ALWAYS AVAILABLE

What happens when resilience is your primary requirement

HELLO!

Claudio Ortolina

I get paid by Erlang Solutions

claudio.ortolina@erlang-solutions.com

@cloud8421


teledrin
SIP
teledrin
SIP

“

Can’t help you, the database is down

-No doctor ever

THINGS YOU MAY HAVE HEARD

“

Elixir is fault-tolerant

-Multiple sources

“

You have to use OTP

-Multiple sources

“

It recovers to a predictable state

-Multiple sources

“

That's awesome! Sign me up

-2014 Claudio

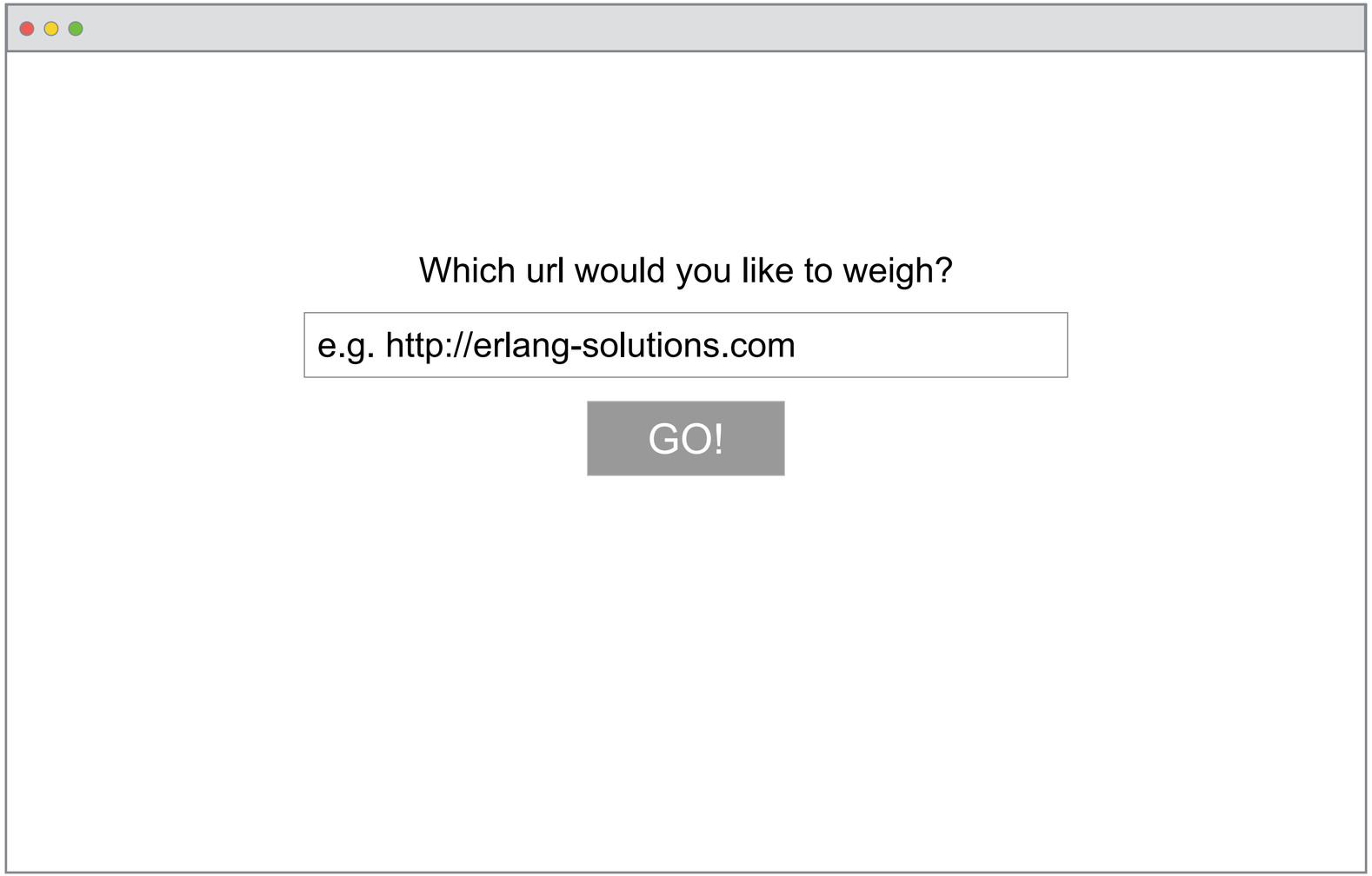
EXCEPT

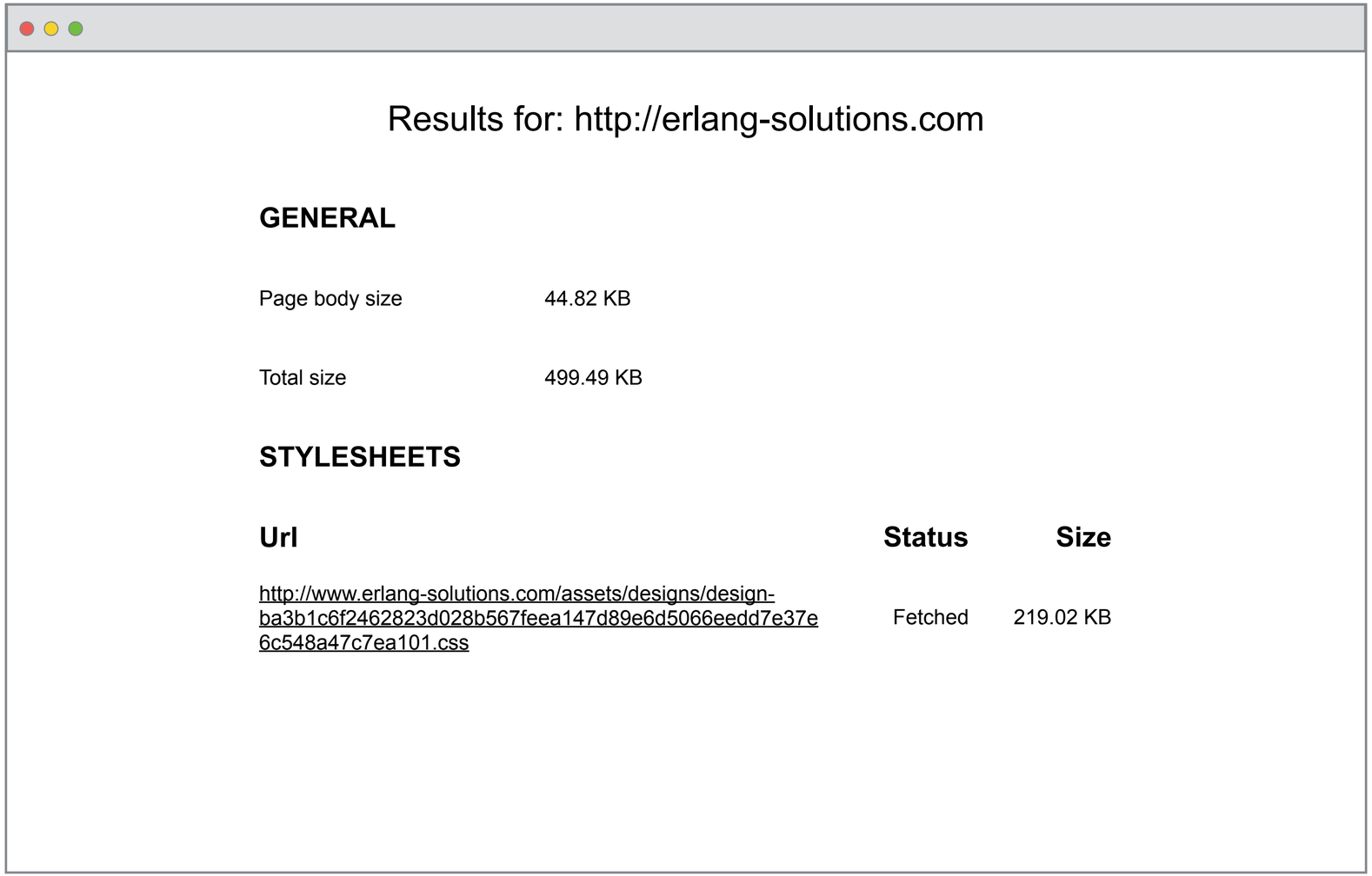
YOU HAVE TO DESIGN FOR AVAILABILITY

It doesn't just happen

SAY HELLO TO LIBRA

Our example application


Which url would you like to weigh?
e.g. http://erlang-solutions.com
GO!



