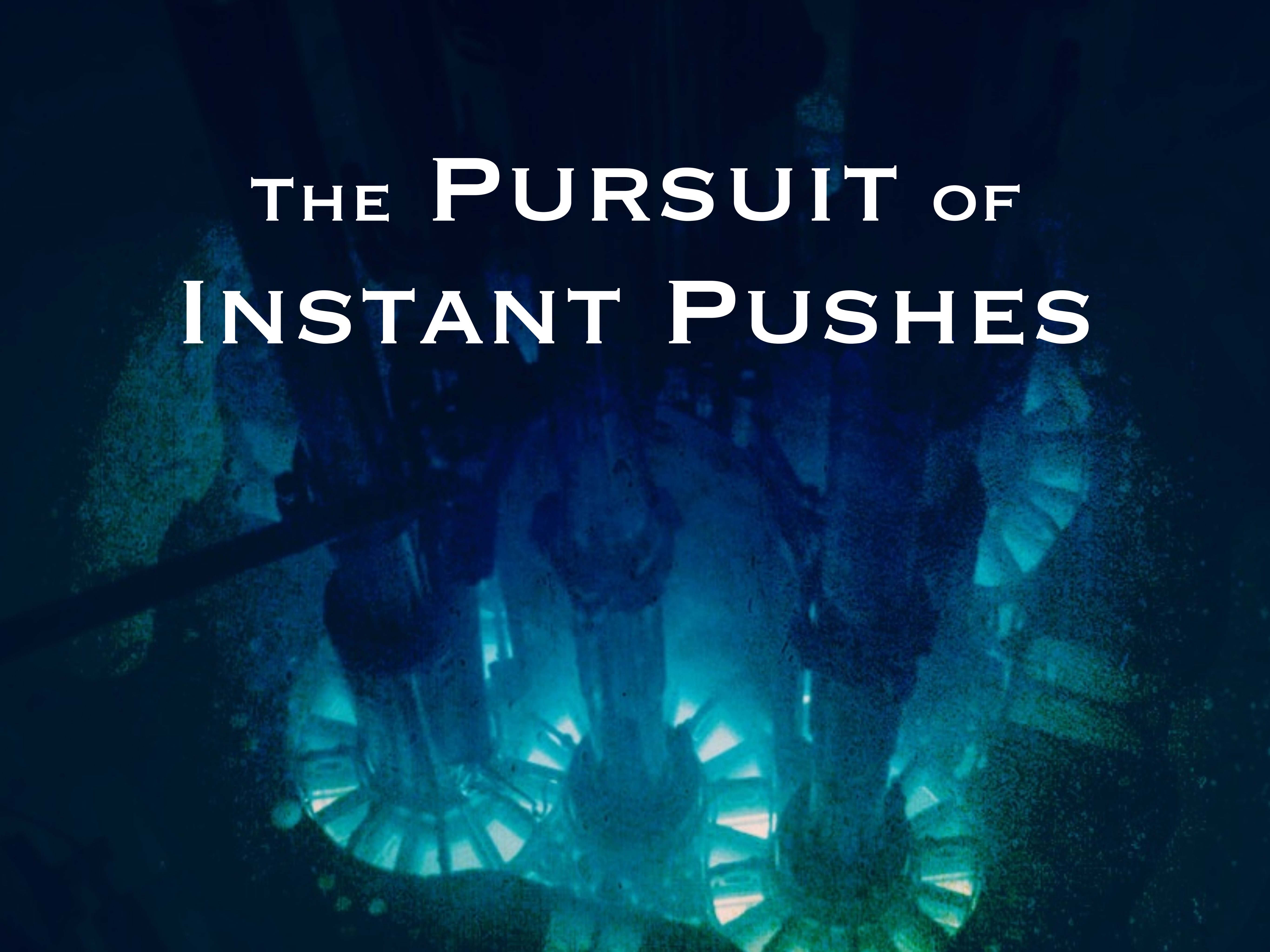
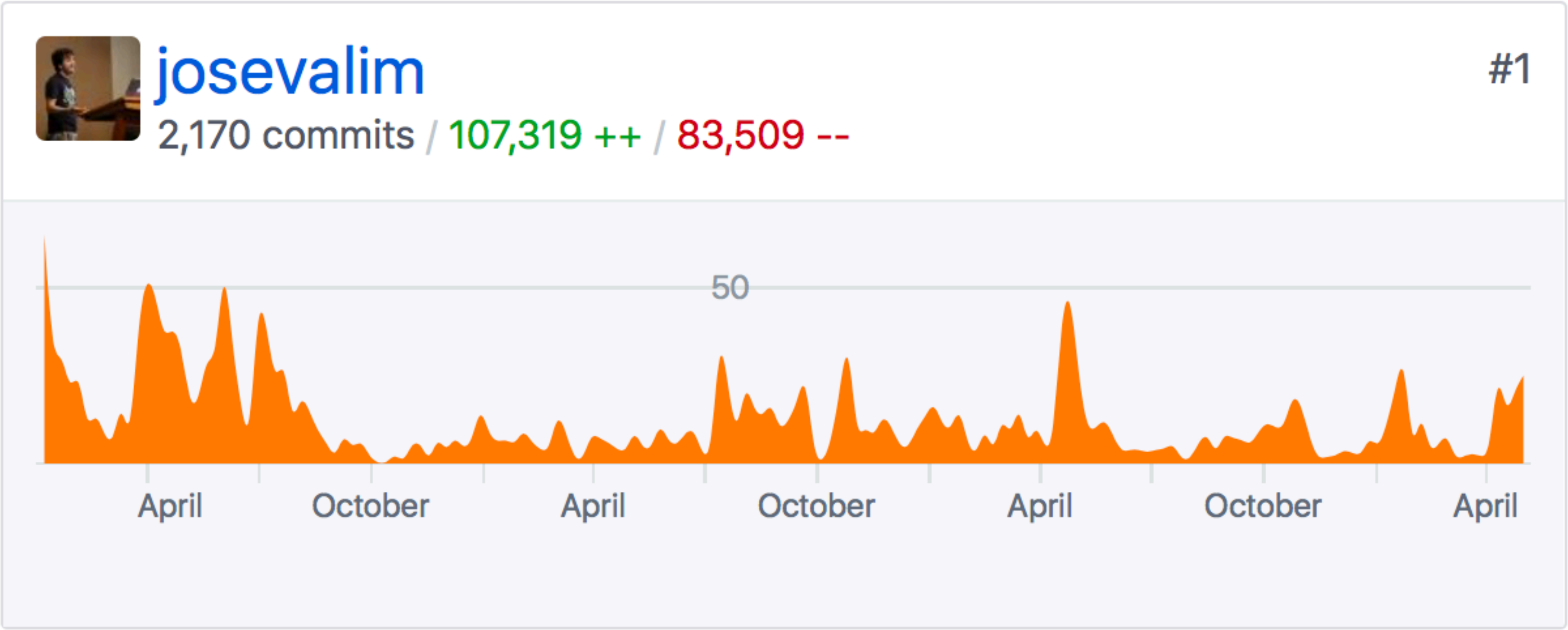
The Pursuit of Instant Pushes



Aleksei Magusev
@lexmag

FORZA FOOTBALL

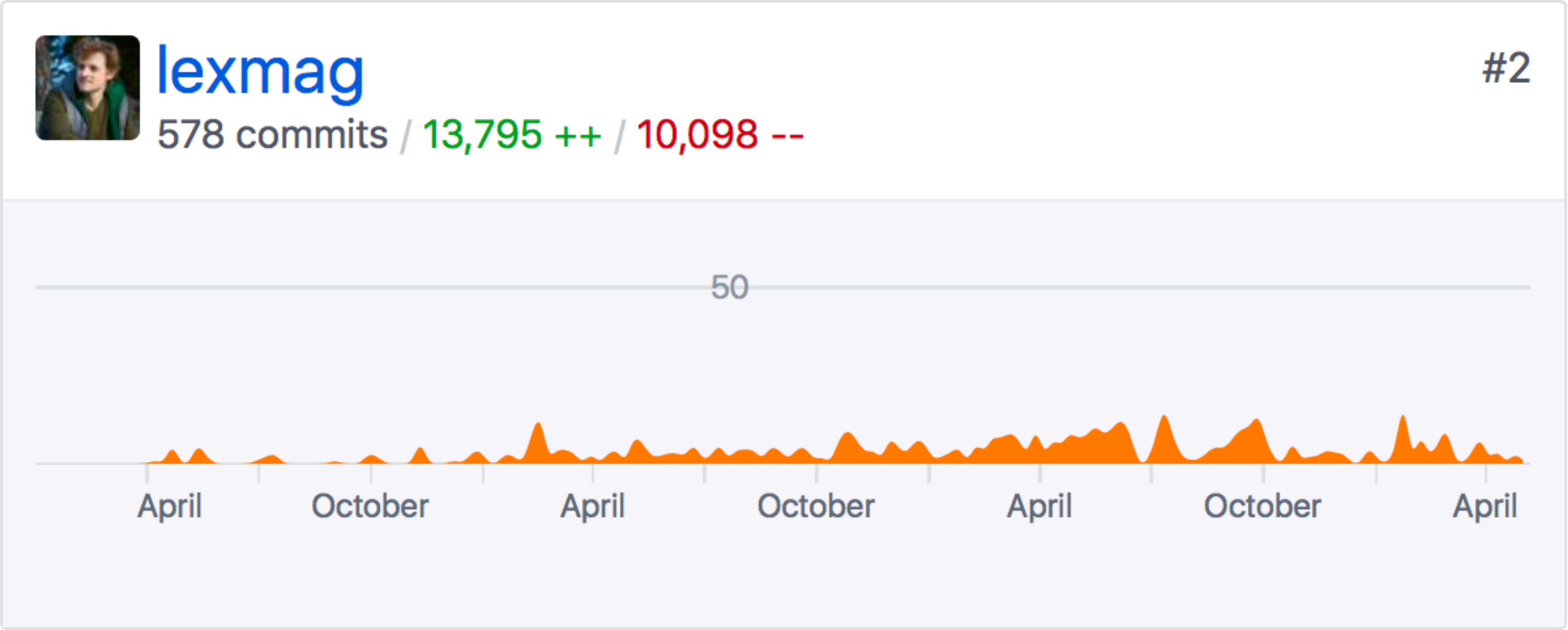
Trying to take over Elixir (not really)



