


IDIOMS FOR DISTRIBUTED APPLICATIONS WITH ELIXIR

@josevalim
@plataformatec

4 Thread Safety

The work done to make Rails thread-safe is rolling out in Rails 2.2. Depending on your infrastructure, this means you can handle more requests with fewer copies of Rails, leading to better server performance and higher utilization of multiple cores.

To enable multithreaded dispatching in production mode of your application, add the following to your `config/environments/production.rb`:



```
config.threadsafe!
```

- More information :
 - [Thread safety for your Rails](#)
 - [Thread safety project announcement](#)
 - [Q/A: What Thread-safe Rails Means](#)

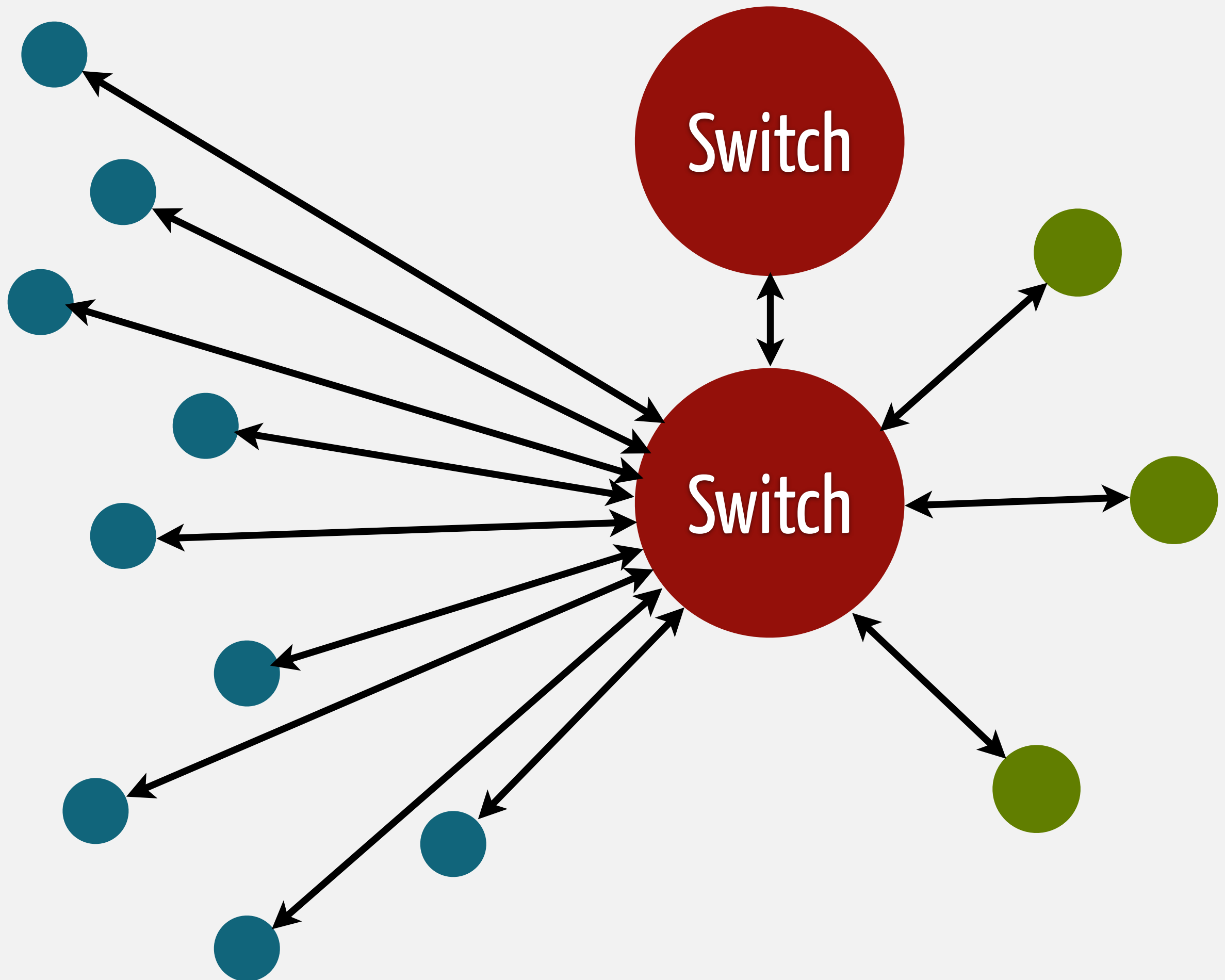
Rails 2.2
threadsafe

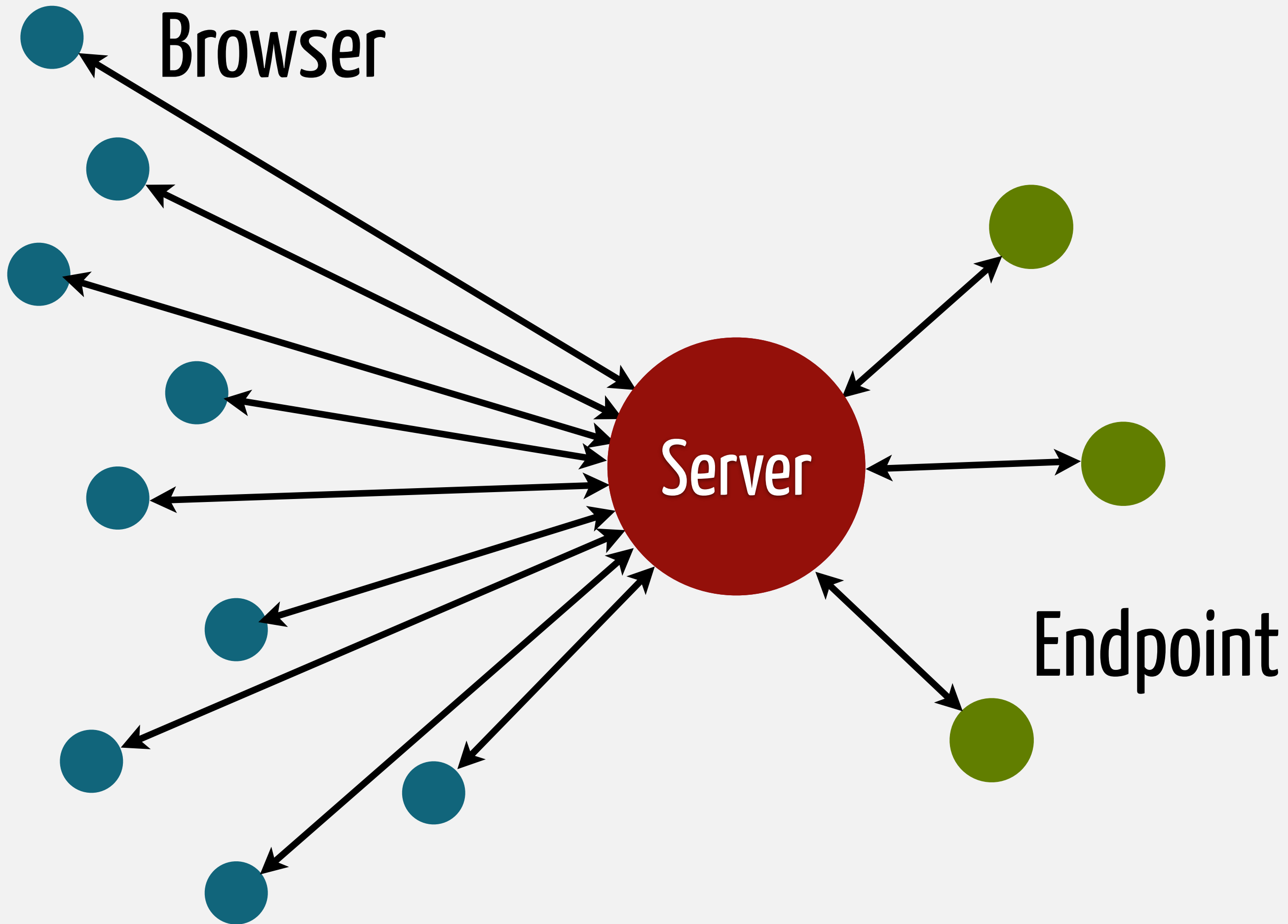
Functional programming

- Explicit instead of implicit state
- Transformation instead of mutation



ERLANG





amazon.com[®]

facebook[®]

ERICSSON 

 **MOTOROLA**

 heroku

 riak

<http://stackoverflow.com/questions/1636455/where-is-erlang-used-and-why>



**2 million connections
on a single node**

[http://blog.whatsapp.com/index.php/
2012/01/1-million-is-so-2011/](http://blog.whatsapp.com/index.php/2012/01/1-million-is-so-2011/)