Results for: http://erlang-solutions.com

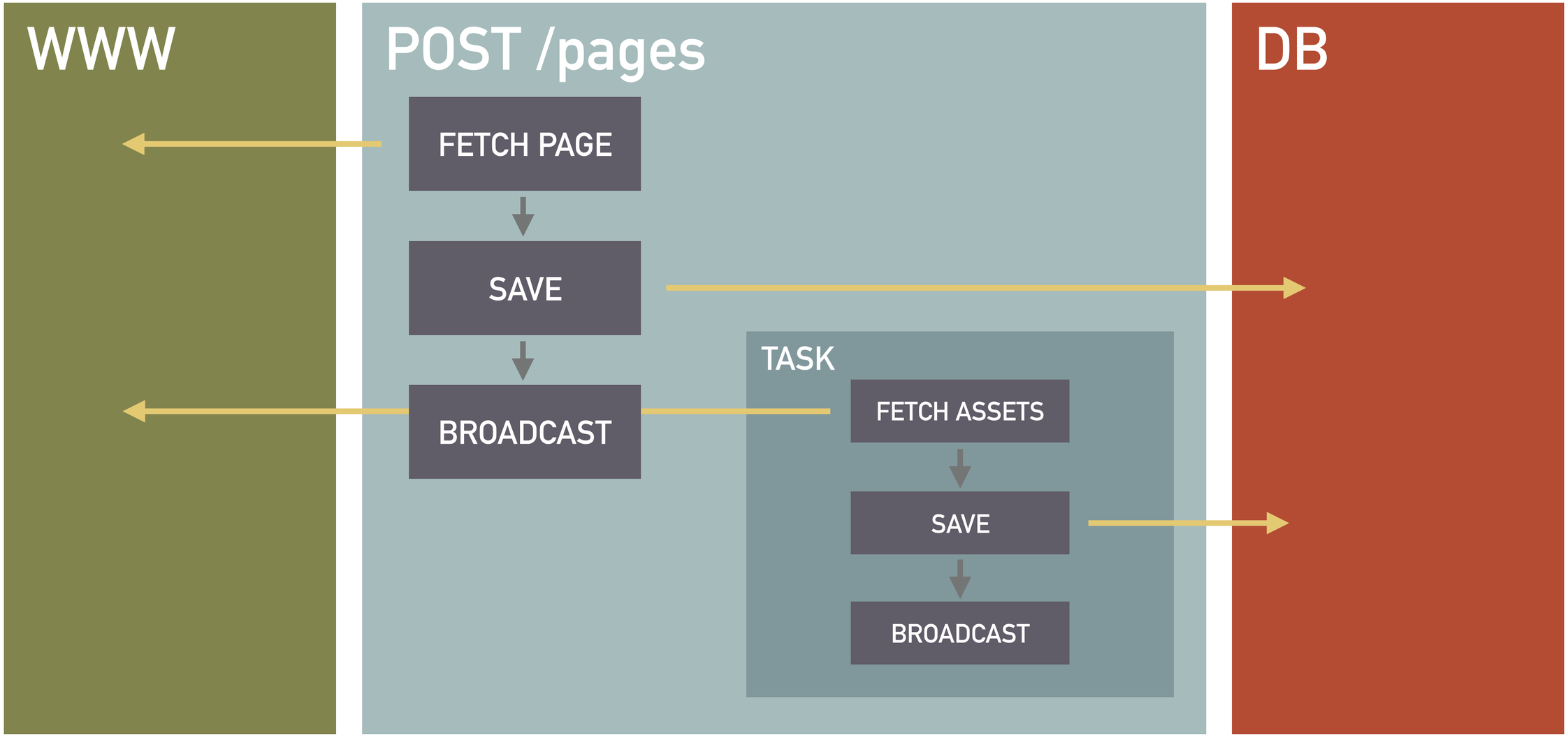
GENERAL

Page body size	44.82 KB
Total size	499.49 KB

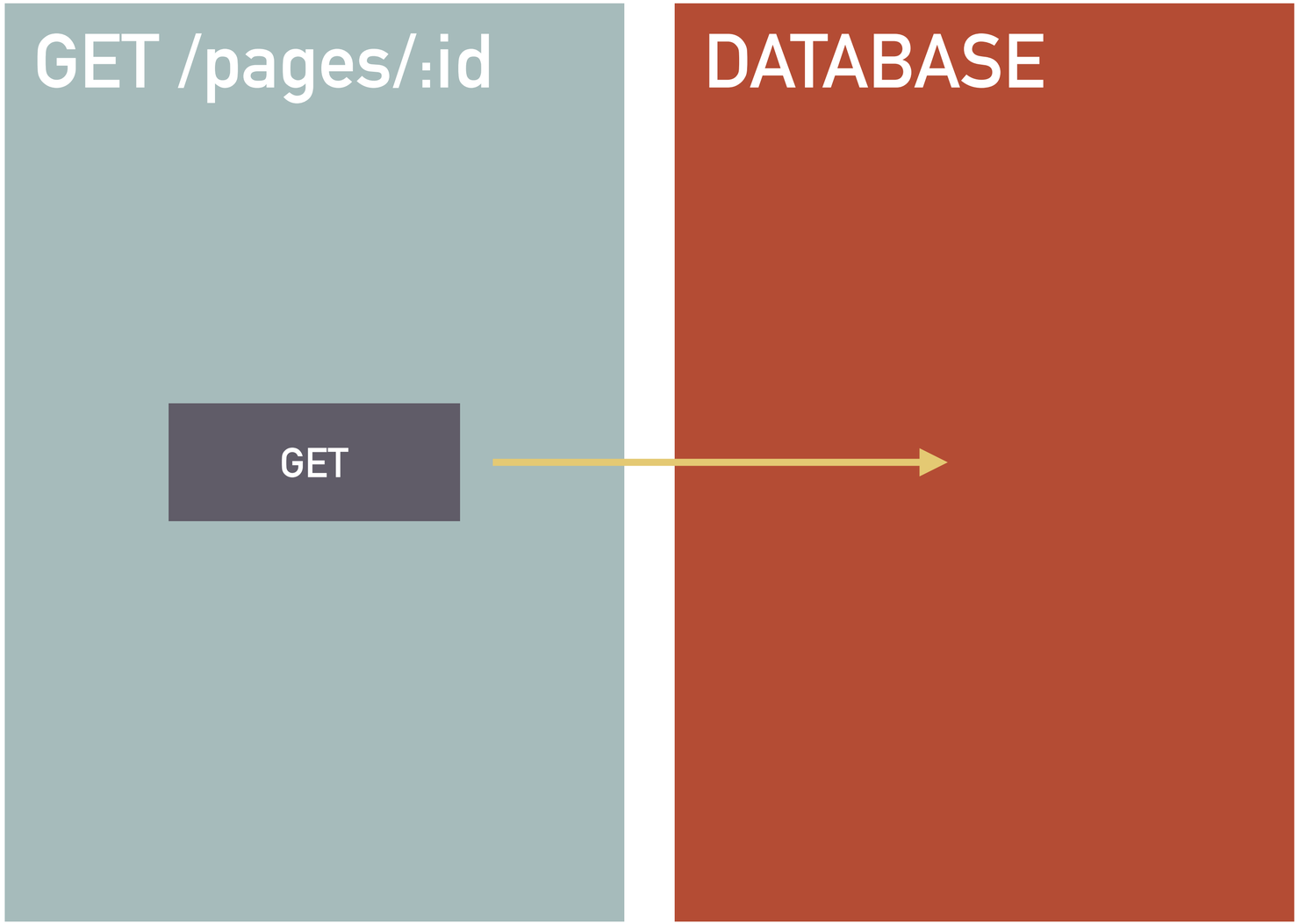
STYLESHEETS

Url	Status	Size
http://www.erlang-solutions.com/assets/designs/design-ba3b1c6f2462823d028b567feea147d89e6d5066eedd7e37e6c548a47c7ea101.css	Fetched	219.02 KB