josevalim

#1

2,170 commits / 107,319 ++ / 83,509 --



lexmag

#2

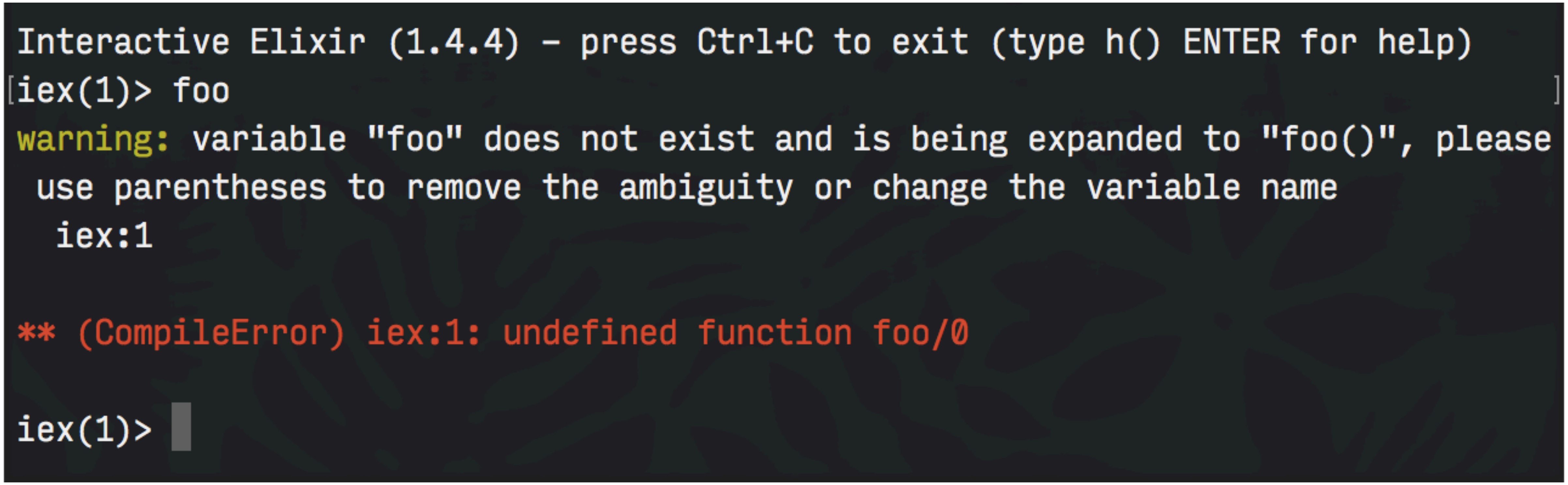
578 commits / 13,795 ++ / 10,098 --

Keeping Elixir in style

lexmag/elixir-style-guide

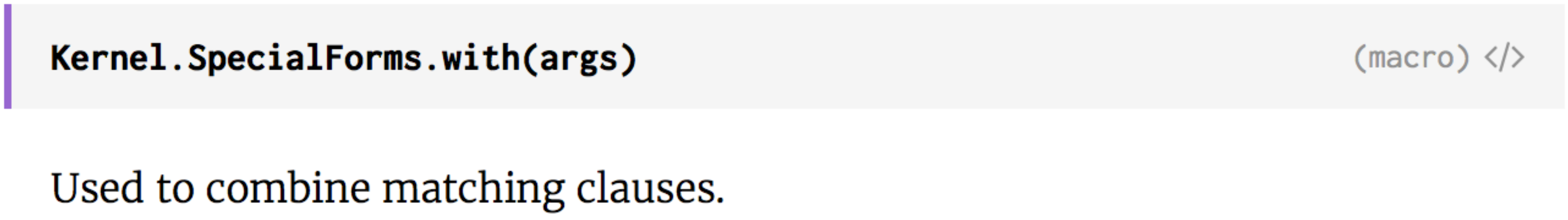
> “This guide is written using the blood of the Elixir maintainers as ink.”
>
> –José Valim

Adding wonderful warnings



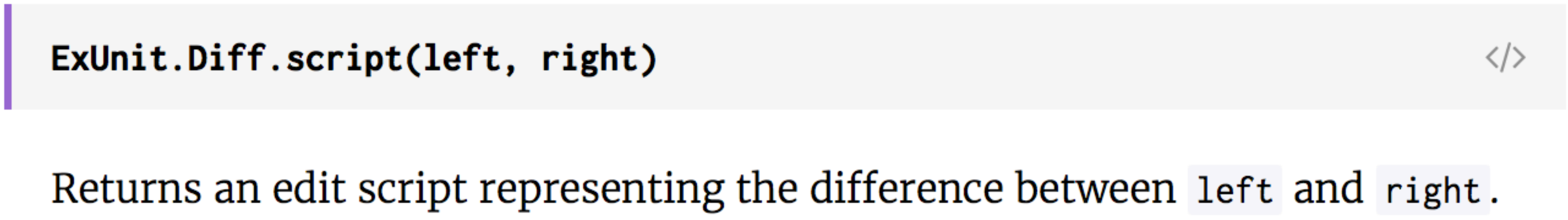
```
Interactive Elixir (1.4.4) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> foo
warning: variable "foo" does not exist and is being expanded to "foo()", please
 use parentheses to remove the ambiguity or change the variable name
  iex:1

** (CompileError) iex:1: undefined function foo/0

iex(1)>
```

At times adding features



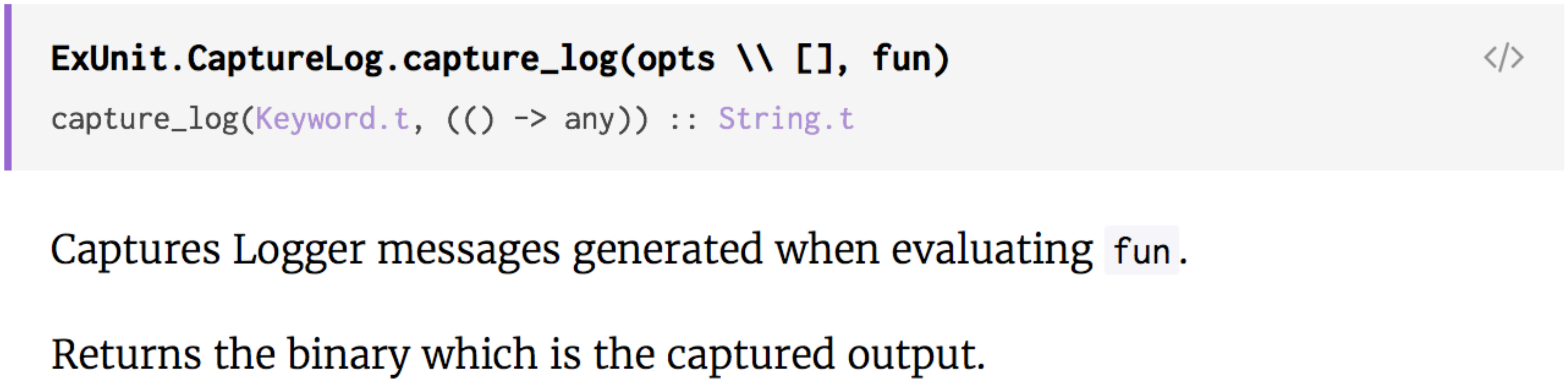
`Kernel.SpecialForms.with(args)` (macro) </>

Used to combine matching clauses.



`ExUnit.Diff.script(left, right)` </>

Returns an edit script representing the difference between `left` and `right`.



