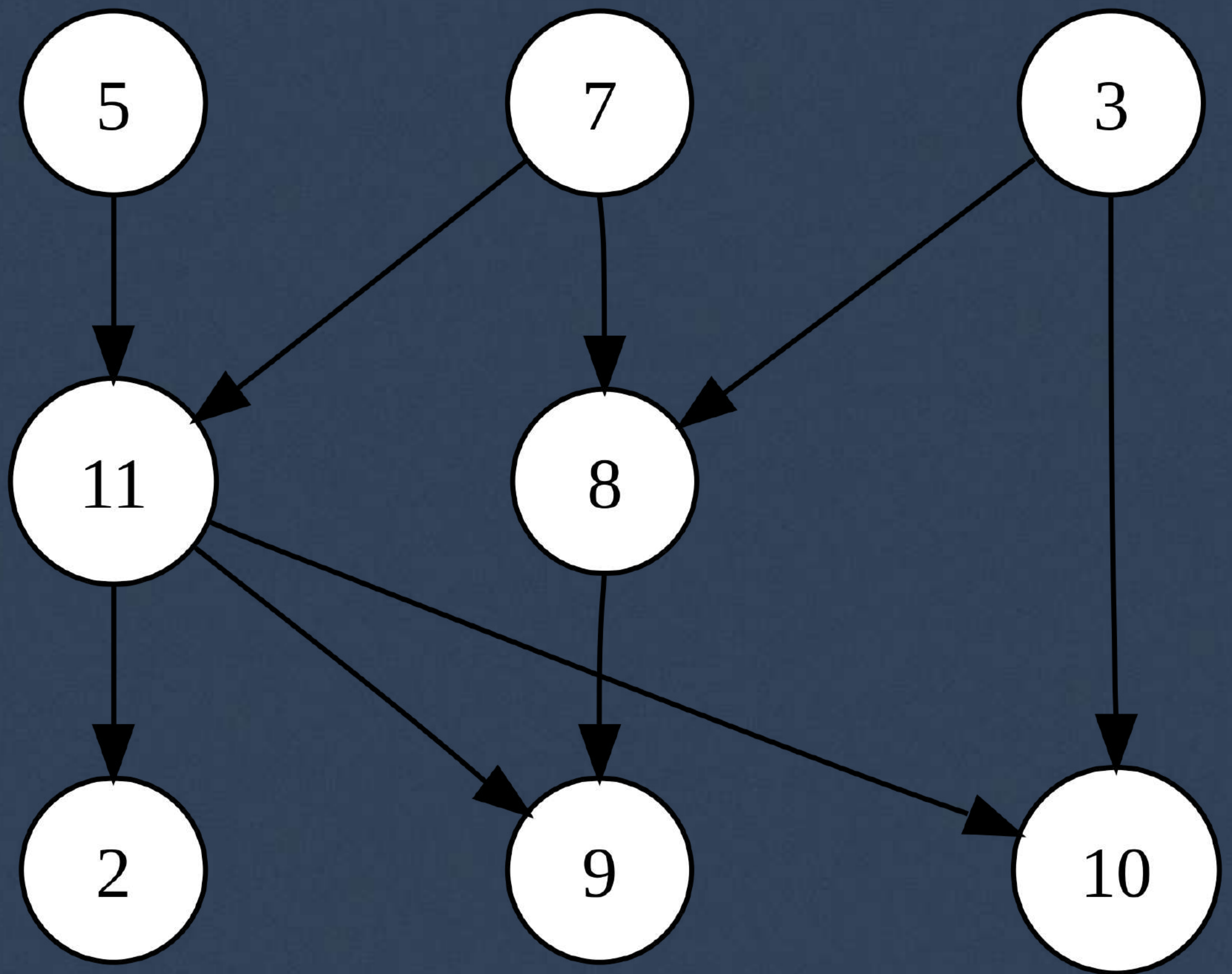
…As you grow more proficient in Elixir, you may want to explore the Erlang STDLIB Reference Manual in more detail.

Some features not in Elixir

- Binary module - useful when you are dealing with binary data that is not necessarily UTF-8 encoded
- Formatted text output - Elixir does not contain a function similar to printf
- ETS and DETS - handle storage of large data structures in memory or on disk
- Digraph Module - contains functions for dealing with directed graphs built of vertices and edges

In mathematics, and more specifically in graph theory, a directed graph (or digraph) is a graph that is made up of a set of vertices connected by edges, where the edges have a direction associated with them.