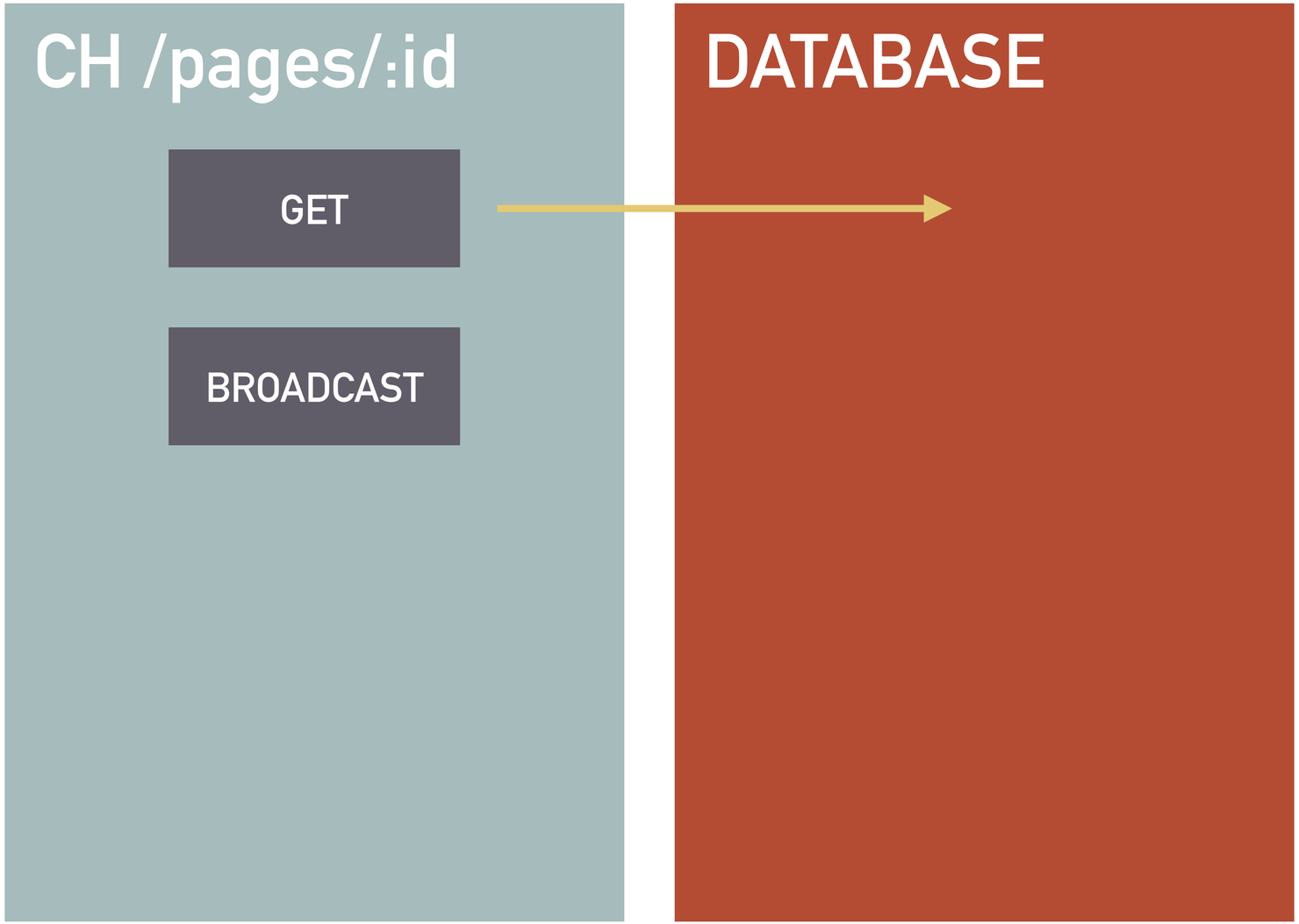
CURRENT STRUCTURE



CURRENT STRUCTURE



CURRENT STRUCTURE

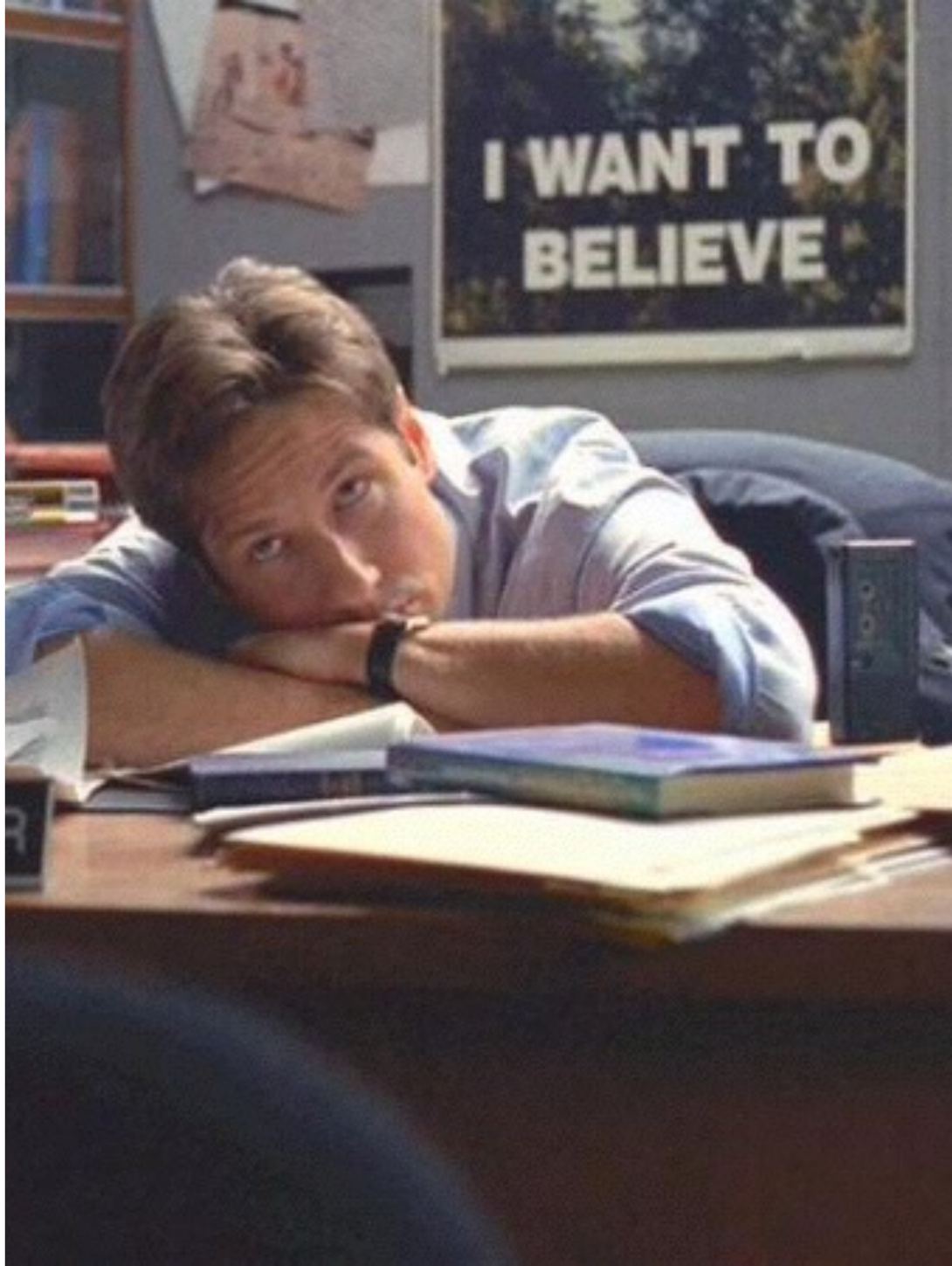


TELEDRIN SCORE™

Not great. Could be better. A for effort.

HARD DEPENDENCY ON THE DATABASE

The truth is out there



OUR GOAL

