

ElixirConf.EU

2017



elixir

@elixirlang / elixir-lang.org



Day 1 Keynote

José Valim

- Elixir 5yrs old, 1.0 → 1.4
- _Seven Languages in Seven Weeks_
- "Elixir" before Elixir
 - defobject >, <
- Elixir Goals → Rethink meta-programming
- Elixir \simeq Data(types) + Modules + Processes
 - http://theerlangelist.com//article/spawn_or_not
 - Time → Runtime
- Elixir 1.5

CQRS/ES + Elixir

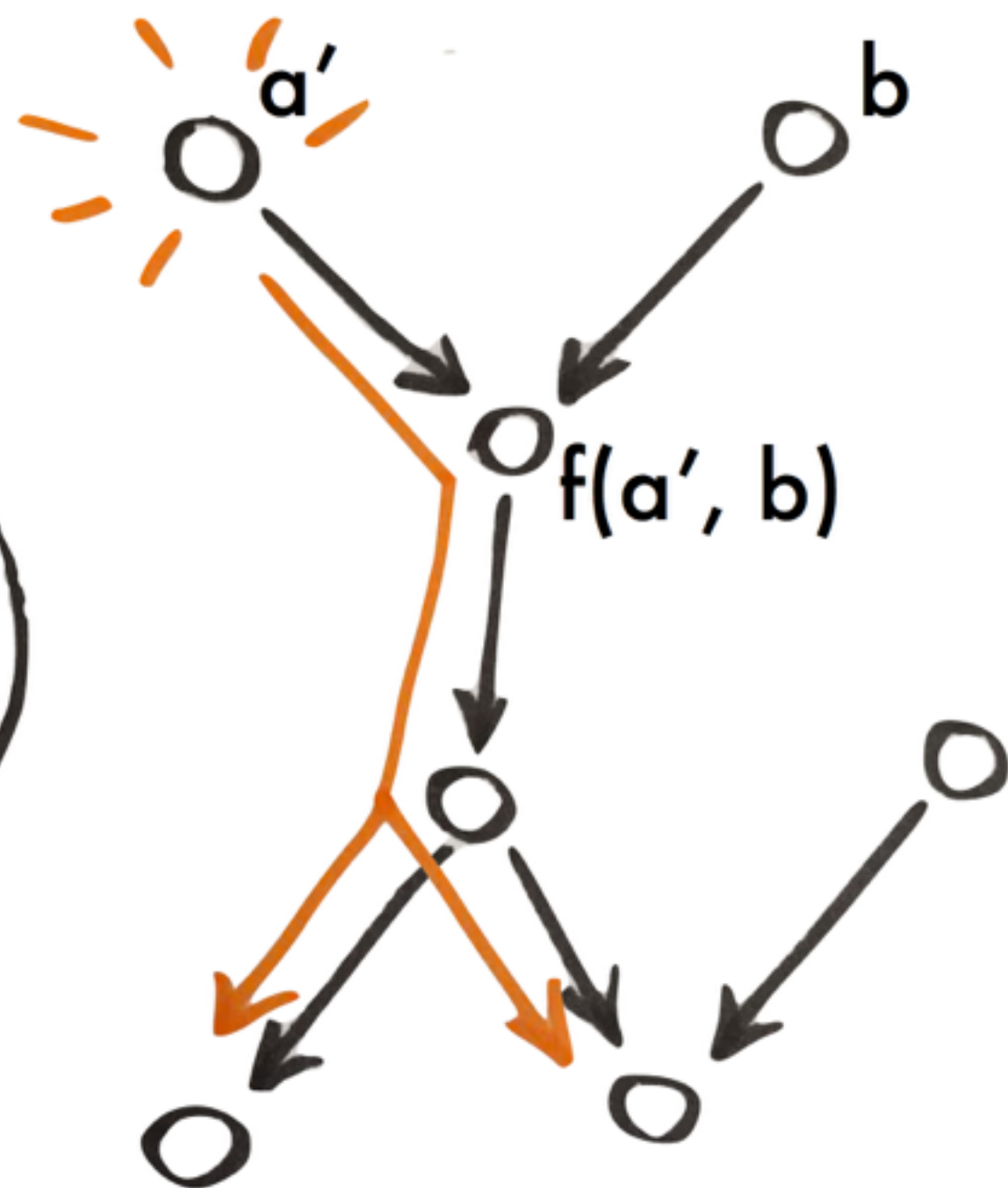
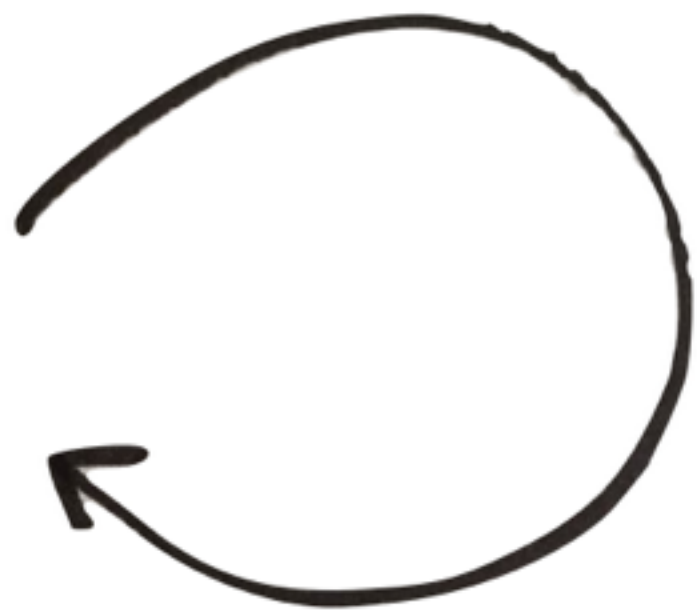
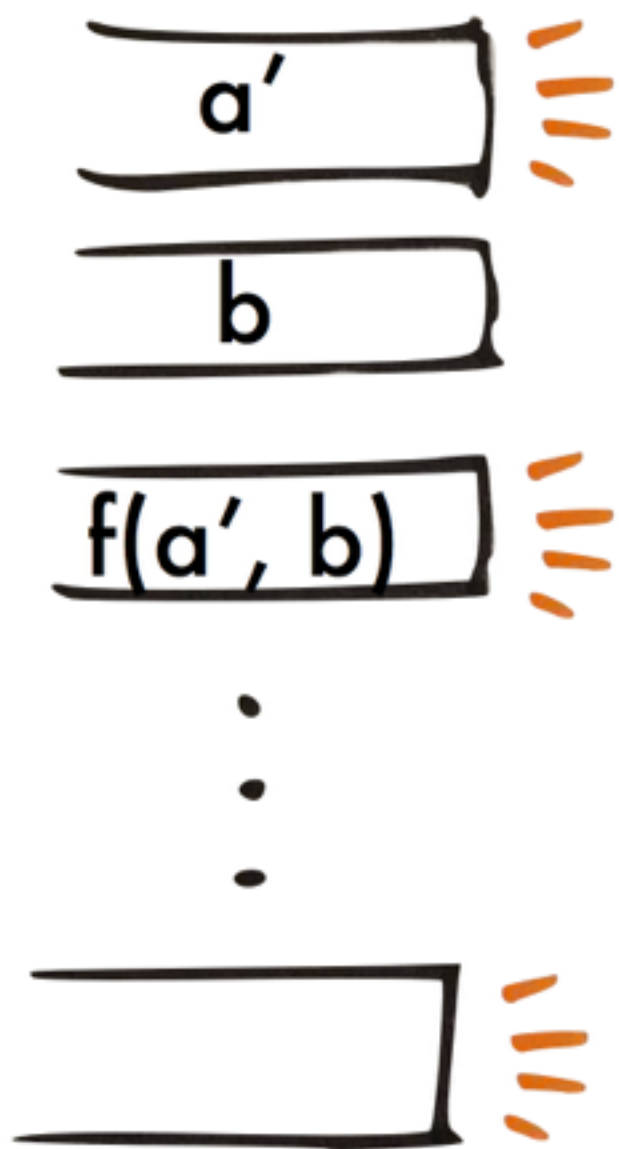
Tonći Galić

- Event Sourcing
 - eventual consistency
- CQRS
 - Command
 - Query
 - Responsibility Segregation
- +Elixir
 - `gen_server/gen_statem/gen_stage`
 - pattern matching
 - etc...
- <https://10consulting.com/2017/01/04/building-a-cqrs-web-application-in-elixir-using-phoenix/>