Intel Xeon CPU X5675 @ 3.07GHz

24 CPU - 96GB

Using 40% of CPU and Memory

Idioms



elixir

- Functional
- Concurrent
- **Distributed**



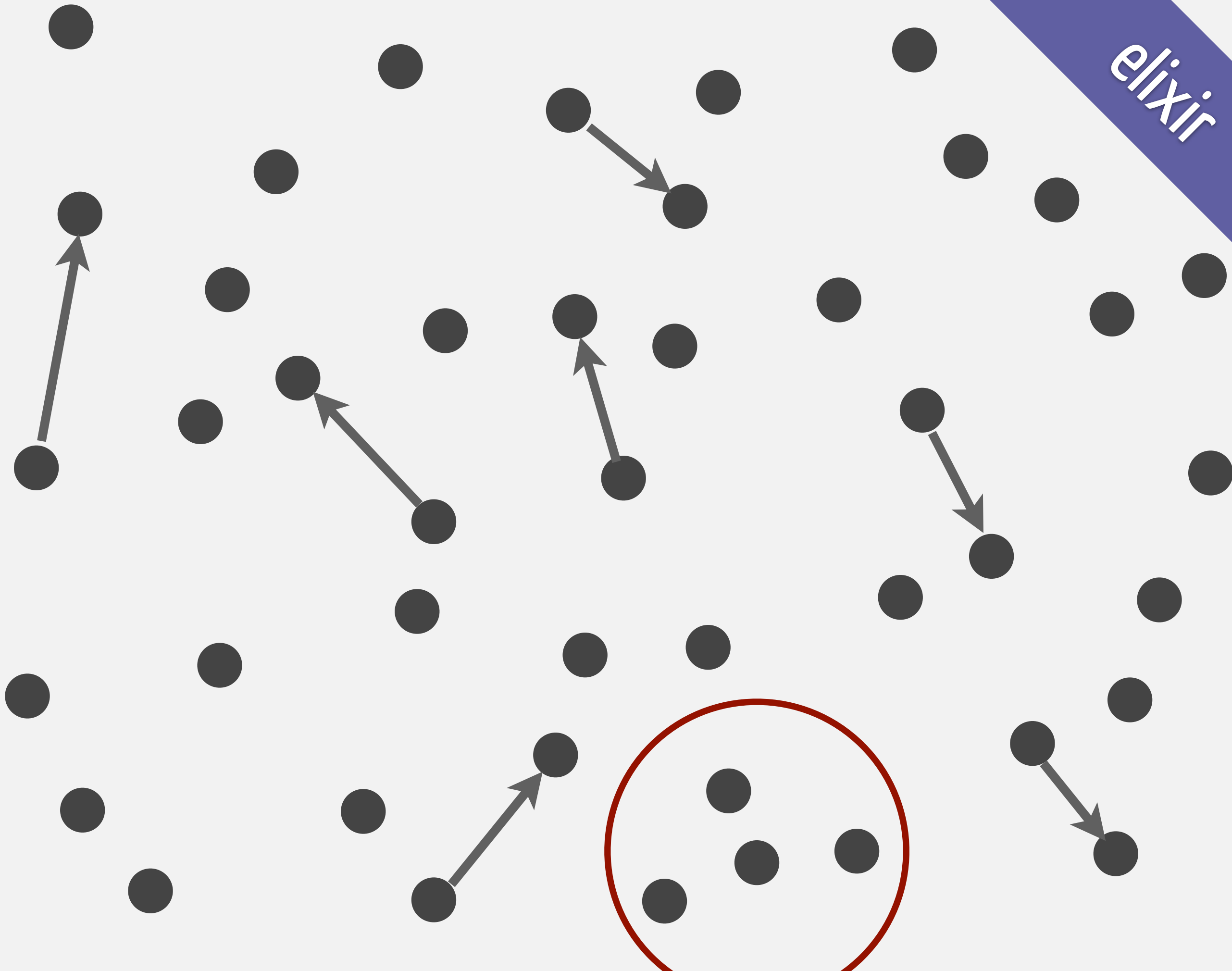
Sequential
code

Sequential
code

elixir



elixir



