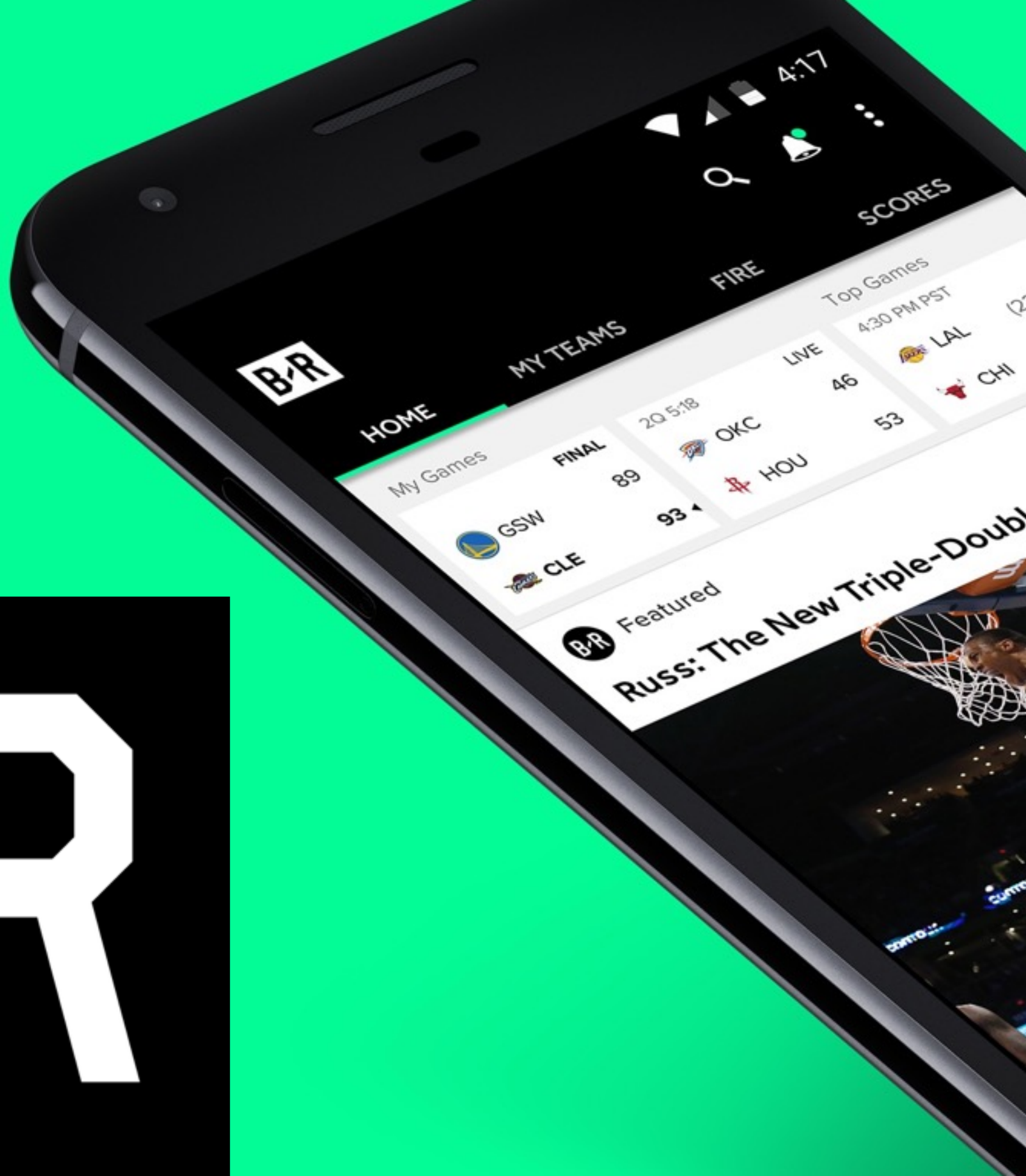
ELIXIR ADOPTION AT SCALE

BEN MARX

@bgmarx

LEAD DEVELOPER

B-R



STATS

>15M APP DOWNLOADS

1.5B VIEWS/MONTH

>16B VIEWS/YEAR

>3B NOTIFICATIONS/MONTH

IN THE BEGINNING

OCTOBER 2014

REASONABLE EXPECTATIONS

CONSISTENT

CODE

TRAINING DEVELOPERS

REASONABLE EXPECTATIONS

FEATURE DEV SLOWS

PIECE

BY

PIECE

NGINX

HAPROXY

TERRAFORM

https://github.com/poteto/terraform/

API

BLOAT

TRANSLATION LAYER

NEW

LEGACY

HAPPY

CONSISTENT

CODE

MESSY ELIXIR CODE

```
def for(url) do
  url = prepend_http(url)

  {_, response } =
    url
    |> discover
    |> HTTPoison.get([], @redirect)
  case response do
    %HTTPoison.Response{status_code: 200, body:
body} ->
      decoded_body =
        Poison.decode!(body)
        |> strip_newline_chars

      cond do
          is_twitter?(url) ->
            format_twitter_response(decoded_body)
          is_instagram?(url) ->
```

CREDO

https://github.com/rrrene/credo

```
def for(url) do
  url
  |> format_url
  |> get_embed_data
  |> format_by_content_type
end
```

```
defp format_url(url) do
 # standardizes url based on certain requirements
end
```

```
defp get_embed_data(response) do
  # attempts to extract embed data from external source
end
```

```
defp format_by_content_type(:content_type, response) do
  # formats responses and handles errors
end
```

TYPESPECS & DIALYZER

```
@type embed :: map()

@spec for(String.t) :: embed
def for(url) do
  url