Nextjournal

Andrea Amantini

- <https://nextjournal.com/>
 - editor + interactive, connected code cells
- Dependencies of code cells → Graph of processes
 - Cuneiform (cuneiform-lang.org)
 - GenStage (:digraph, BroadcastDispatcher)
- Applications
- Phoenix Channel
- Runner App
 - Erlang Ports
 - Docker



Visualization

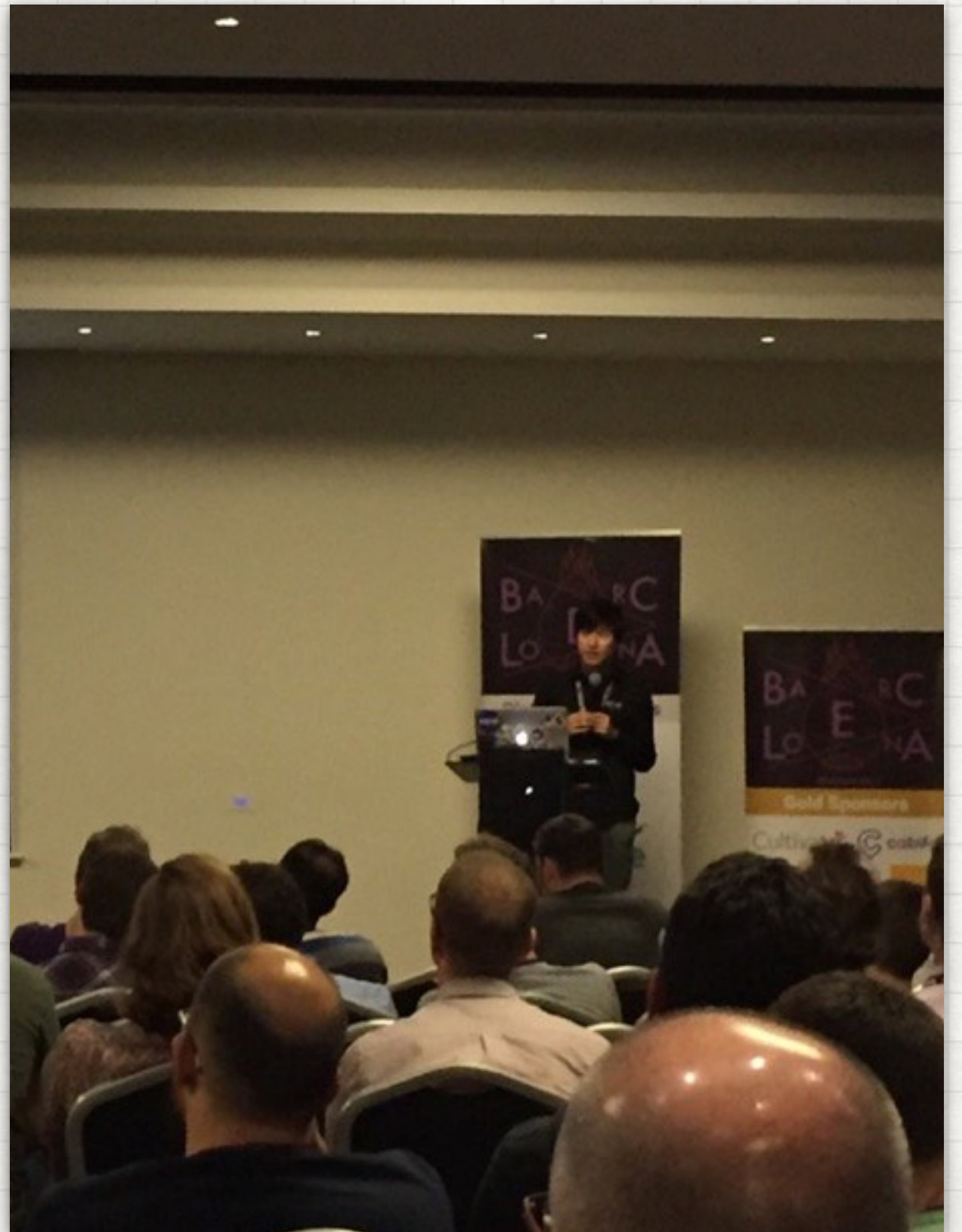
Dániel Vámosi

- <https://github.com/koudelka/visualixir>
- <http://www.erlang.pl/>
- “Paparazzi”