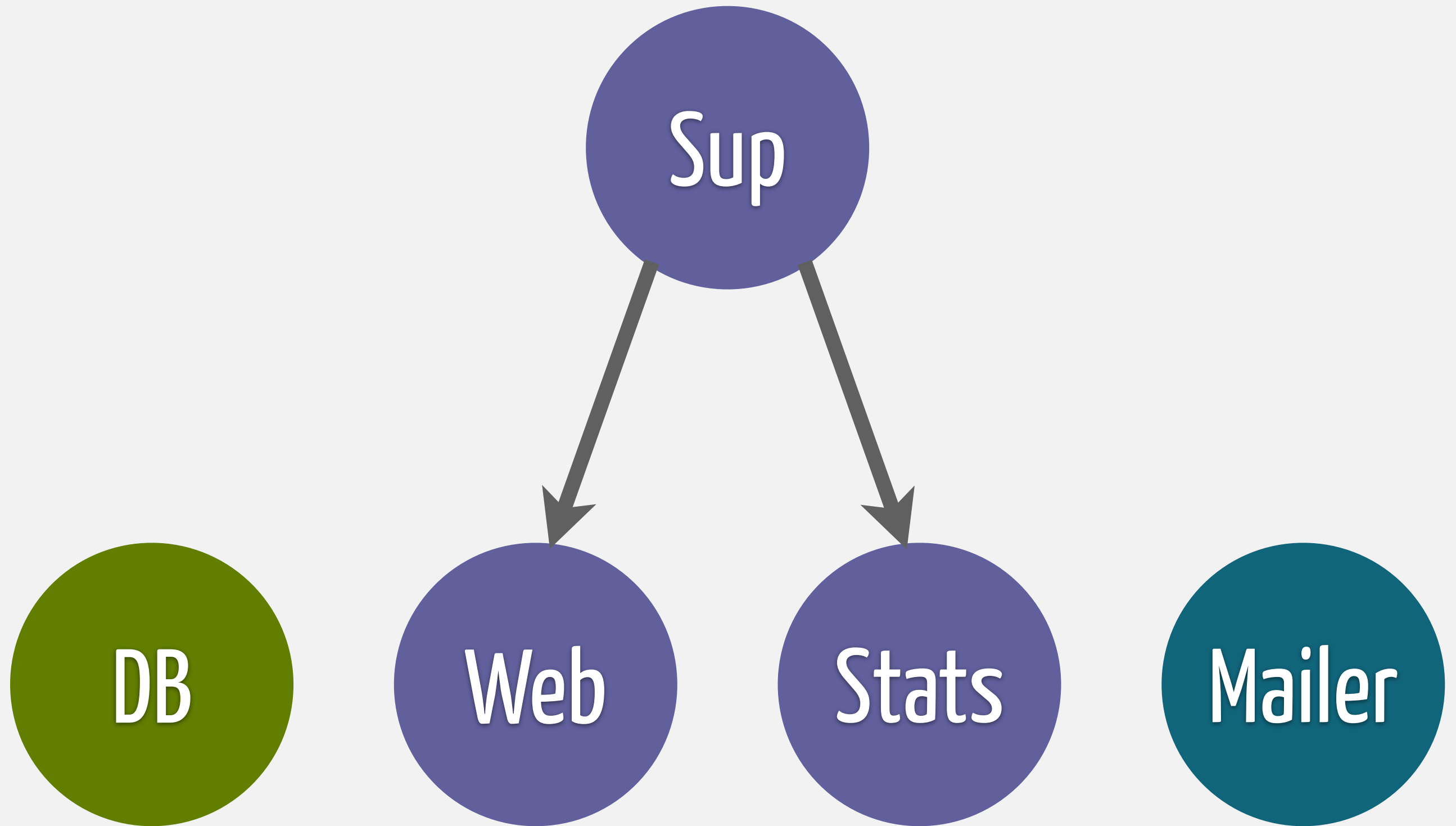


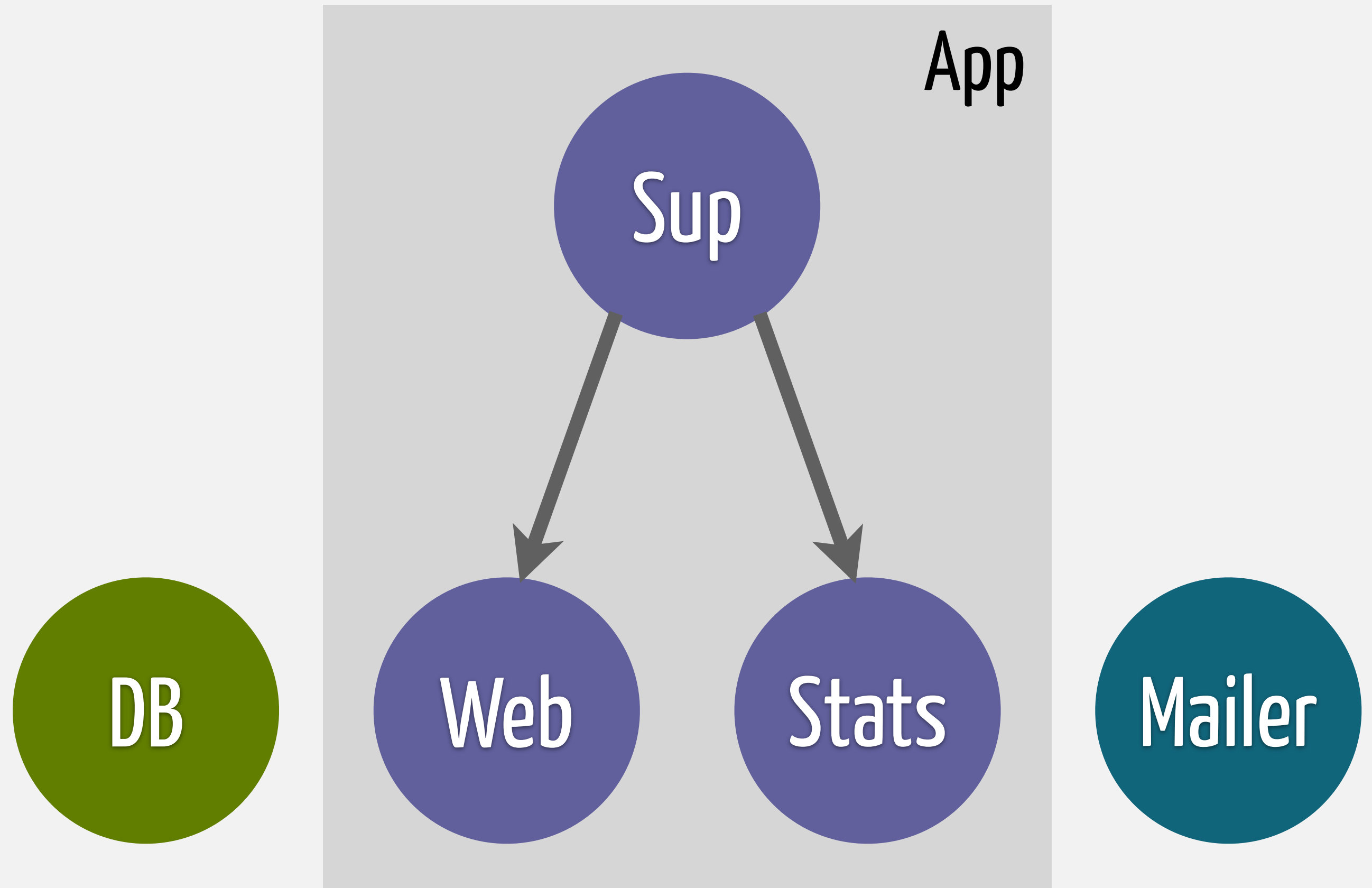
DB

Web

Stats

Mailer

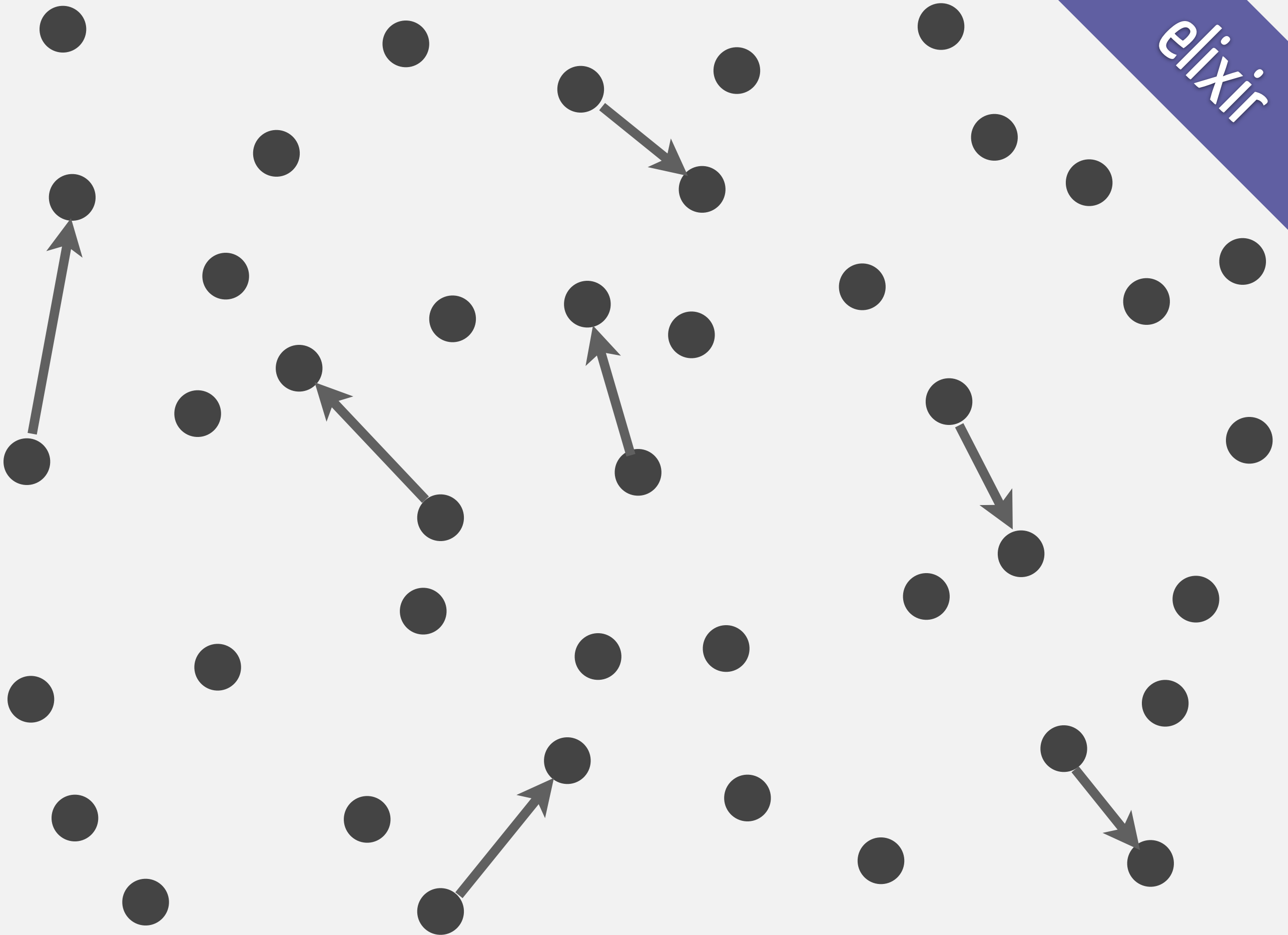




- Processes
- Supervisors
- Applications
- Message passing

- Concurrent
- "Fail fast"
- Fault tolerant
- Distributed?