  |> parse_response
  |> format_by_content_type
end
```

DOCUMENTATION

```
@doc """
Attempts to fetch and extract embed
data for a url and format it
according to the defined content type
"""
@spec for(String.t) :: embed
def for(url) do
  url
  |> get_embed_data
  |> format_by_content_type
end
```

```
@doc """
Attempts to fetch and extract embed data for a
url and format it according to the defined
content type
"""
@spec fetch_and_extract_embed(String.t) :: embed
def fetch_and_extract_embed(url) do
  url
  |> get_embed_data
  |> format_by_content_type
end
```

CODE REVIEWS

Erlang indeeds facilitates solving many of these problems. But at the end of the day, it's still just a progamming language.

Designing for Scalability with Erlang/OTP

STATIC ANALYSIS

TYPESPECS

DOCUMENTATION

TESTING

SERVICE REVIEW

DATABASE PERF

LOGGING

MONITORING

FOCUS ON THE CODE

TRAINING DEVELOPERS

EXPECT

PUSHBACK

MAKE IT EASY

TESTS MATTER

HOW TO MEASURE SUCCESS?

ONGOING PROCESS

MISTAKES HAPPEN

RAPID DEVELOPMENT

CONTENT TYPES

```
def insert_changeset(struct,
params \\ %{}, content_type) do
  struct
  |> cast(params, @fields)
  |> cast_content_type(params,
content_type)
  # omitted
```

```
def cast_changeset(“article”, #omitted) do
  changeset
  |> # validations omitted
end
def cast_changeset("gif", #omitted) do
  changeset
  |> # validations omitted end
def cast_changeset(“photo”, #omitted) do
  changeset
  |> # validations omitted
end
```