LIGHTNING TALK

Introducing ExLoglite

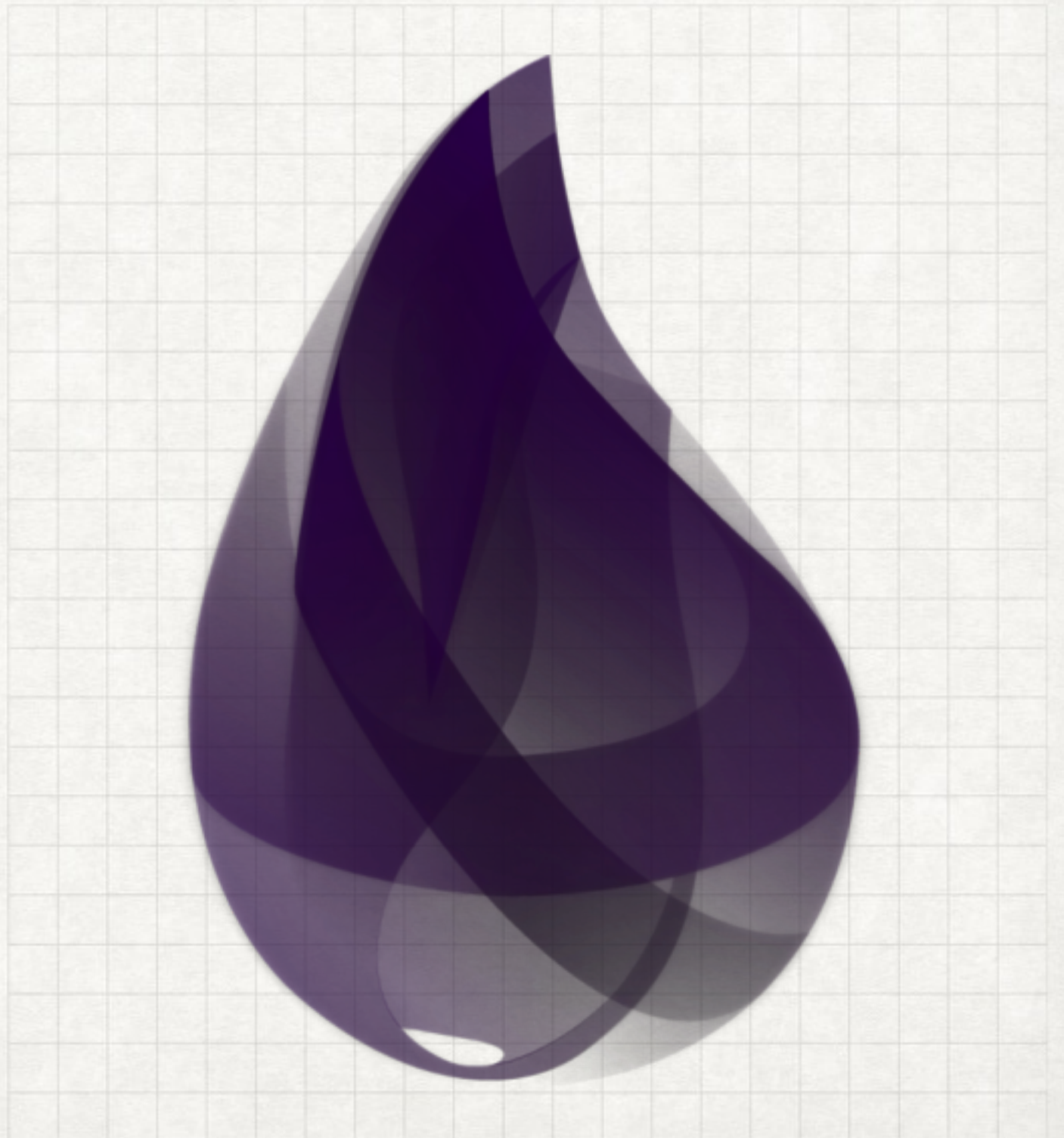




Day 2 Keynote

Tetiana Dushenkivska & Keith Salisbury

- `_The Age of Elixir_`
- `Community`
- `Elixir Shanghai !!`



QuickCheck

Thomas Arts

- Improved Elixir integration (eqc_ex)
 - PBT x ex_loglite
- Shrinking
- Quviq customers: Volvo, Dropbox, Basho...
- Erlang → Elixir for B2B
- Spec! (JSON Schema)
 - generator, validator
- Real-time system challenge (timestamp...)
- JSON Schema + QuickCheck for random data generation

```
defmodule ExLogLite.LogModelEQC do
  use ExUnit.Case
  use EQC.ExUnit

  alias ExLogLite.LogModel

  property "`build` always return string of a specific
length" do
    forall {s, n} ← {utf8(), choose(1, 1000)} do
      ensure byte_size(LogModel.build(s, n)) == n
    end
  end
end
end
```


AquarHEAD L. (ruby-2.4.1) ~/Projects/elixir_projects/ex_log

```
± MIX_ENV=test mix eqc.install --mini
* fetching http://quviq.com/downloads/eqcmini.zip
* stored 52 files in /Users/aquarhead/.mix/archives/eqcmini
* installing archive eqc-2.01.0
* deleted downloaded /Users/aquarhead/.mix/archives/eqcmini
```