Libra needs to work during a database outage

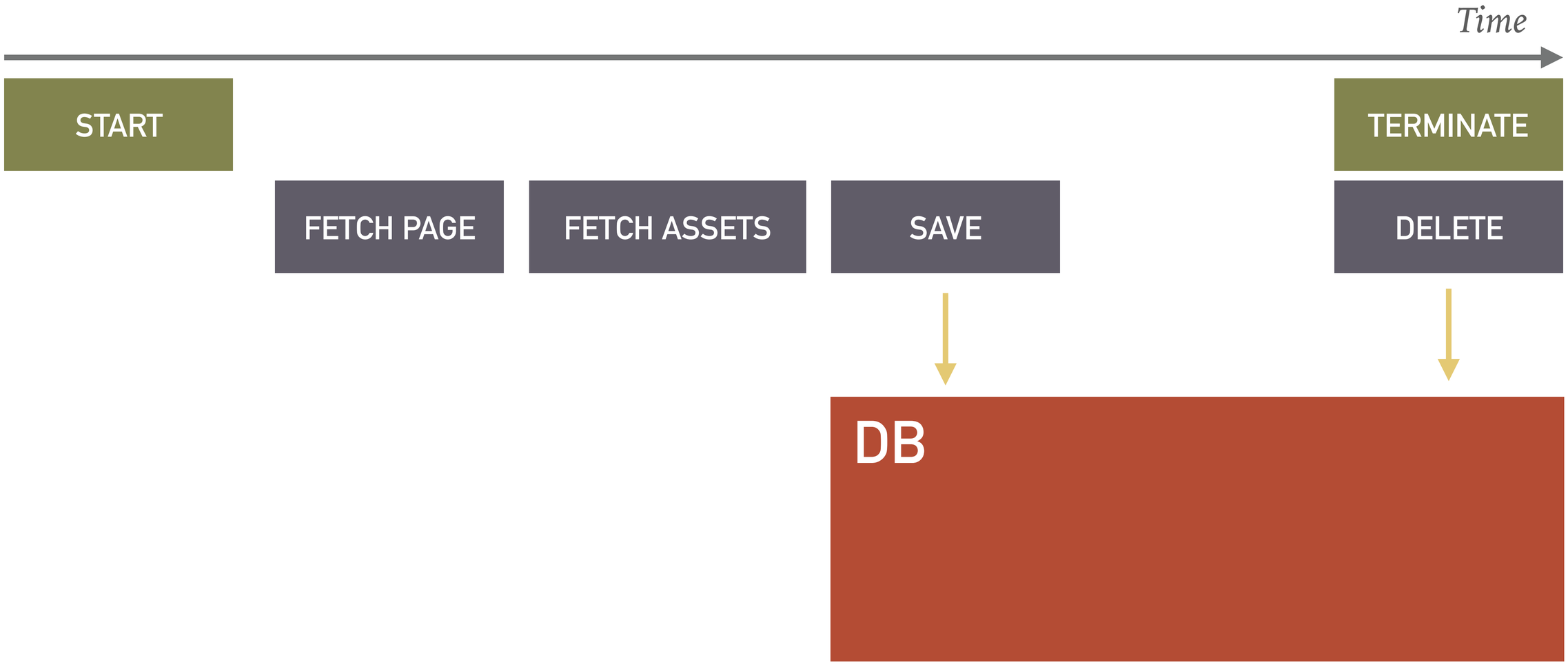
MASTER PLAN

- Keep page and assets data in memory
- Have one process per page
- Each process will manage its own lifecycle
- Each process will try to write the data to the database (as a form of backup)