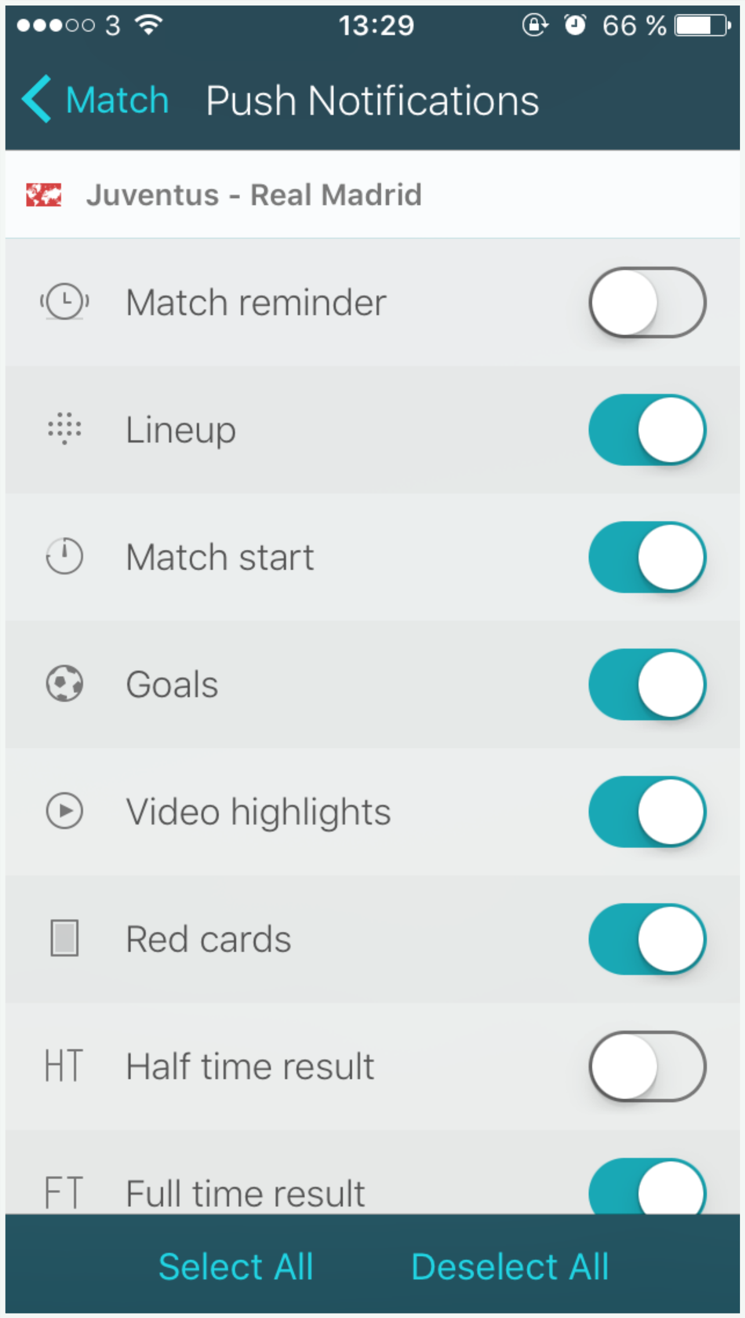
`ExUnit.CaptureLog.capture_log(opts \\ [], fun)` </>

`capture_log(Keyword.t, (() -> any)) :: String.t`

Captures Logger messages generated when evaluating `fun`.

Returns the binary which is the captured output.

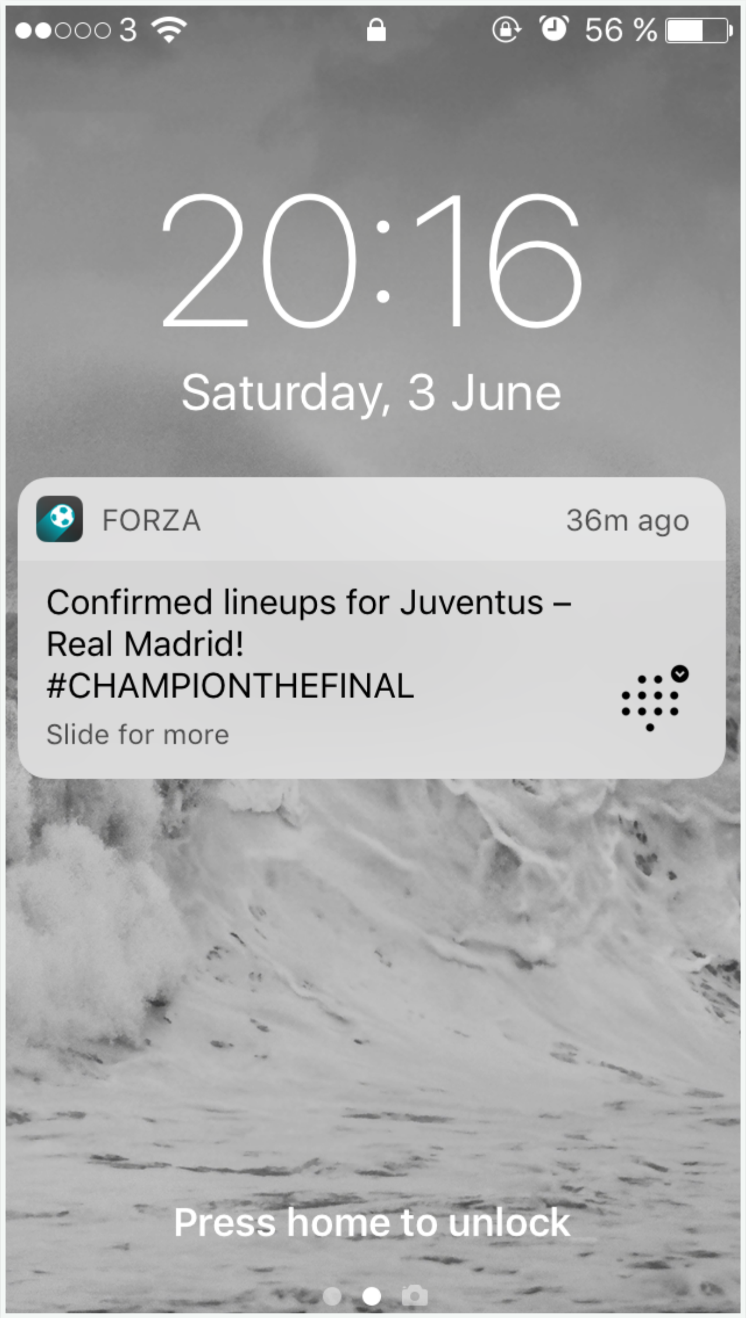
What is this all about?



Match Push Notifications

Juventus - Real Madrid

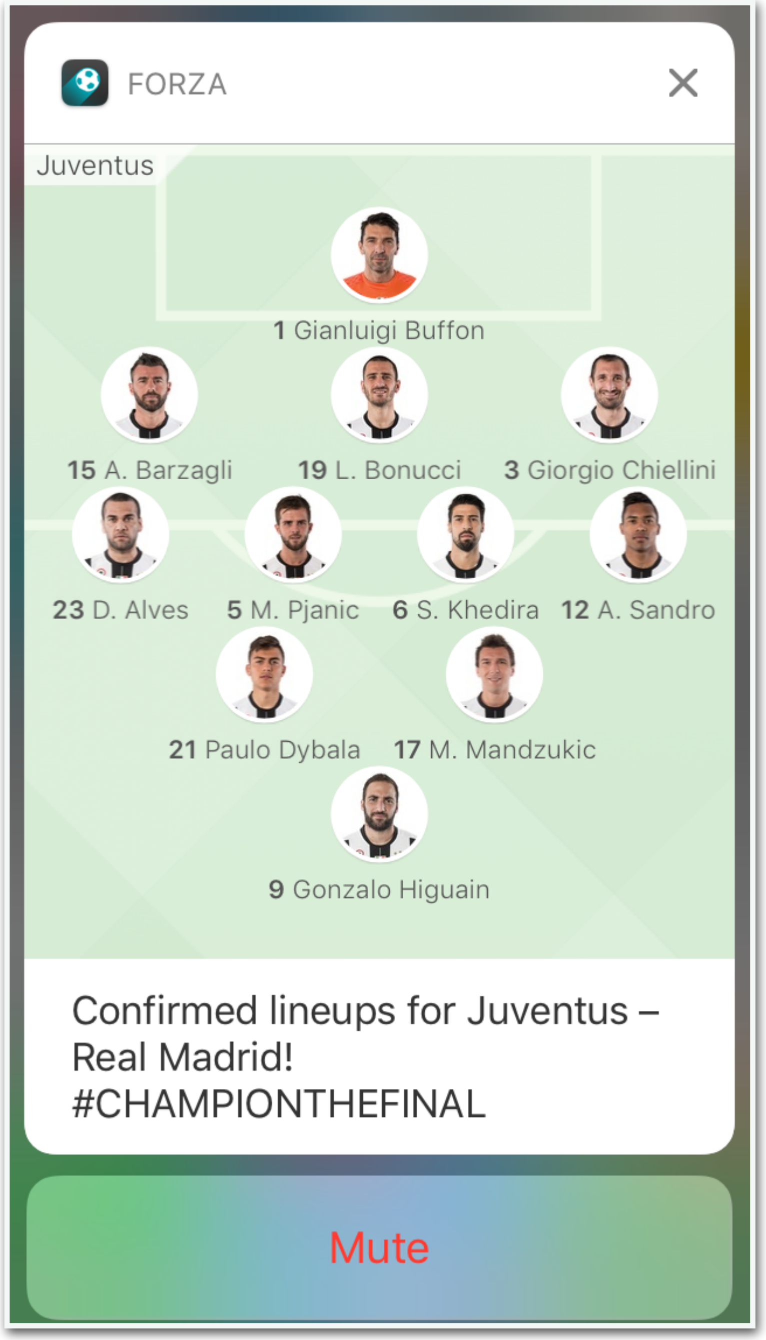
- Match reminder ○
- Lineup ●
- Match start ●
- Goals ●
- Video highlights ●
- Red cards ●
- HT Half time result ○
- FT Full time result ●