AquarHEAD L. (ruby-2.4.1) ~/Projects/elixir_projects/ex_log

```
± MIX_ENV=test mix eqc --numtests 2000
```

.....11.....

.....

OK, passed 2000 tests

.....

Finished in 0.3 seconds

1 property, 4 tests, 0 failures

Randomized with seed 512077

GraphQL

Bruce Williams & Benjamin Wilson



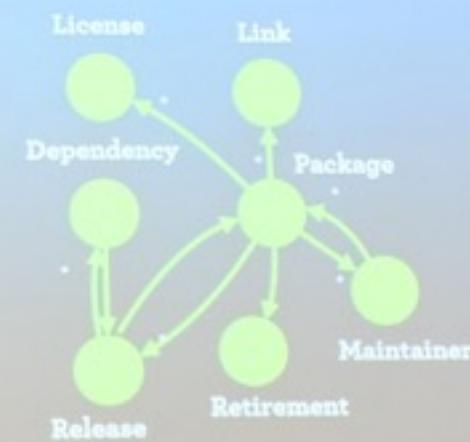
GraphQL

Bruce Williams & Benjamin Wilson

- 3yrs of Elixir @ CargoSense
- RESTful APIs → GraphQL
 - Absinthe
 - and friends...
- Schema → Objects → Fields (type, another obj)
- GraphQL Document
 - Query (like an object)
 - or Mutation, results are like queries
 - or Subscriptions

Designing a Schema

Explanation here:



```
object :package do
  # -
  field :releases, list_of(:release)
  field :retirement, :retirement
  field :maintainers, list_of(:maintainer)
  field :licenses, list_of(:license)
end

object :release do
  # -
  field :package, non_null(:package)
  field :dependencies, list_of(:dependency)
end

object :dependency do
  # - scalars
  field :resolved_release, :release
end

object :retirement do
  # -
end

object :link do
  # -
  field :package, non_null(:package)
end

# -
```