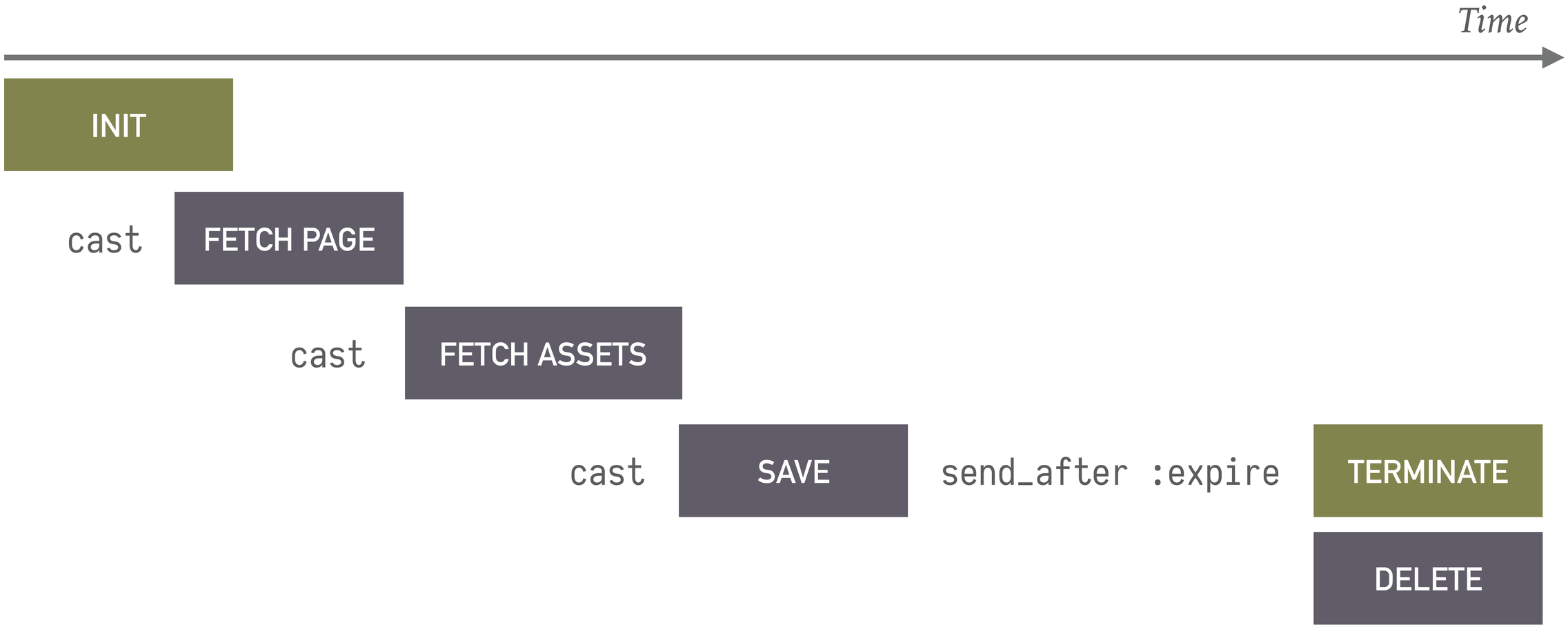
INSPECTOR LIFECYCLE

GenServer in action

INSPECTOR WORKER – LIFECYCLE



INSPECTOR WORKER - GENSERVER IMPLEMENTATION



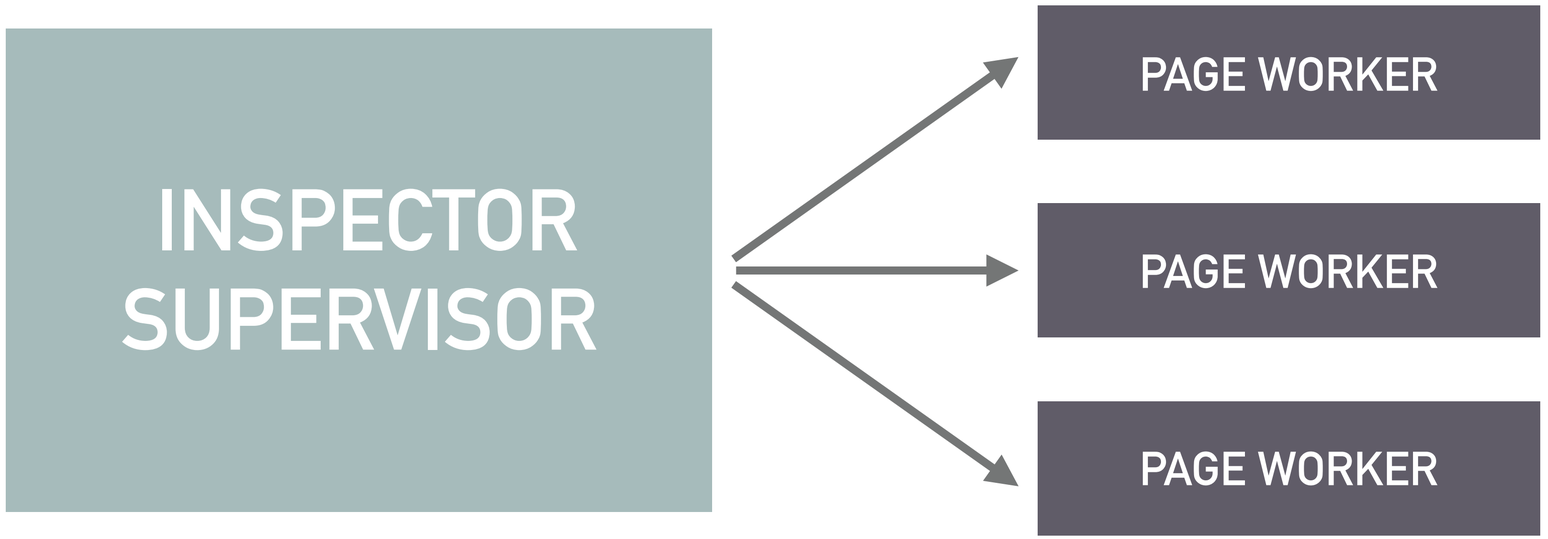
BENEFITS

- Each step is a callback definition
- Clear way to schedule next step
- Each process self-destructs independently (no poll or global scheduler)
- Persistence is an implementation detail (e.g. can swap storage engine)

MANAGING WORKERS

Supervisor in action