FORZA 36m ago

Confirmed lineups for Juventus – Real Madrid! #CHAMPIONTHEFINAL

Slide for more


FORZA

Juventus

1 Gianluigi Buffon

15 A. Barzagli, 19 L. Bonucci, 3 Giorgio Chiellini

23 D. Alves, 5 M. Pjanic, 6 S. Khedira, 12 A. Sandro

21 Paulo Dybala, 17 M. Mandzukic

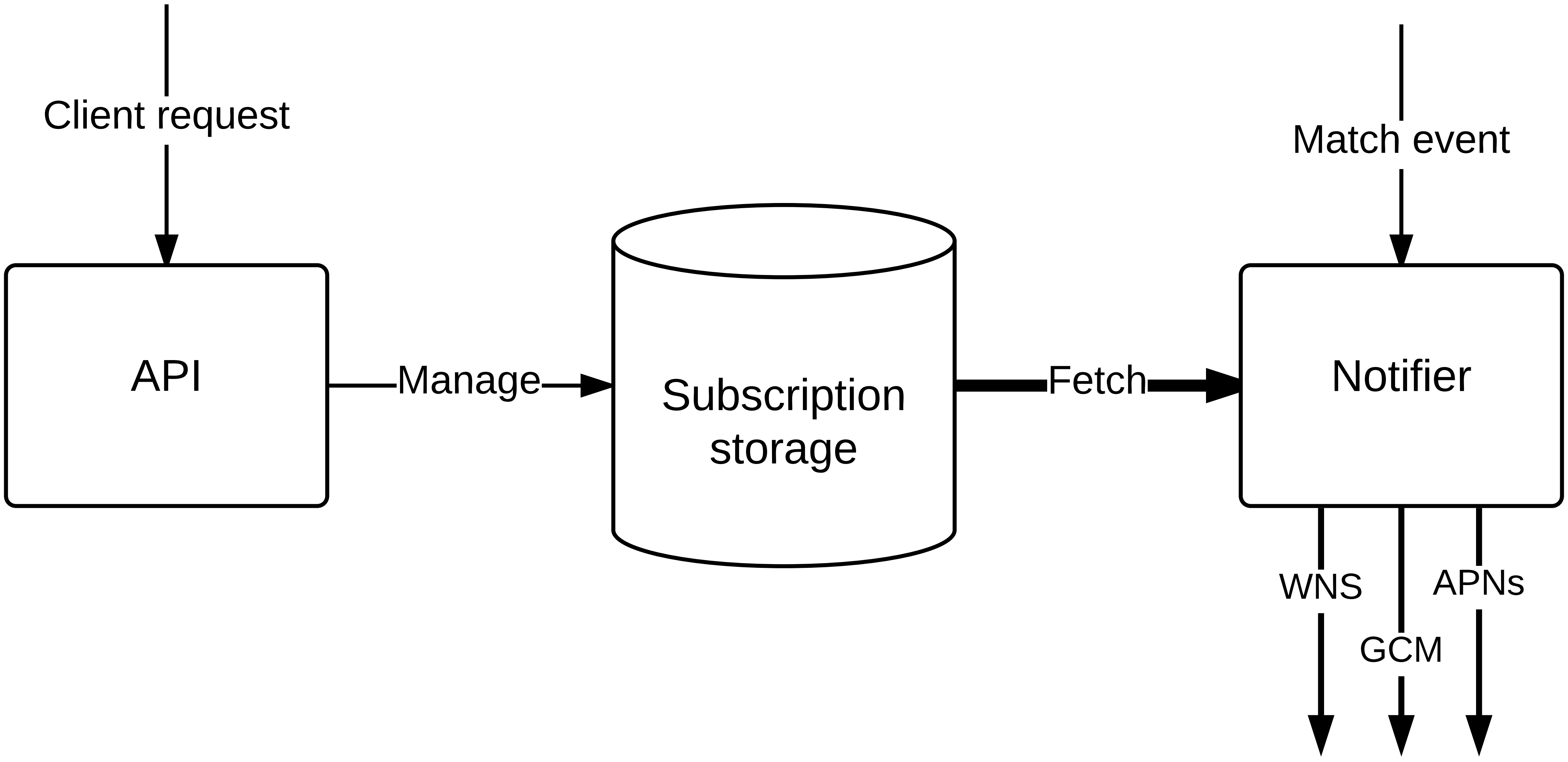
9 Gonzalo Higuain

Confirmed lineups for Juventus – Real Madrid! #CHAMPIONTHEFINAL

What is this all about?

- 3 operating systems
  - iOS
  - Android
  - Windows
- More than 1 billion push notifications each month
- Most popular events have 2–3 million subscribers
- Hundred millions subscriptions are stored
- Push notifications are very time-sensitive

What is this all about?



Subscription handling

Notification sending

Subscription anatomy

device_token	xfL3k6QxHagF...Nf8Y01a
subject	team 42
topic	goal
language	en
country	SE

Subscription anatomy

device_token	xfL3k6QxHagF...Nf8Y01a
subject	team 42
topic	goal
language	en
country	SE

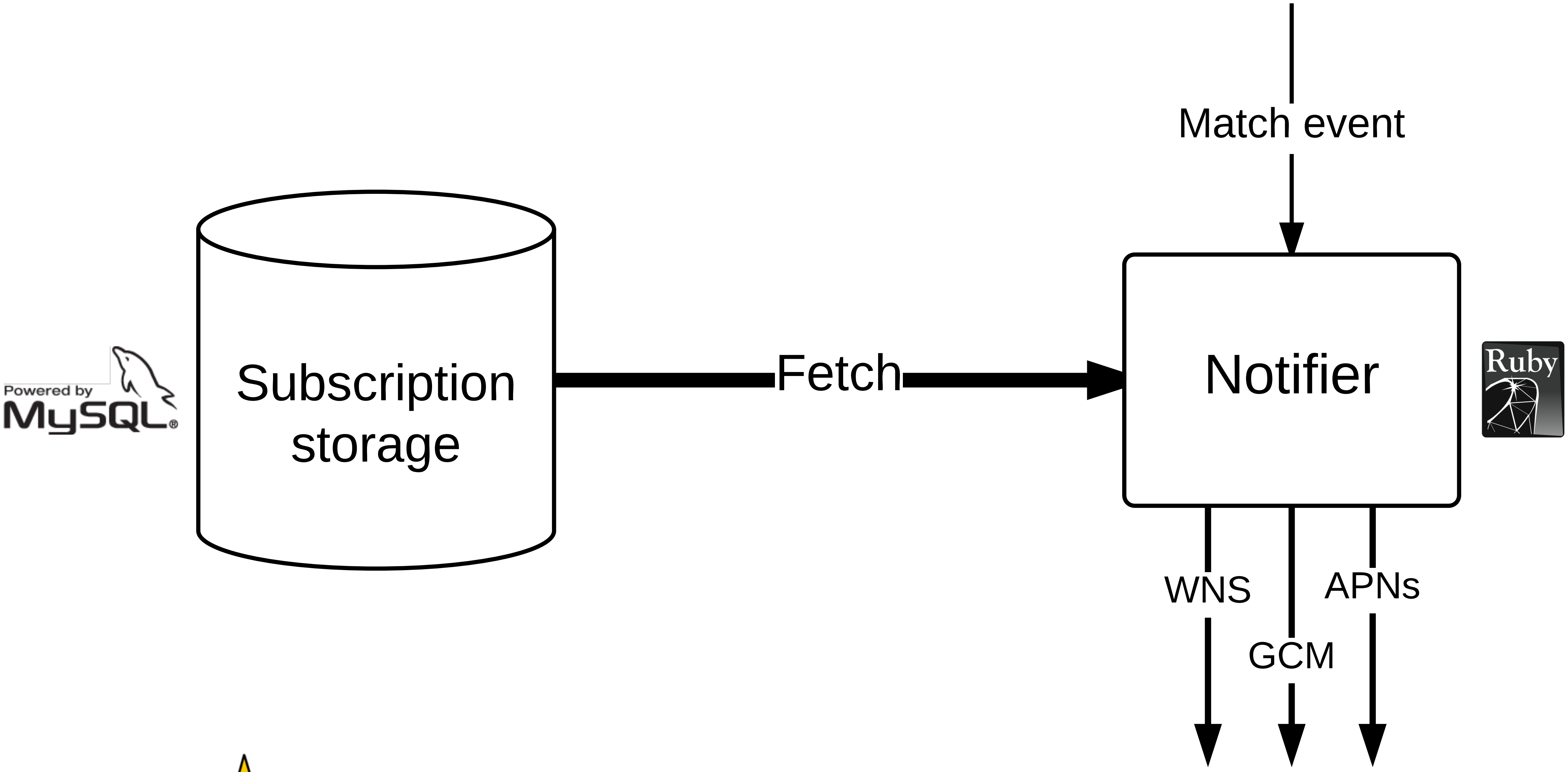
device_token	xfL3k6QxHagF...Nf8Y01a
subject	tournament 15
topic	goal
language	en
country	SE

