

Microservices under the



Umbrella

@gemcfadyen

ElixirConf EU 

@MakisOtman

Georgina McFadyen



8th Light

Software Crafter



gemcfadyen



@gemcfadyen



Makis Otman



8th Light

Software Crafter



Maikon



@MakisOtman

Agenda

Plan

Introduction to Umbrella Applications

Architectural Roadmap

Technical Setup

Morning Break

Demo: GenServer

Exercise 1: Build Search

Lunch Break

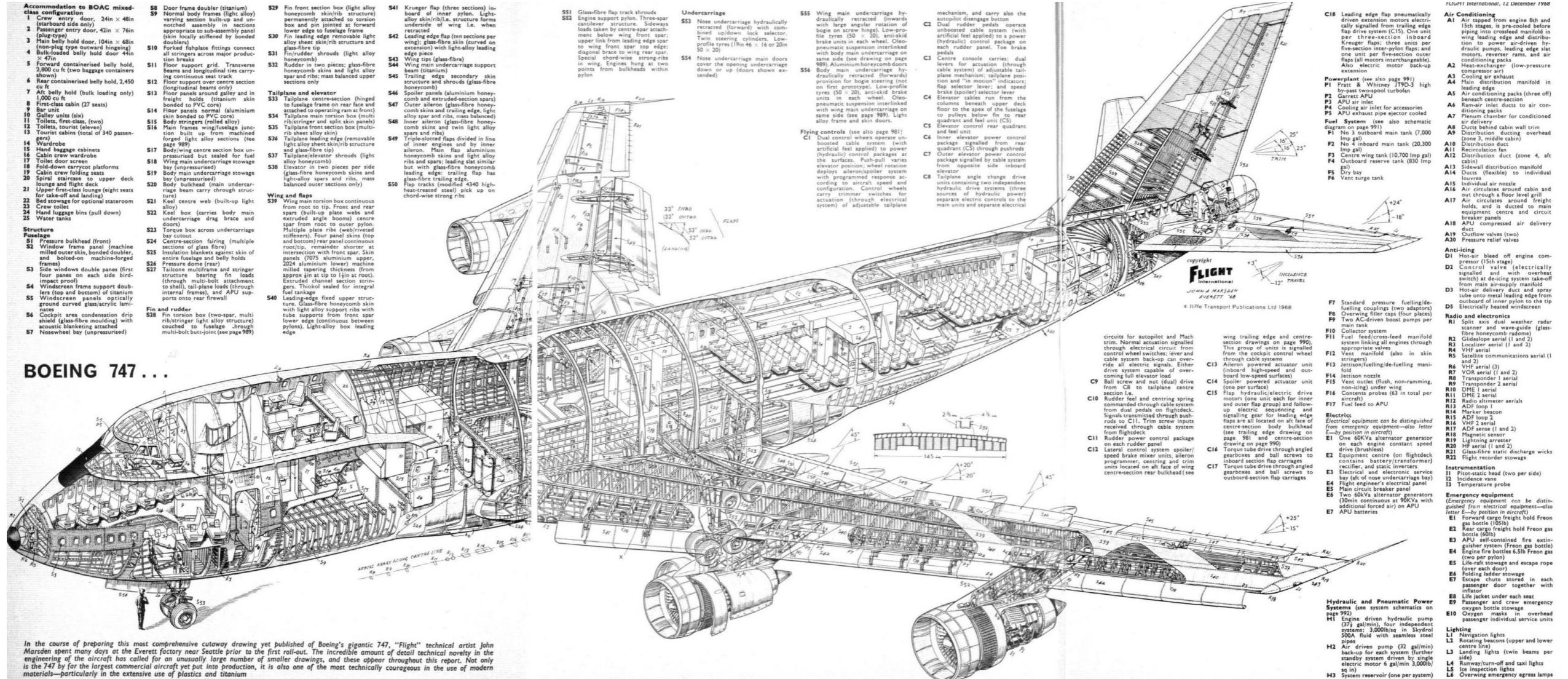
Exercise 2: Add Persistence

Exercise 3: Build and Deploy to AWS

Afternoon Break

Exercise 4: Distributed Umbrella

Wrap up





Umbrella Applications

Creating an Umbrella

```
$ mix new my_app --umbrella
```

Umbrella Structure

```
├── apps
├── config
│   └── config.exs
├── mix.exs
└── README.md
```