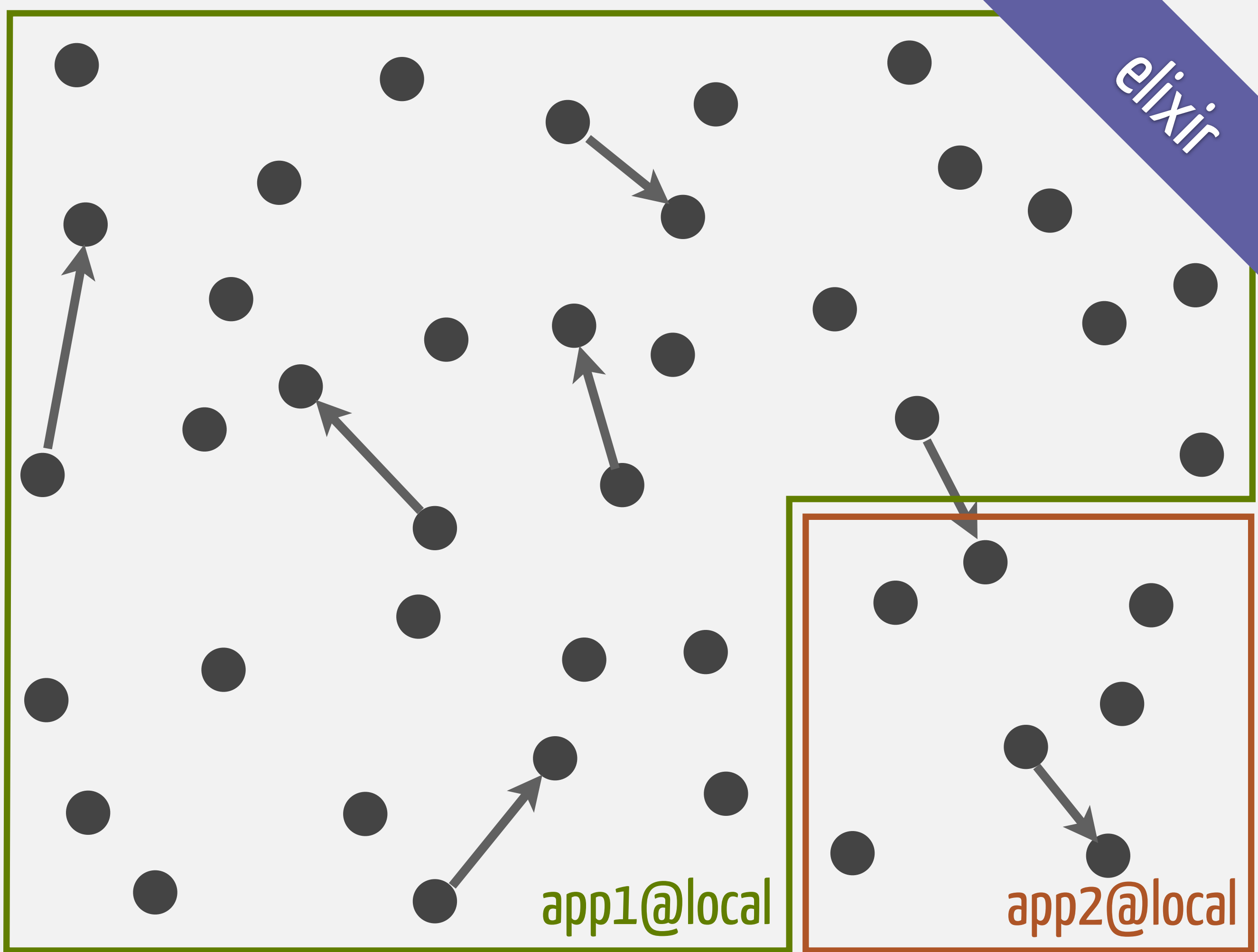
elixir



elixir

app1@local

app2@local





elixir

- Compatibility
- Extensibility
- Productivity

Compatibility

Extensibility

Now we need to go meta. We should now think of a language design as being a pattern for language designs. A tool for making more tools of the same kind.

Guy Steele – “Growing a language”
at ACM 00PSLA 1998

```
defmacro unless(expr, opts) do
  quote do
    if(!unquote(expr), unquote(opts))
  end
end

unless(is_number(x), do: raise("oops"))
```

```
defmodule MathTest do
  use ExUnit.Case

  test "basic operations" do
    assert 1 + 1 == 2
  end
end
```

```
/Volumes/Work/github/elixir[master+]$ make test_eex
```

```
==> eex (exunit)
```

```
.....F.F.....
```

Failures:

1) test evaluates with assigns (EEx.SmartEngineTest)

**** (ExUnit.ExpectationError)**

expected: "1"

to be equal to (==): "2"

at test/eex/smart_engine_test.exs:11

2) test simple chars lists (EEx.TokenizerTest)

**** (ExUnit.ExpectationError)**

expected: [{:text,2,"foo"}]

to be equal to (==): [{:text,1,"foo"}]

at test/eex/tokenizer_test.exs:8

Finished in 0.07 seconds

48 tests, 2 failures

make: *** [test_eex] Error 1

```
/Volumes/Work/github/elixir[master+*]$
```

```
from p in Post,  
where: p.published_at < now and  
       p.author == "José",  
order: p.created_at
```

Goal #3

Productivity

- First-class documentation
- Tooling (ExUnit, IEx, Mix)
- Hex packages

Demo time!



```
defprotocol String.Inspect
  only: [BitString, List,
  defimpl String.Inspect, for:
    def inspect(false), do:
    def inspect(true), do:
    def inspect(nil), do:
    def inspect(""), do:
    def inspect(atom) do
```

Elixir is a functional, meta-programming aware language built on top of the Erlang VM. It is a dynamic language that focuses on tooling to leverage Erlang's abilities to build concurrent, distributed and fault-tolerant applications with hot code upgrades.

To install Elixir or learn more about it, check our [getting started guide](#). We also have [online documentation available](#) and a [Crash Course for Erlang developers](#). Or you can just keep on reading for a few code samples!