INSPECTOR SUPERVISOR – SIMPLE ONE FOR ONE



BENEFITS

- API to spawn new workers
- Workers are monitored and restarted in case of abnormal termination
- Supervisor always knows which children exist: no zombie processes

ROUTING TO A WORKER

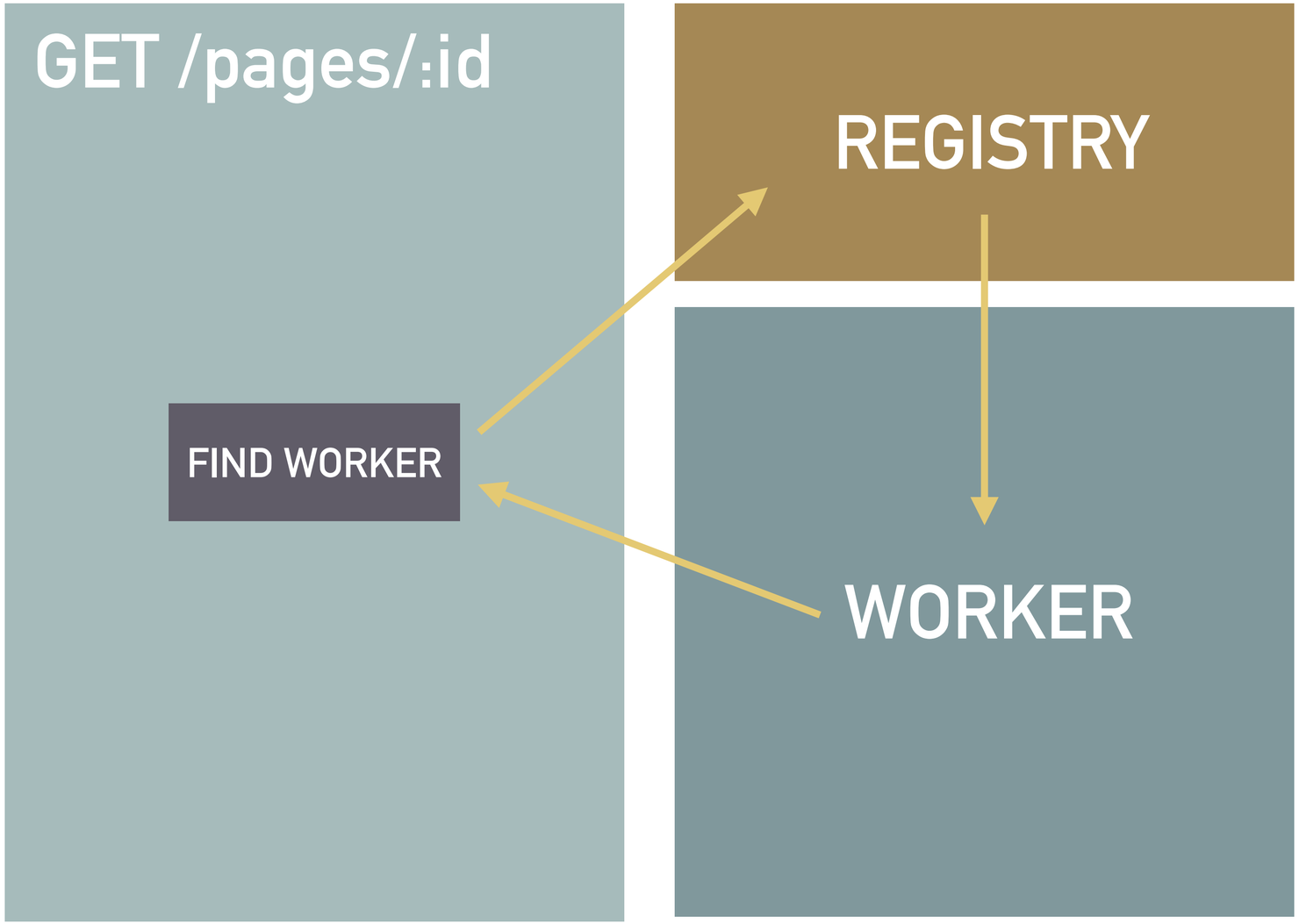
Registry in action

ROUTING TO A WORKER

HOW TO FIND A WORKER

- Each worker needs to have a uuid
- At start, a worker can self-assign a uuid as its own registered name
- To do that, it needs to use a process registry