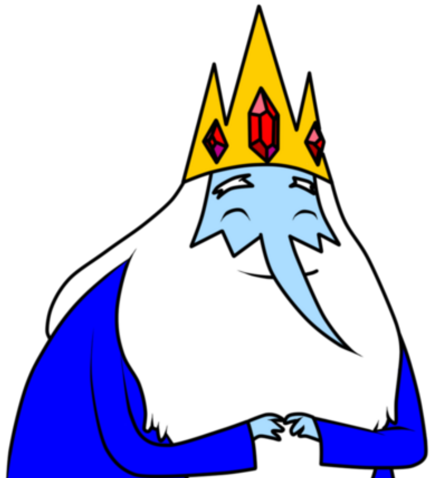
Overlaps do happen

Notification sending steps

1. Receive match event
2. **Fetch all relevant subscriptions**
3. Filter out token duplicates
4. Build translated messages
5. Perform dispatching

Slow push notifications are slow



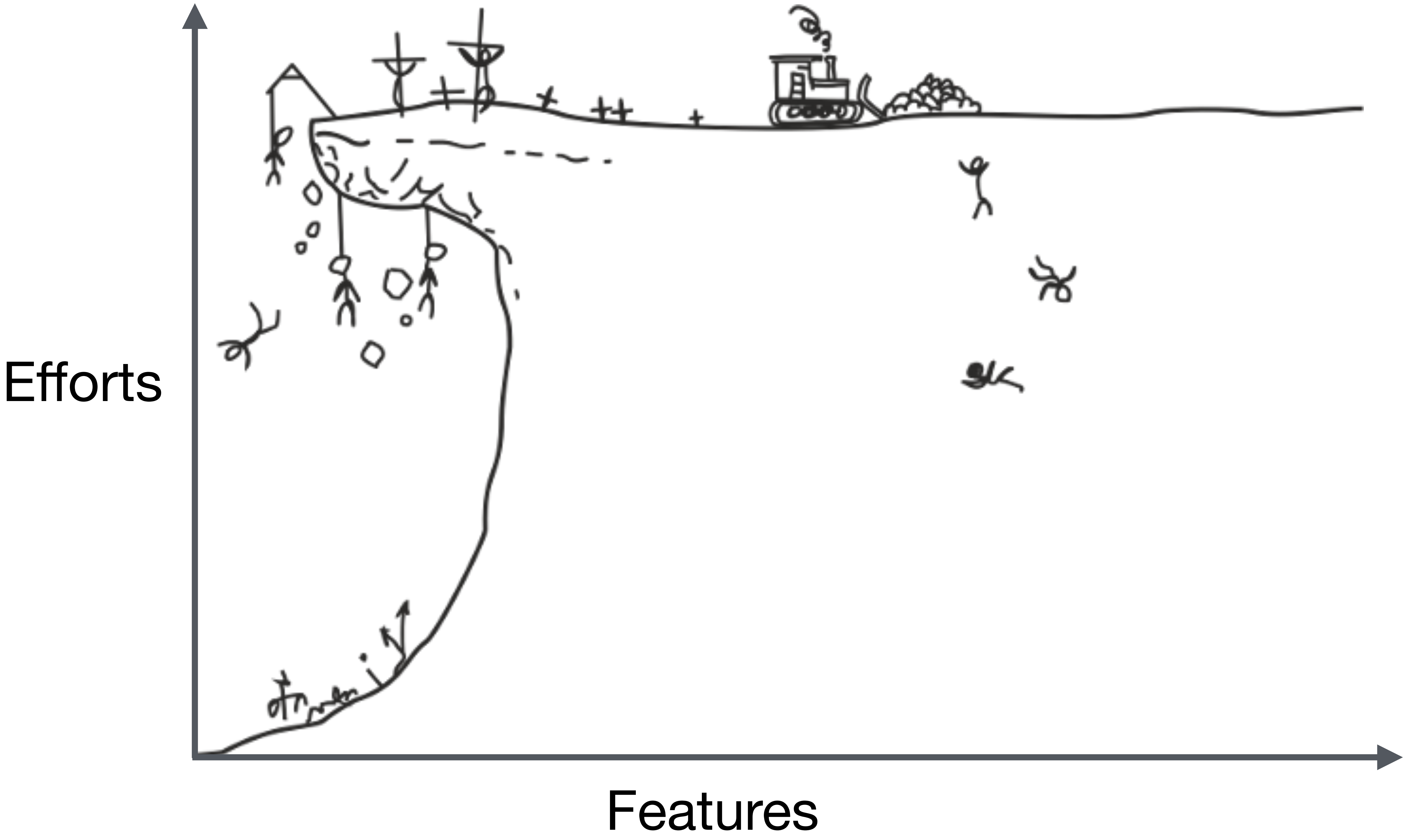


Minutes to fetch 1 million subscriptions.

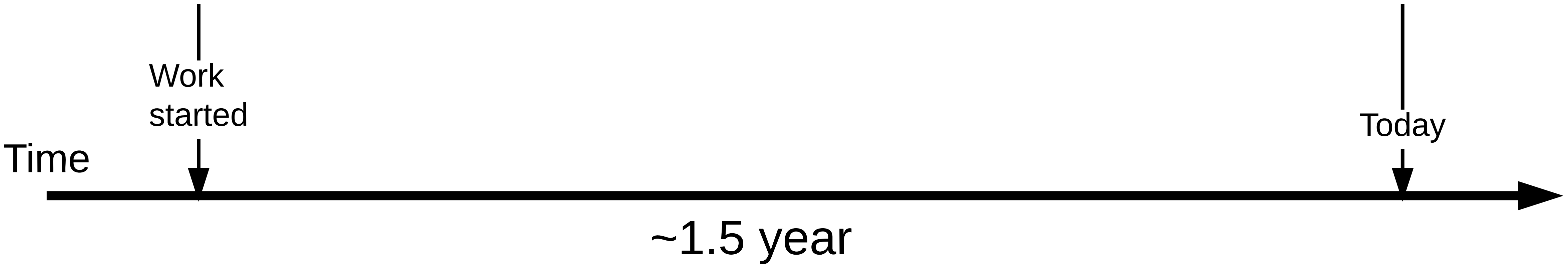
Notification sending issues

- Blocking subscriptions fetching
- Normalized data (many JOIN clauses)
- Everything is sequential
- Tightly coupled components

Keep it calm and start from scratch



How long did it take?



Time

Work started

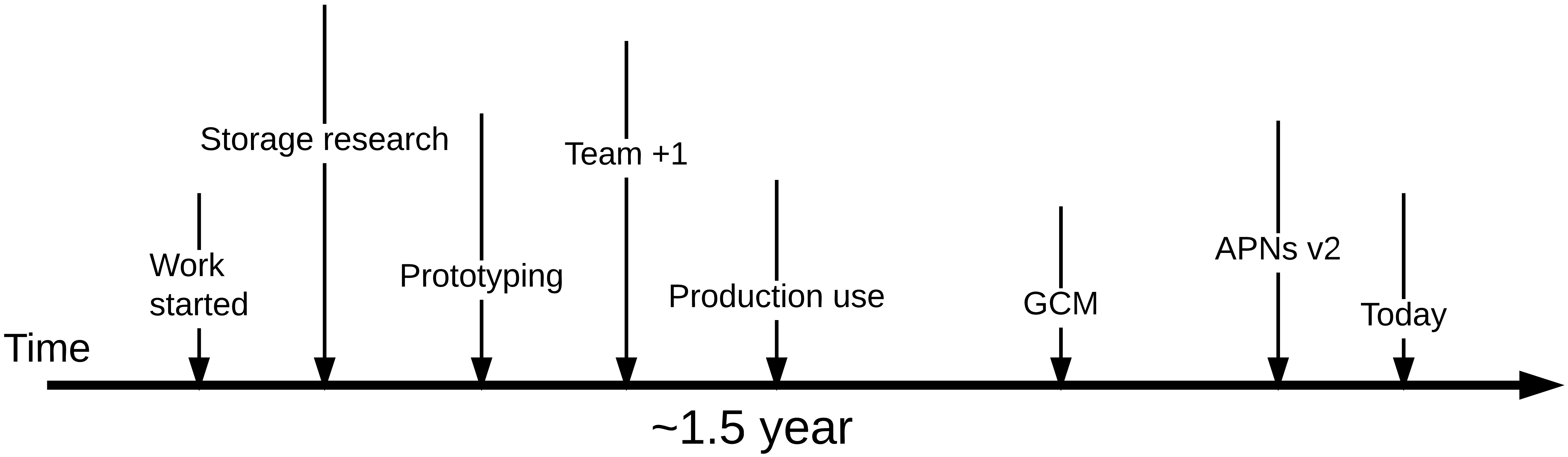
Today

~1.5 year

Really that long?—Yes!

- 2-person team
- Storage research and system prototyping
- Infrastructure for metrics collecting from scratch
- Migration to Amazon Web Services

Really that long? — Yes!