News: [Elixir v0.13.0 released](#),
[hex.pm](#) and [ElixirConf](#)
[announced](#)

JOIN THE COMMUNITY

- [#elixir-lang](#) on freenode IRC
- [elixir-talk](#) mailing list (questions)
- [elixir-core](#) mailing list (development)
- [Issues tracker](#)
- [@elixirlang](#) on Twitter

IMPORTANT LINKS

- [Wiki](#) with articles, projects and talks done by the community
- [Crash course](#) for Erlang developers

LEARNING RESOURCES

Language highlights

Everything is an expression

```
defmodule Hello do
  IO.puts "Defining the function world"

  def world do
    IO.puts "Hello World"
  end

  IO.puts "Function world defined"
end
```

elixir-lang.org



Elixir Conf 2014 Schedule

(16 presentations available)

[Sort by Date Posted](#)

Total views: 25,274

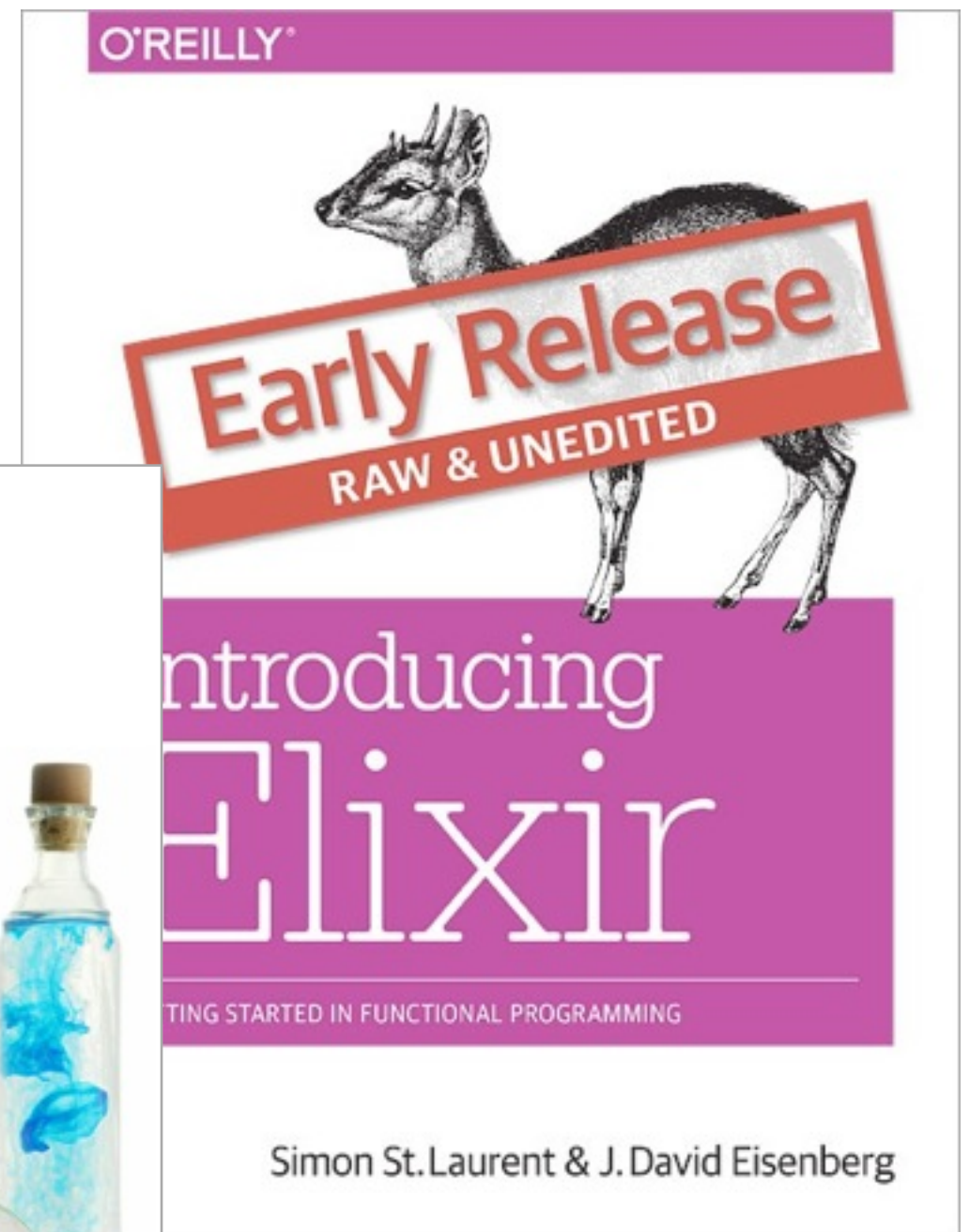
[All Events](#)

38

 Tweet

 8+1

14





plataformatec

tecnologia e engenharia de software



elixir

@elixirlang / elixir-lang.org