NUMBER OF CONTRIBUTORS

APPLICATION	NUMBER OF CONTRIBUTORS	PROJECT AGE
MONOLITH	59	BEGINNINNG OF TIME

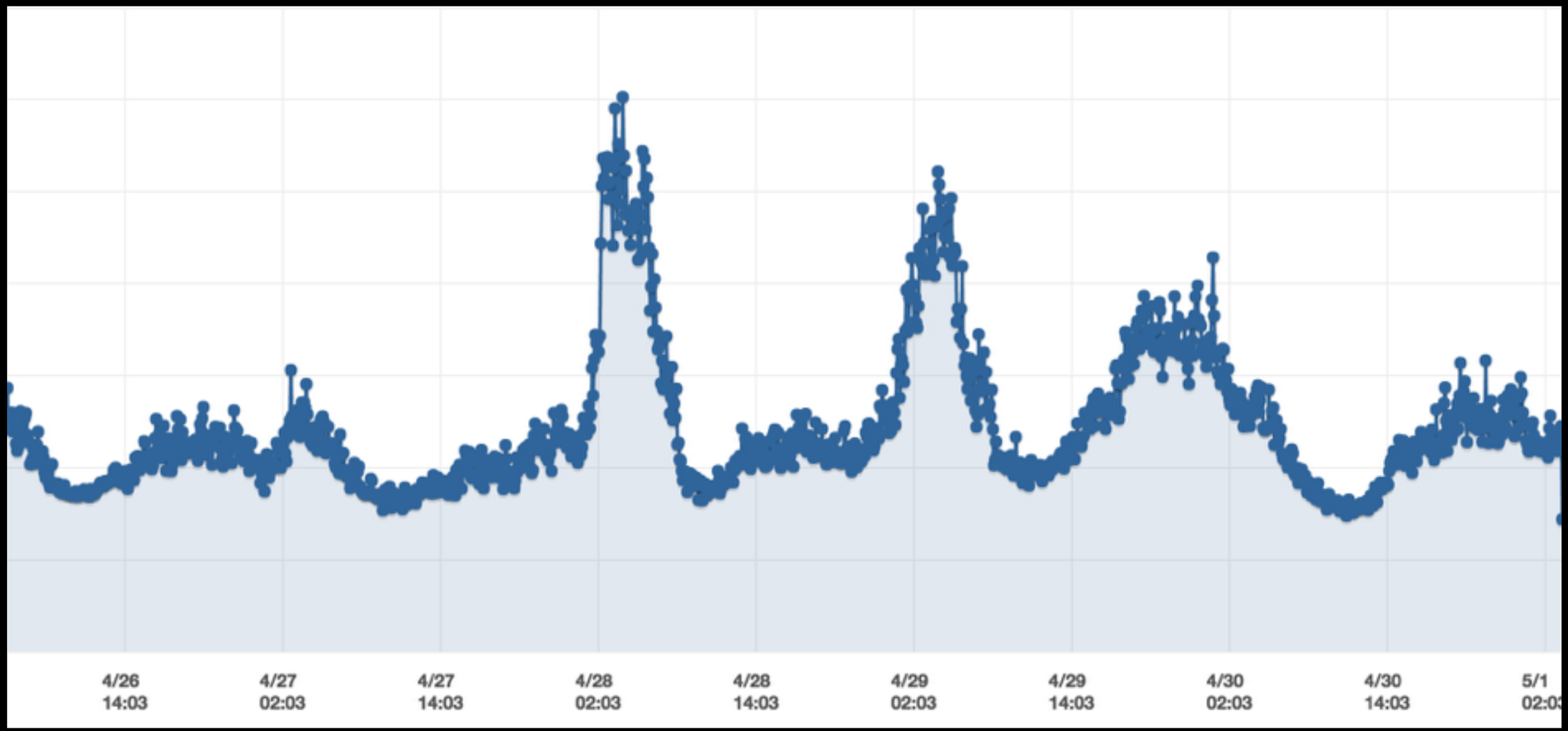
APPLICATION	NUMBER OF CONTRIBUTORS	PROJECT AGE
FIRST ELIXIR APP POWERS CLIENTS	23	2.5 YEARS
OEMBED	12	2 YEARS
METADATA	10	8 MONTHS