Storage research and system prototyping

- 200 million subscriptions sample data set
- 4 million subscriptions fetching
- At least 2 storage drivers to evaluate

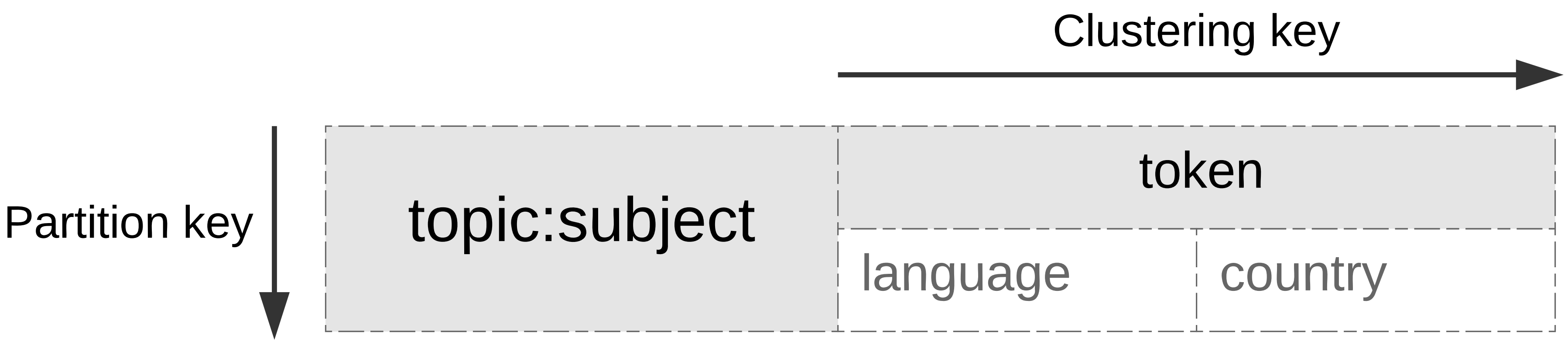
> "If you don't make experiments before starting a project, then your whole project will be an experiment."
>
> –Mike Williams

Cassandra is the answer

- Customizable replication
- Tunable eventual consistency
- Result streaming
- Data compression
- Immutable sequentially-written data
- Last-write-wins conflict resolution

Immutable sequentially-written data

<table>
<tr><td rowspan="2">goal:team:42</td><td colspan="2">xfL3k6QxHagF...Nf8Y01a</td><td rowspan="2">...</td></tr>
<tr><td>en</td><td>SE</td></tr>
<tr><td rowspan="2">goal:tournament:15</td><td colspan="2">xfL3k6QxHagF...Nf8Y01a</td><td rowspan="2">...</td></tr>
<tr><td>en</td><td>SE</td></tr>
<tr><td colspan="4">...</td></tr>
</table>



Clustering key

Partition key

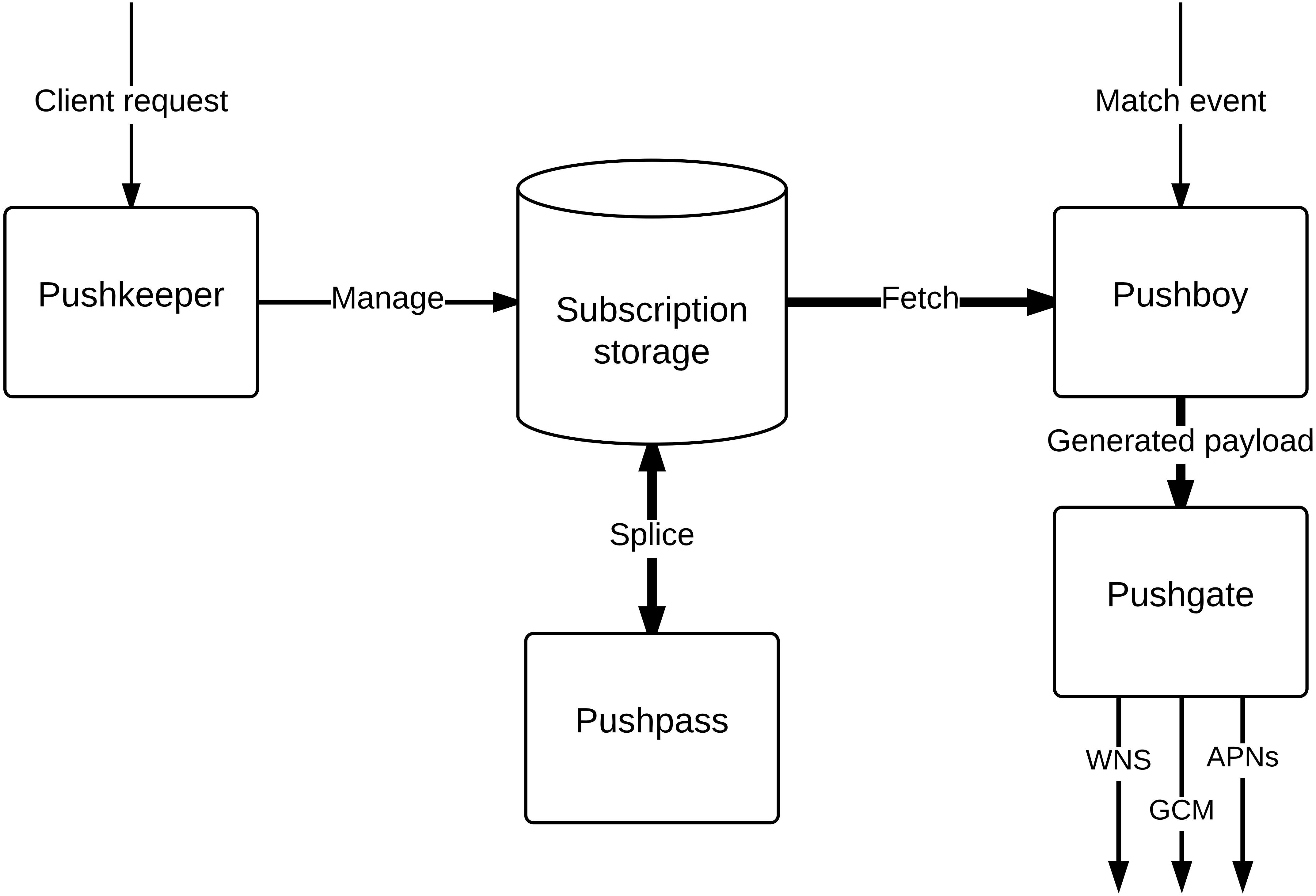
topic:subject

token

language

country

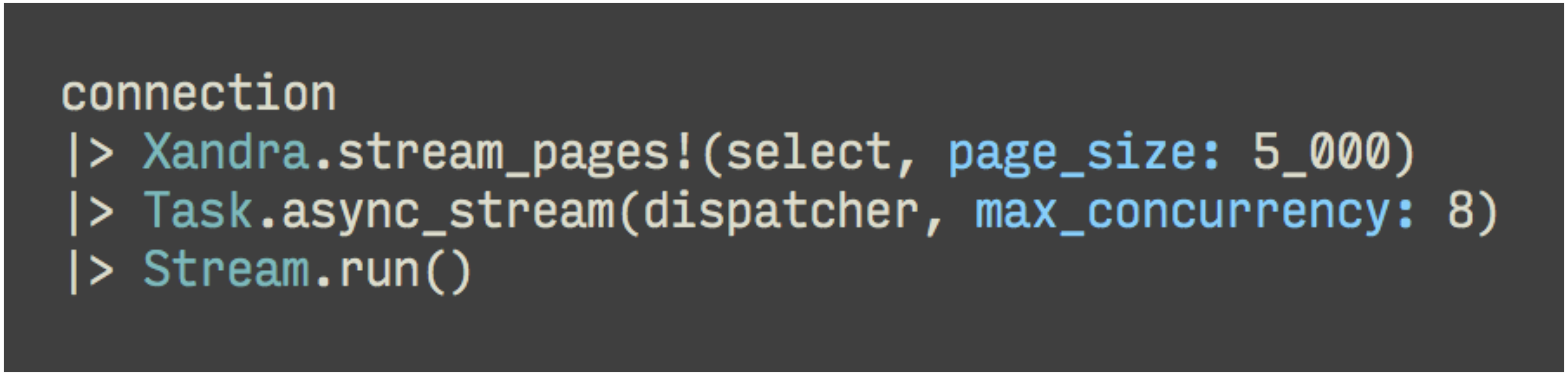
Long live the new system!



Long live the new system!