Directed Graph



Edges, edges, edges
SLICE

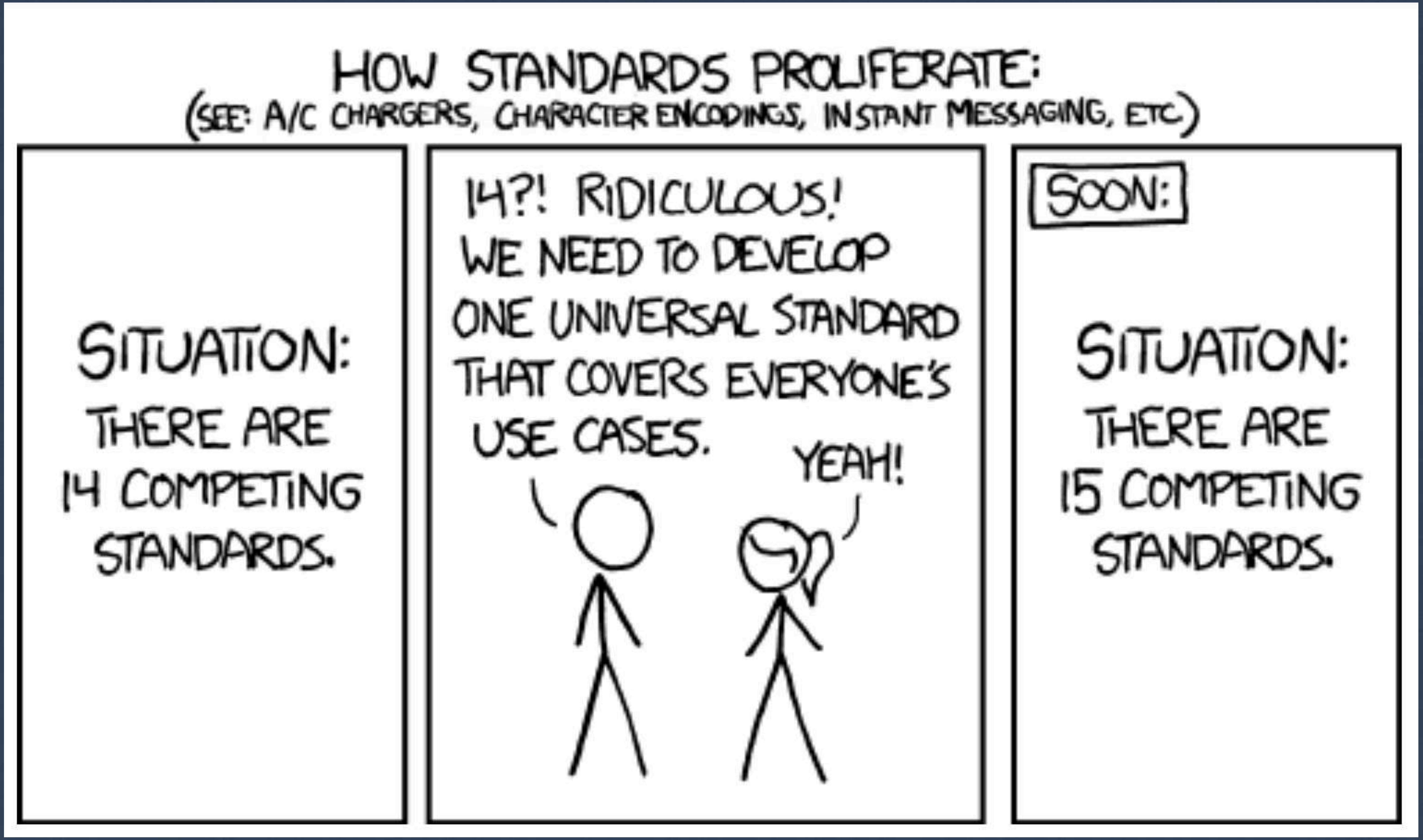
Other features not in Elixir

- Math Module - contains common mathematical operations covering trigonometry, exponential, and logarithmic functions
- Queue Module - implements FIFO queues efficiently
- Rand Module - has functions for returning random values and setting the random seed
- Zip and Zlib modules - read a write compressed files