CURRENT STRUCTURE



BENEFITS

- Each worker manages its own name
- Registry gets automatically updated when workers die
- Elixir > 1.4 ships a local registry (i.e. for a single node) that fits this exact purpose

PUBLIC API

On a need to know basis

HIDE IMPLEMENTATION DETAILS

```elixir
defmodule Libra.Inspector do
  @type url :: String.t
  @type uuid :: String.t

  @spec process_url(url) :: {:ok, uuid} | {:error, term}
  @spec get_page(uuid) :: {:ok, Libra.Page.t} | {:error, :not_found}
end
```

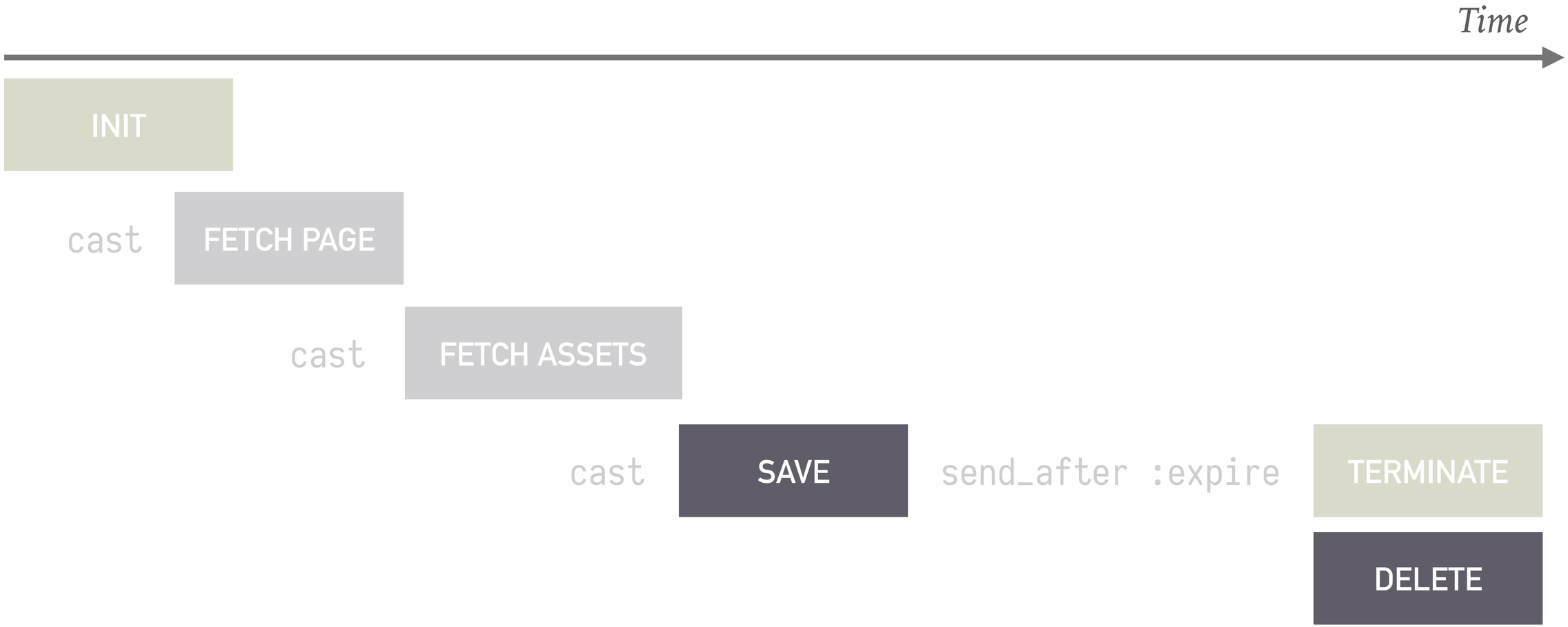
BENEFITS

- Consumer of the API is unaware of the implementation details
- Easily refactor and/or change completely

SURVIVE DB OUTAGES

Try to the rescue!

INSPECTOR WORKER – WRAP RELEVANT CALLS IN A TRY BLOCK



CAVEATS

- Use `try..rescue` sparingly
- Only rescue relevant errors (`DBConnection.ConnectionError`)
- Opportunity to introduce backoff and retry

THE ROAD FROM HERE

RESTORING STATE

- Option 1: at start, read all relevant entries from Postgres and start relevant workers (this needs to be handled in batches, possibly with a dedicated Repo)
- Option 2: lazily spawn workers if relevant database row is found (limits availability)

FROM SINGLE NODE TO MULTI NODE

- How to store state: replicate on each node? Shard it?
- From single node registry to distributed registry
- Deploy components independently, e.g. 2 web nodes and 5 inspector nodes (hint hint umbrella)

DESIGNING FOR AVAILABILITY REQUIRES THINKING

But it's worth the effort

THANKS!

Any questions?

BYE!

Claudio Ortolina

I get paid by Erlang Solutions

claudio.ortolina@erlang-solutions.com

@cloud8421