- The Erlang runtime system and Elixir
- Subscription streaming
- Concurrent payload building and dispatching
- Pushgate is reusable
- Less resources are required than previously
- Components scale independently

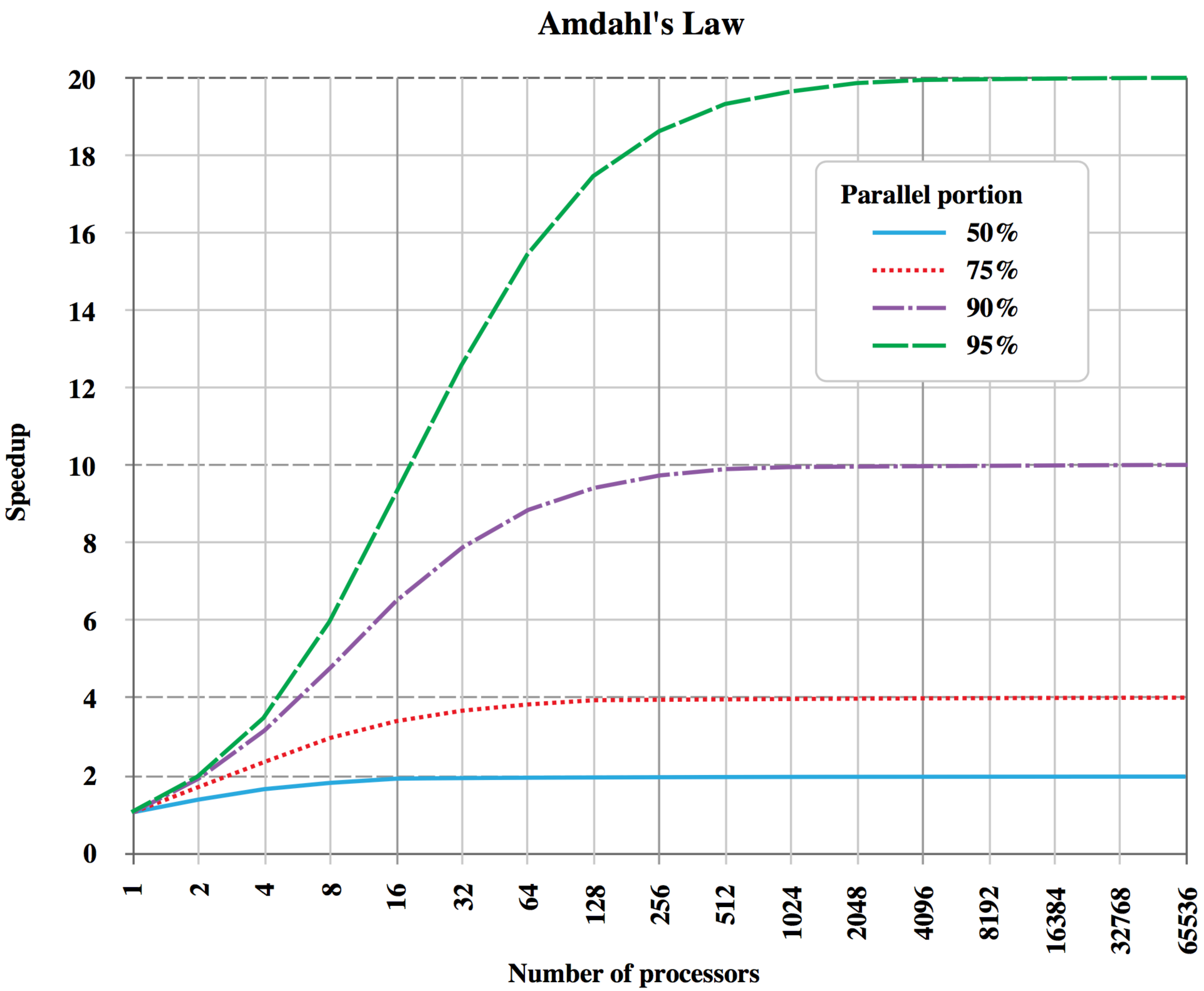
Long live the new system!



```
connection
|> Xandra.stream_pages!(select, page_size: 5_000)
|> Task.async_stream(dispatcher, max_concurrency: 8)
|> Stream.run()
```

lexhide/xandra

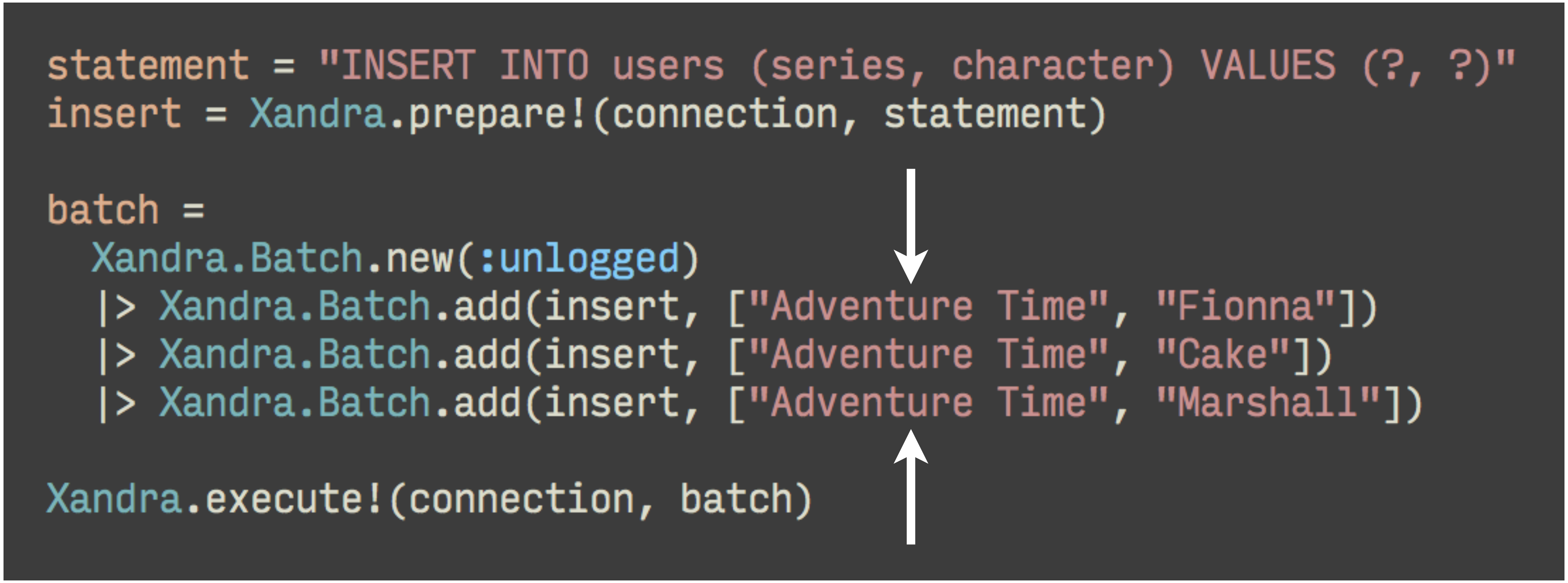
Beyond the speed of light



Beyond the speed of light

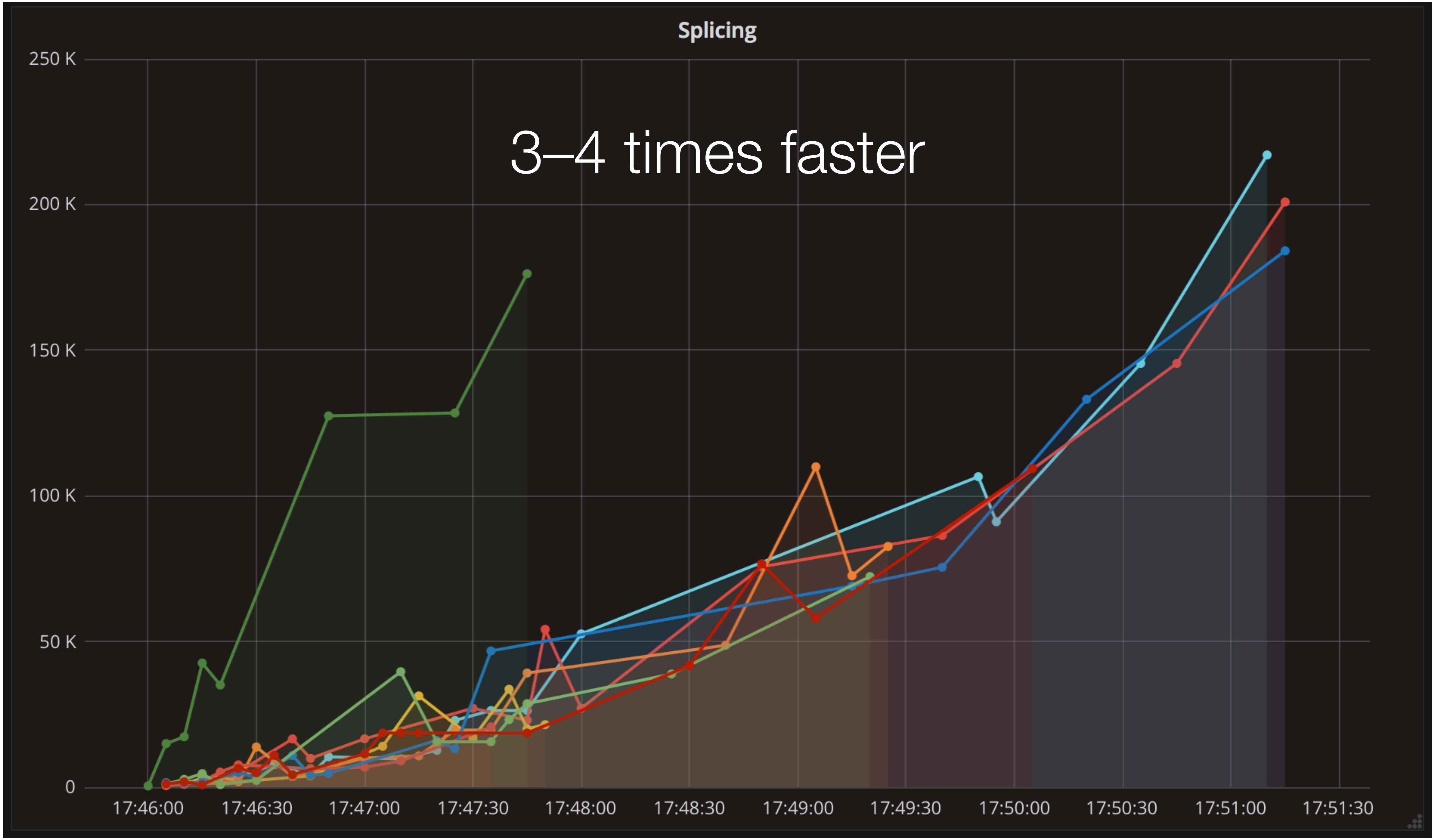
- Avoid data copying
- Favor batch operations
- Decode binary data with single match context
- Encode data to iodata
- Watch out your run queue