Go lower in the stack

Elixir -> Erlang -> Core Erlang -> Kernel -> SSA -> BEAM

Telemetry


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Telemetry

1. Originally written in Elixir
2. Decision was made to re-write in Erlang

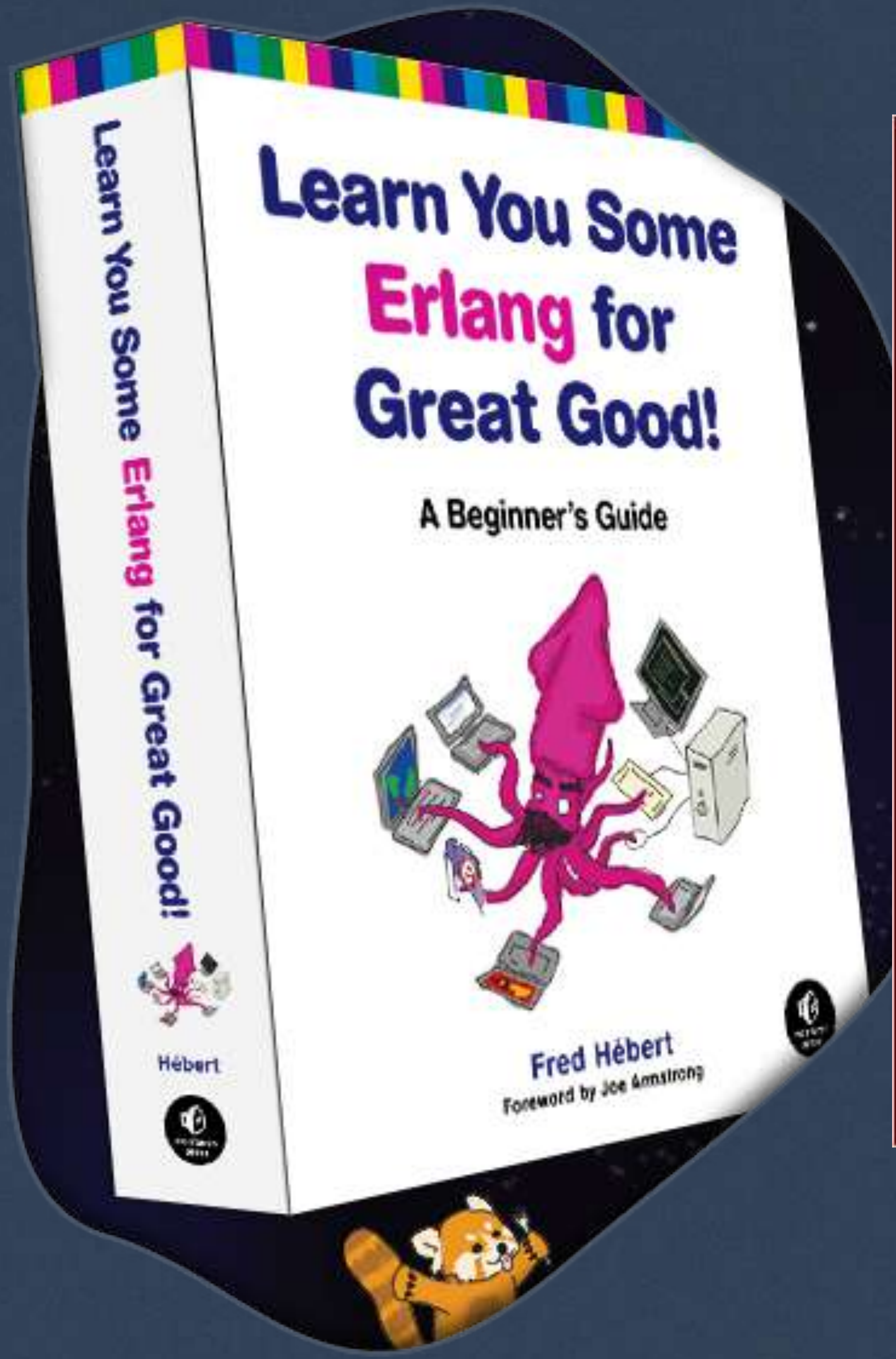
Still not convinced?