Barceló Sant

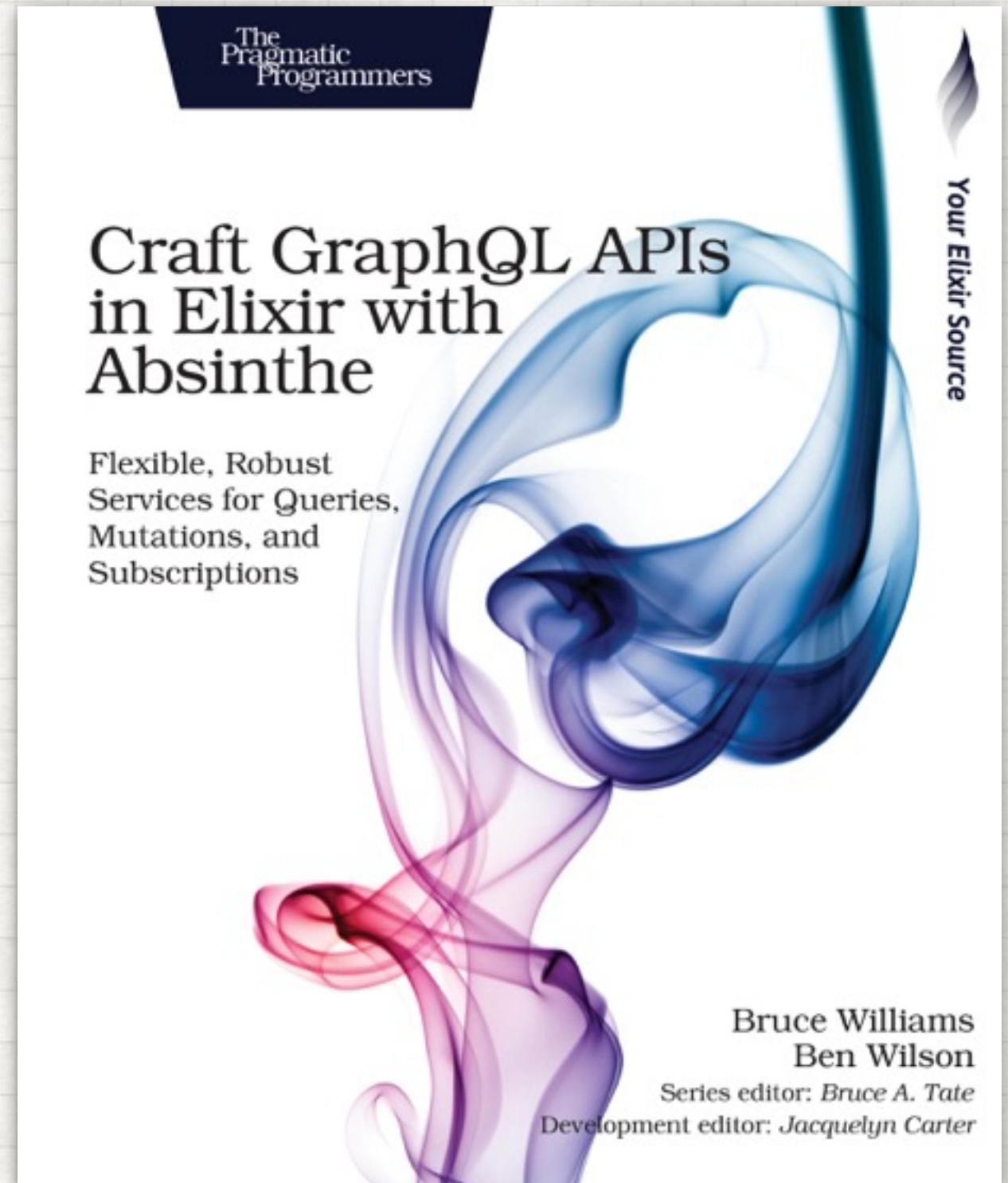
Absinthe

Bruce Williams & Benjamin Wilson

- Resolver → Macro (Map.get)
- Execution / Data loading:
 - Batching
 - absinthe_ecto
 - lazy loading
 - combine sub-tree query
 - Projection (eager loading)
- Extending Absinthe (hexdocs)
 - Phases ▷ Pipeline
 - Limit by complexity, etc...
 - middleware (per-field)

COMING SOON

GraphQL BOOK



Scalable Mobile Game Backend

Petri Kero

- Distillery release
- Docker → Kubernetes
 - libcluster
- Distributed Locks
 - Global PG
 - Sloppy Quorum
- Use AI client for load testing
- 420k concurrents @ 8-node, 36 vCPU cluster
 - scaling up steps...

MOBILE GAME BACKEND

CHALLENGES

- Lots of concurrent users
- Complex interactions between players
- Persistent world with frequent state mutation
- Single unified game world

ELIXIR TO THE RESCUE

- Built-in clustering
- Distributed messaging
- Great for stateful servers

MORE BENEFITS

- Fault tolerant
- Rapid development with small team
- Tooling & documentation
- Learning curve

BOTTLENECKS

- 1k: OS file limit for process
- 3k: Logger overwhelmed
- 4k: Ranch max connection setting
- 10k: DynamoDB provisioned bandwidth
- 20k: Test client spawn rate
- 50k: Disk full (after 12h)

MORE BOTTLENECKS

- 51k: CPU limit on 3-node 8 vCPU system
- 150k: Locker overwhelmed
- 175k: DynamoDB client HTTPS overhead
- 420k: Client spawn rate