Single partition batch operation



```
statement = "INSERT INTO users (series, character) VALUES (?, ?)"
insert = Xandra.prepare!(connection, statement)

batch =
  Xandra.Batch.new(:unlogged)
  |> Xandra.Batch.add(insert, ["Adventure Time", "Fionna"])
  |> Xandra.Batch.add(insert, ["Adventure Time", "Cake"])
  |> Xandra.Batch.add(insert, ["Adventure Time", "Marshall"])

Xandra.execute!(connection, batch)
```

Single partition batch operation



3–4 times faster

Single binary match context

- Efficiency Guide in the Erlang documentation
- PR #5859 in the Elixir repository
- PR #30 in the Msgpax repository
- PR #85 in the Xandra repository

lexmag/msgpax

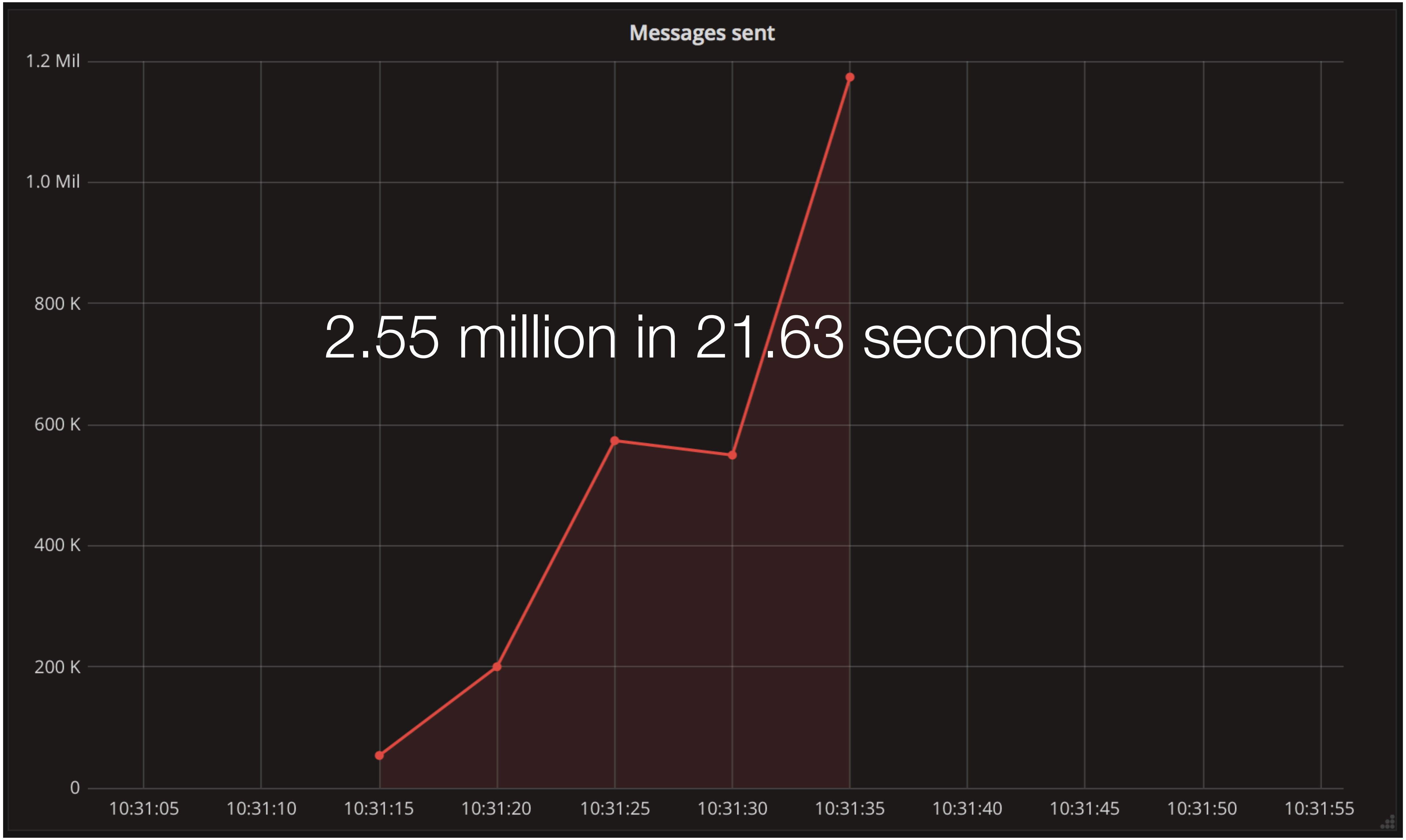
Results time

1 million subscribers

Before:
45–70 seconds
blocking

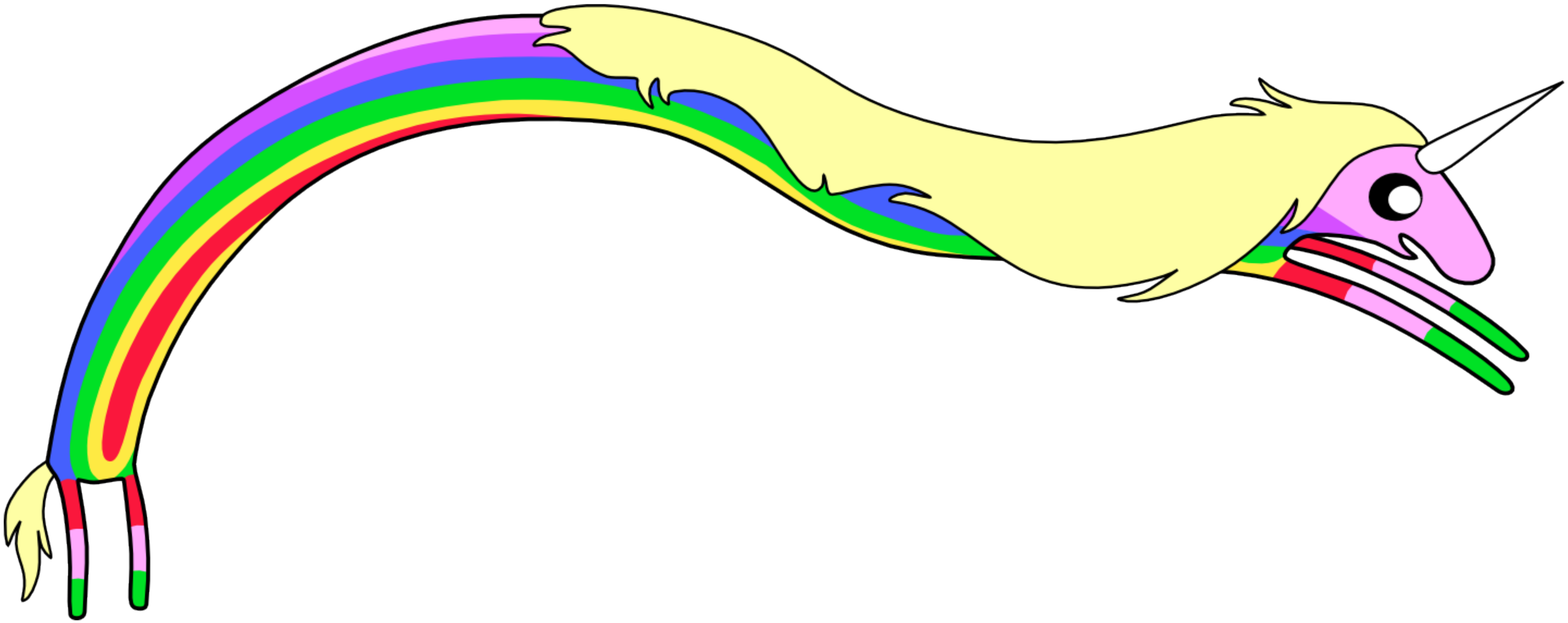
After:
10–15 seconds
streaming

Results time



2.55 million in 21.63 seconds

Absolutely correct systems are like unicorns



Absolutely correct systems are like unicorns

- At-least-once delivery
  - ✓ Idempotent handling or deduplication

Absolutely correct systems are like unicorns

- At-least-once delivery
  - ✓ Idempotent handling or deduplication
- Client request reordering
  - ✓ Monotonic time
  - ✓ Cassandra treats deletes as upserts
  - ✓ Last write wins

ferd.ca/tout-est-terrible.html

Thank you!