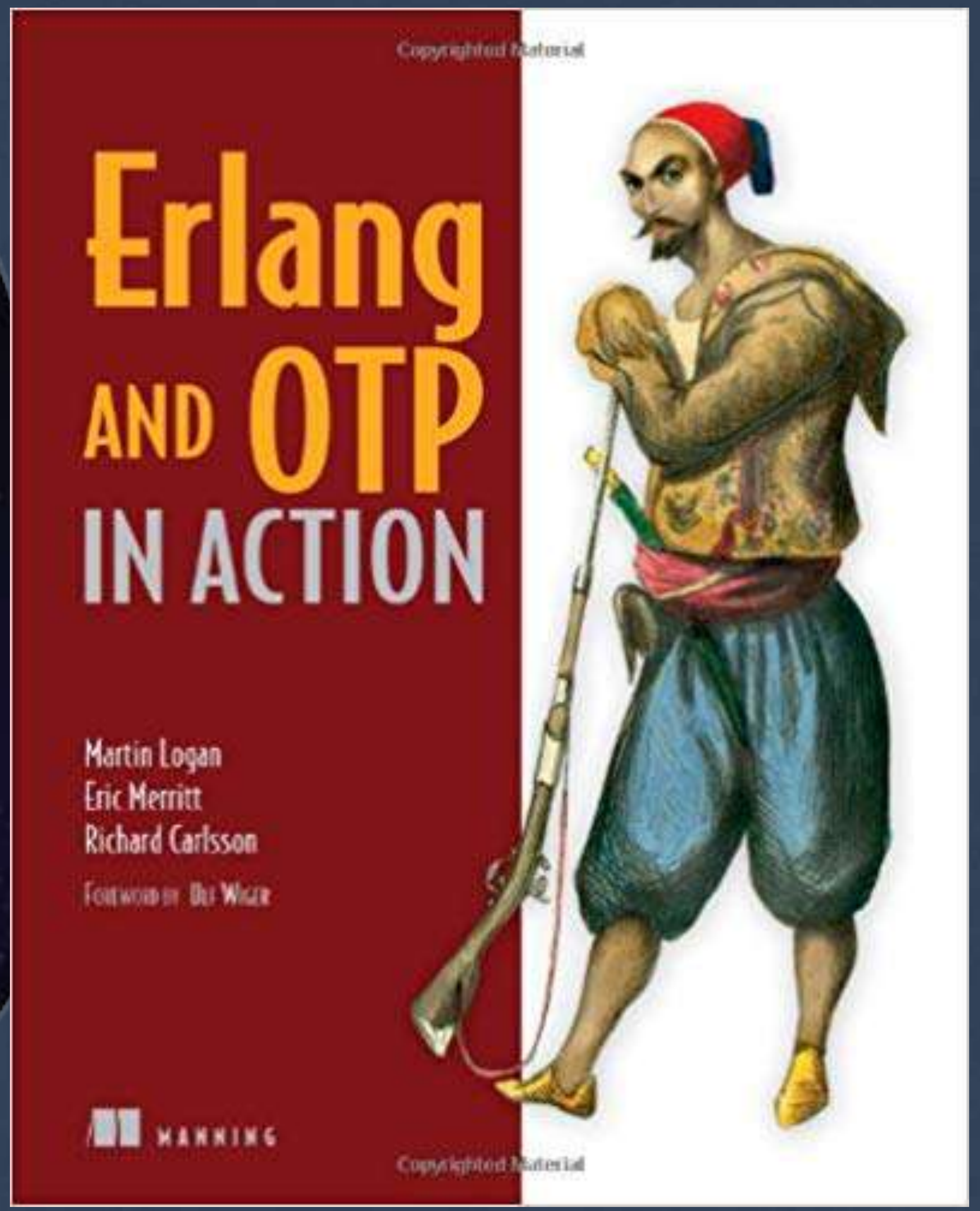
It’s from Sweden

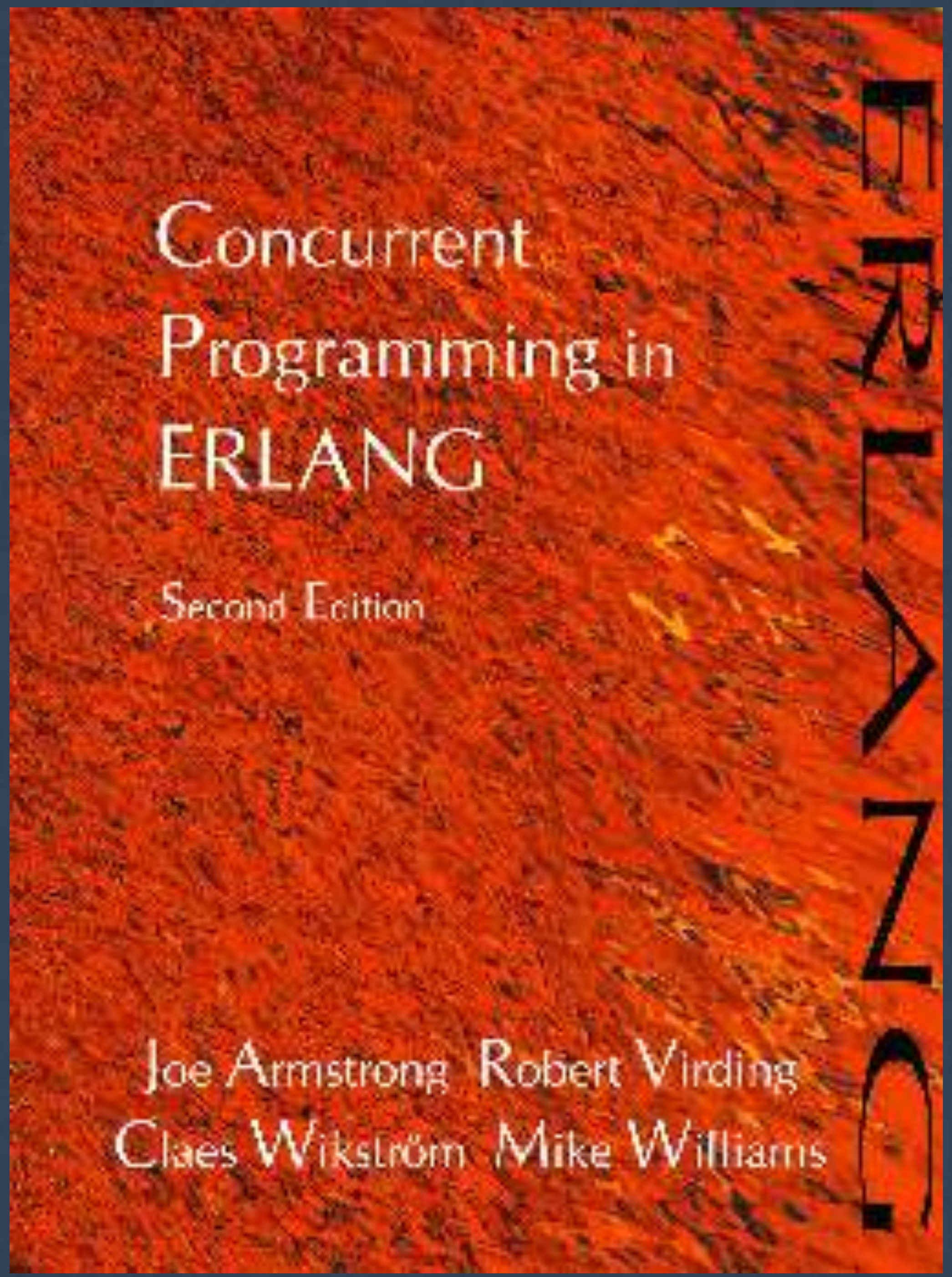
GRAMMY AWARDS
ERICSSON
YES
NO
GH688
CCX

How do I get started?