NFL DRAFT

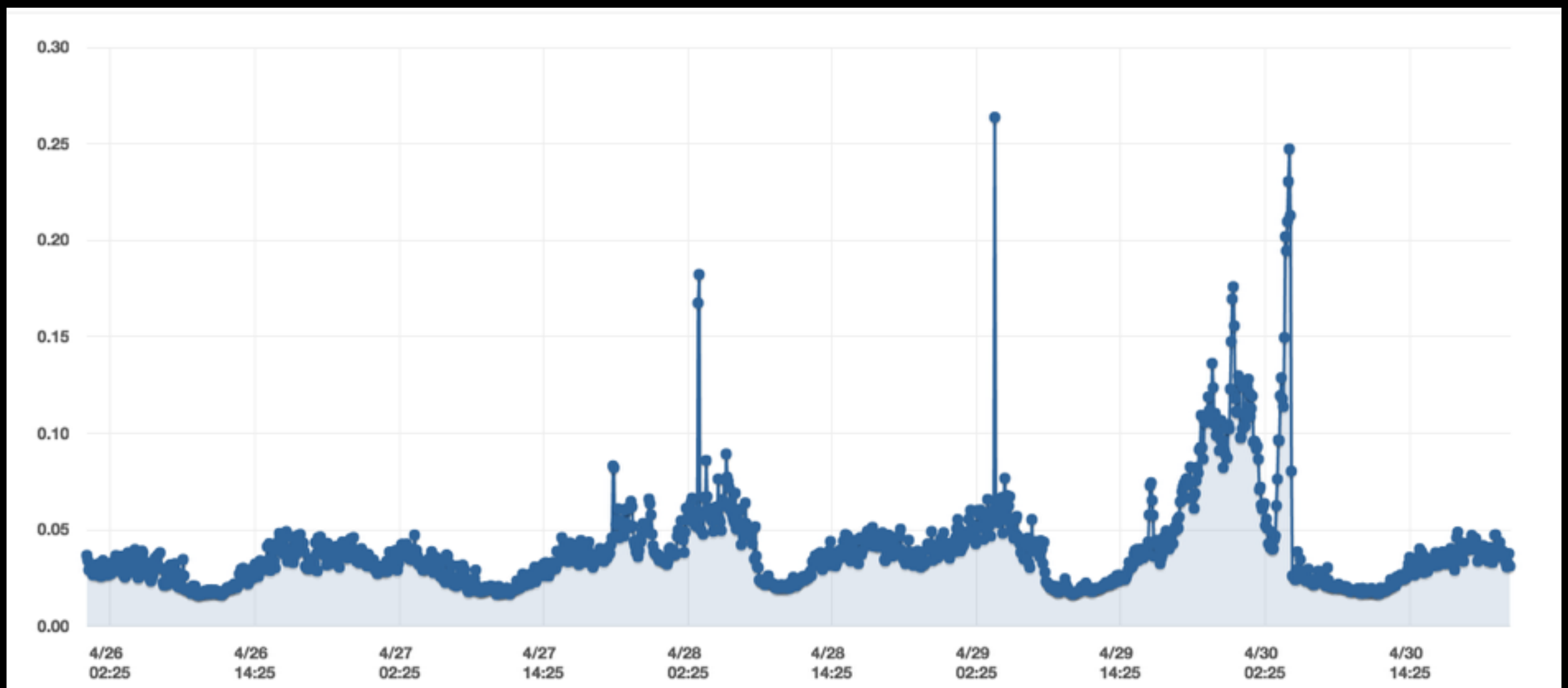
B/R STATE OF THE UNION

HIGHEST CONCURRENTS

SPIKES


4/26
14:03
4/27
02:03
4/27
14:03
4/28
02:03
4/28
14:03
4/29
02:03
4/29
14:03
4/30
02:03
4/30
14:03
5/1

LATENCIES


0.30
0.25
0.20
0.15
0.10
0.05
0.00
4/26 02:25
4/26 14:25
4/27 02:25
4/27 14:25
4/28 02:25
4/28 14:25
4/29 02:25
4/29 14:25
4/30 02:25
4/30 14:25

SPEND TIME DEVELOPING

The Pragmatic Programmers

Your Elixir Source

Adopting Elixir

From Concept to Production

Ben Marx, José Valim, Bruce Tate

edited by Jacquelyn Carter

QUESTIONS?

@bgmarx