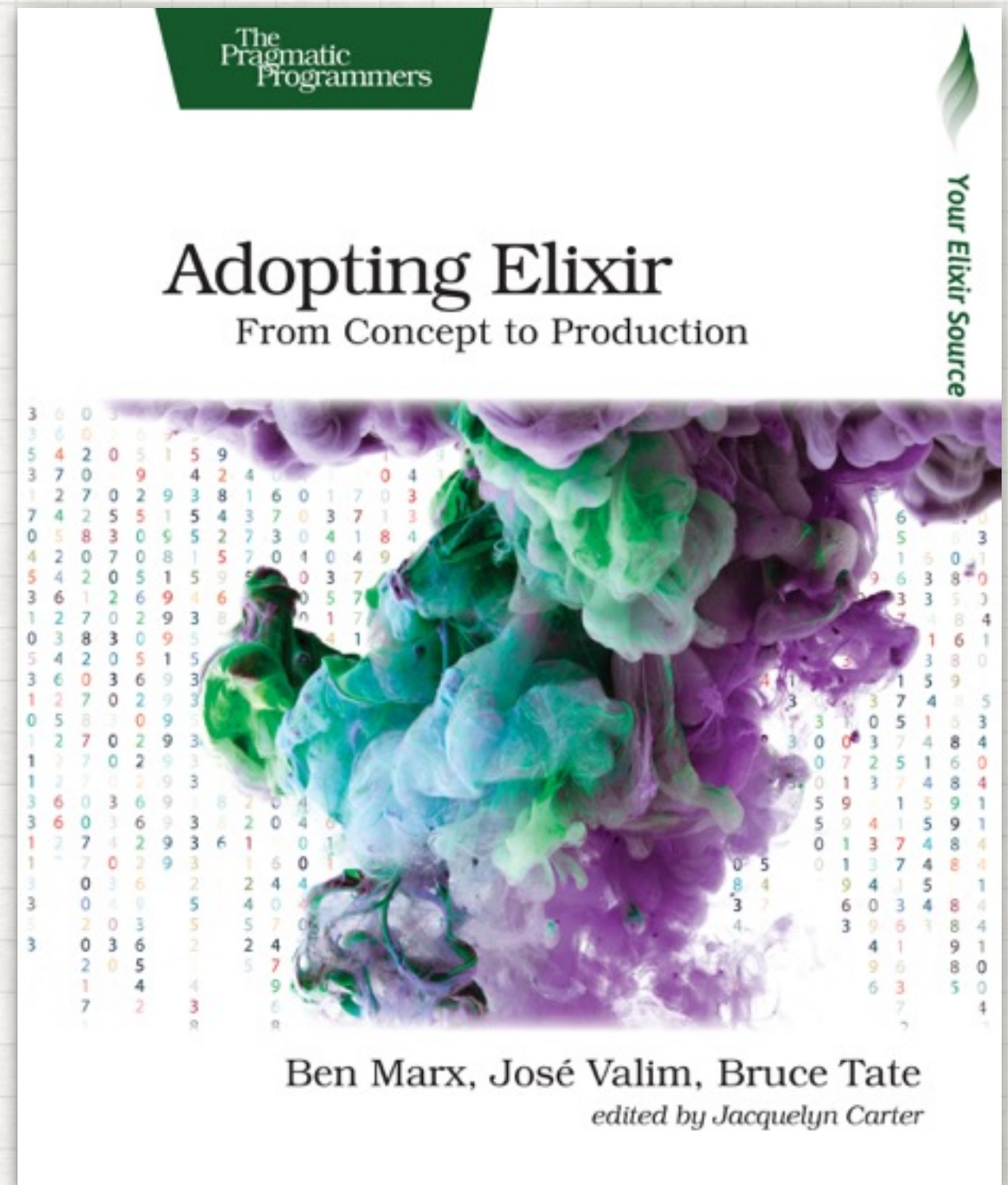
Closing Keynote

Chris McCord

- Why Phoenix 1.3
 - Lonestar ElixirConf 2017- KEYNOTE: Phoenix 1.3
 - still rc >,<
- Phoenix 1.4
 - Metrics
 - Instrumenting

NEW BOOK²

ADOPTING ELIXIR



See U Next Time

@aquarhead