Resources


Learn You Some Erlang for Great Good!
A Beginner's Guide
Fred Hébert
Foreword by Joe Armstrong
Hébert

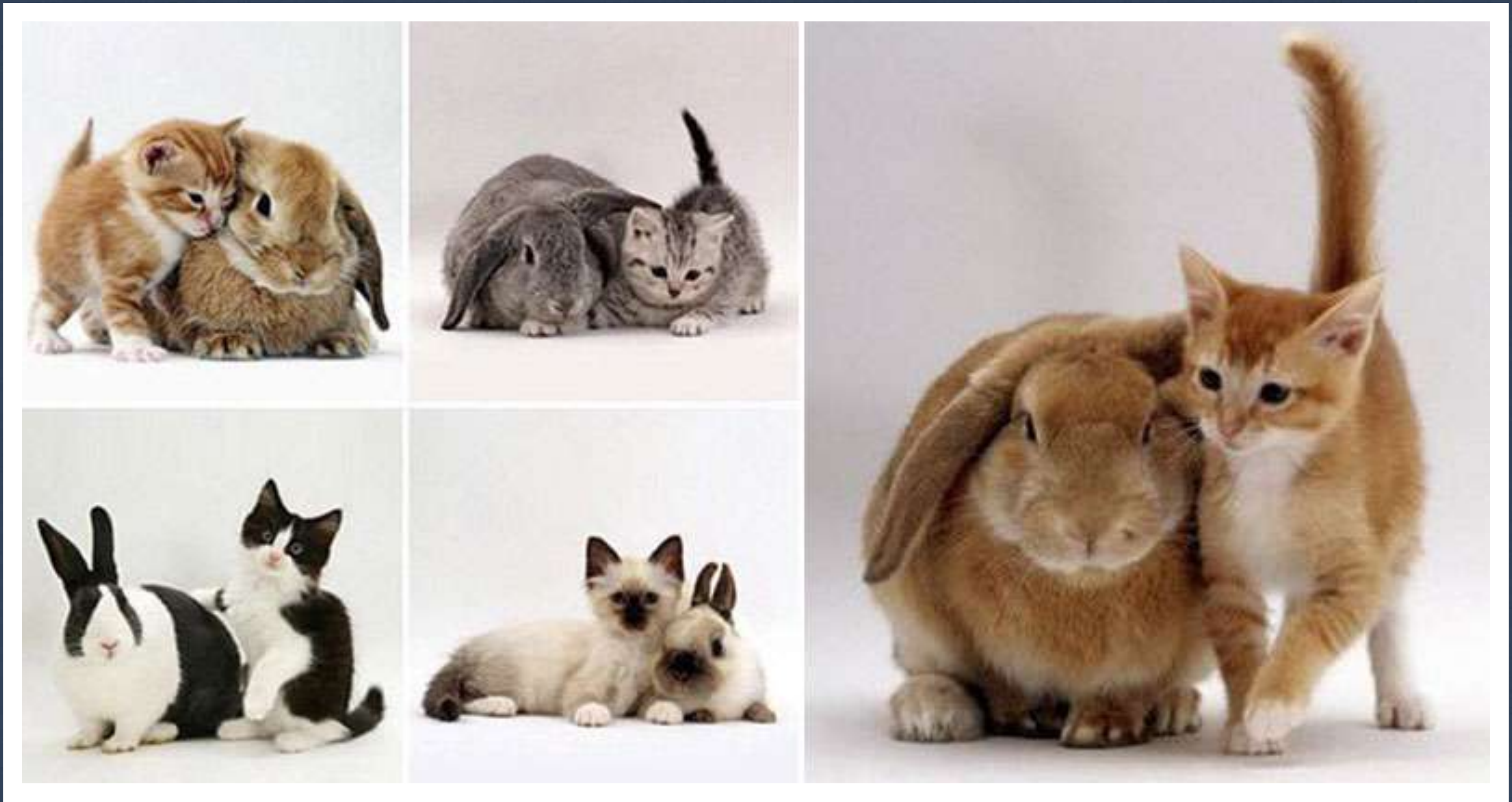

Erlang AND OTP IN ACTION
Martin Logan
Eric Merritt
Richard Carlsson
FOREWORD BY ULF WIGER
MANNING
Copyrighted Material


Concurrent Programming in ERLANG
Second Edition
Joe Armstrong Robert Virding
Claes Wikström Mike Williams
ERLANG

Adopting Erlang
?

Community

1. Slack channel (erlanger.slack.com)
2. Incorporate “Erlang” into your local meetup title
3. Make some Erlanger friends



Thank You!

$impleBet