Creating Sub-Applications

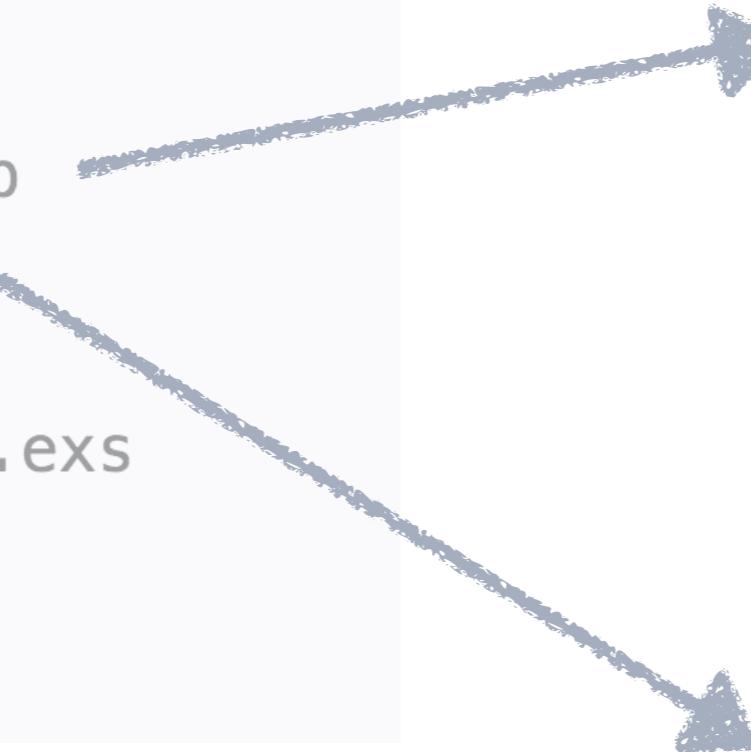
```
$ cd apps
```

```
$ mix phoenix.new web_app
```

```
$ mix new app_1
```

```
├── apps
│   └── web_app
│       └── app_1
├── config
│   └── config.exs
└── mix.exs
    └── README.md
```

```
└── apps
    └── web_app
        └── app_1
    └── config
        └── config.exs
    └── mix.exs
    └── README.md
```



```
└── config
└── lib
    └── web_app
        └── endpoint.ex
        └── repo.ex
    └── web_app.ex
└── mix.exs
└── README.md
└── test
└── web
```

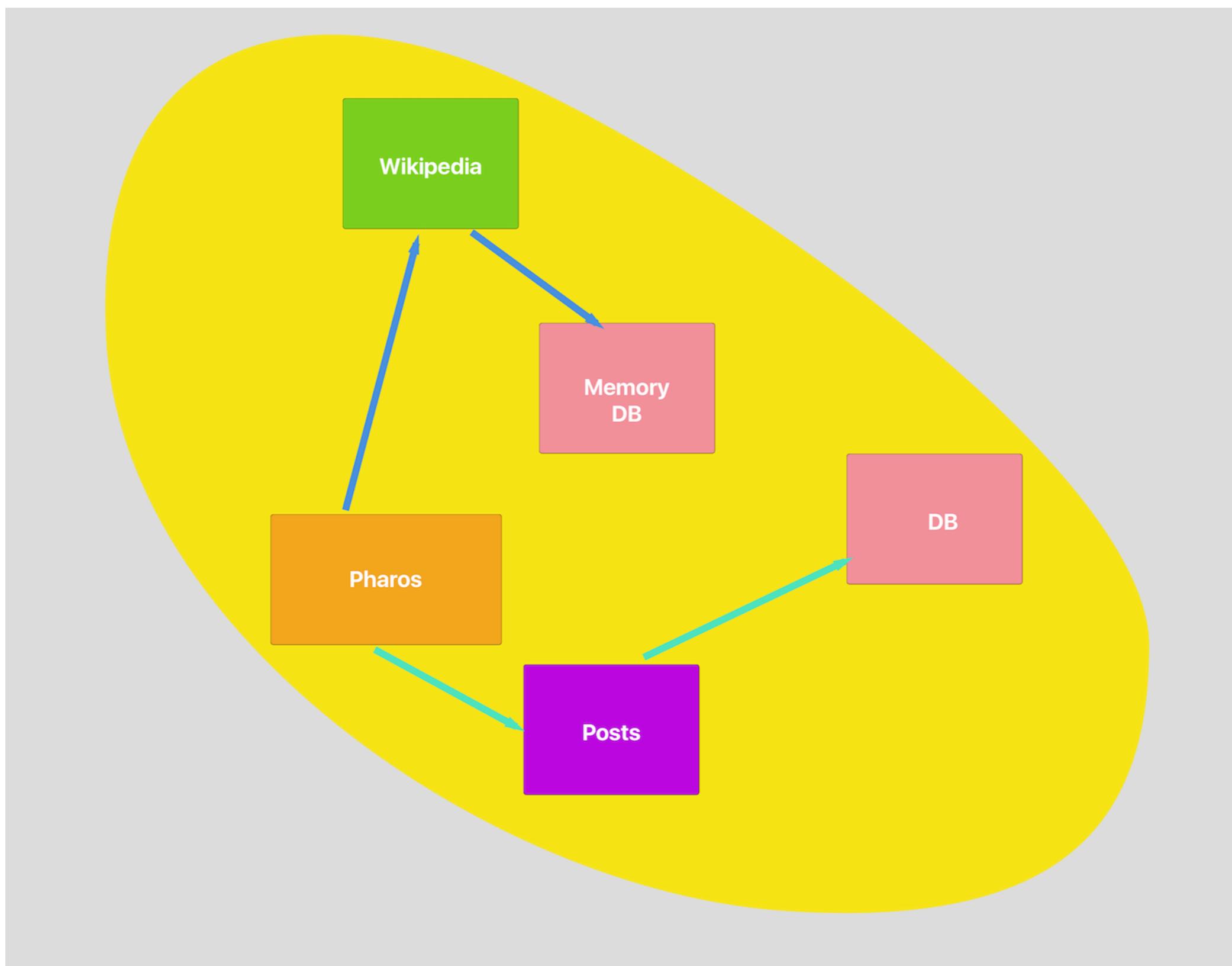
```
└── lib
    └── app_1.ex
└── mix.exs
└── README.md
└── test
    └── app_1_test.exs
    └── test_helper.exs
```

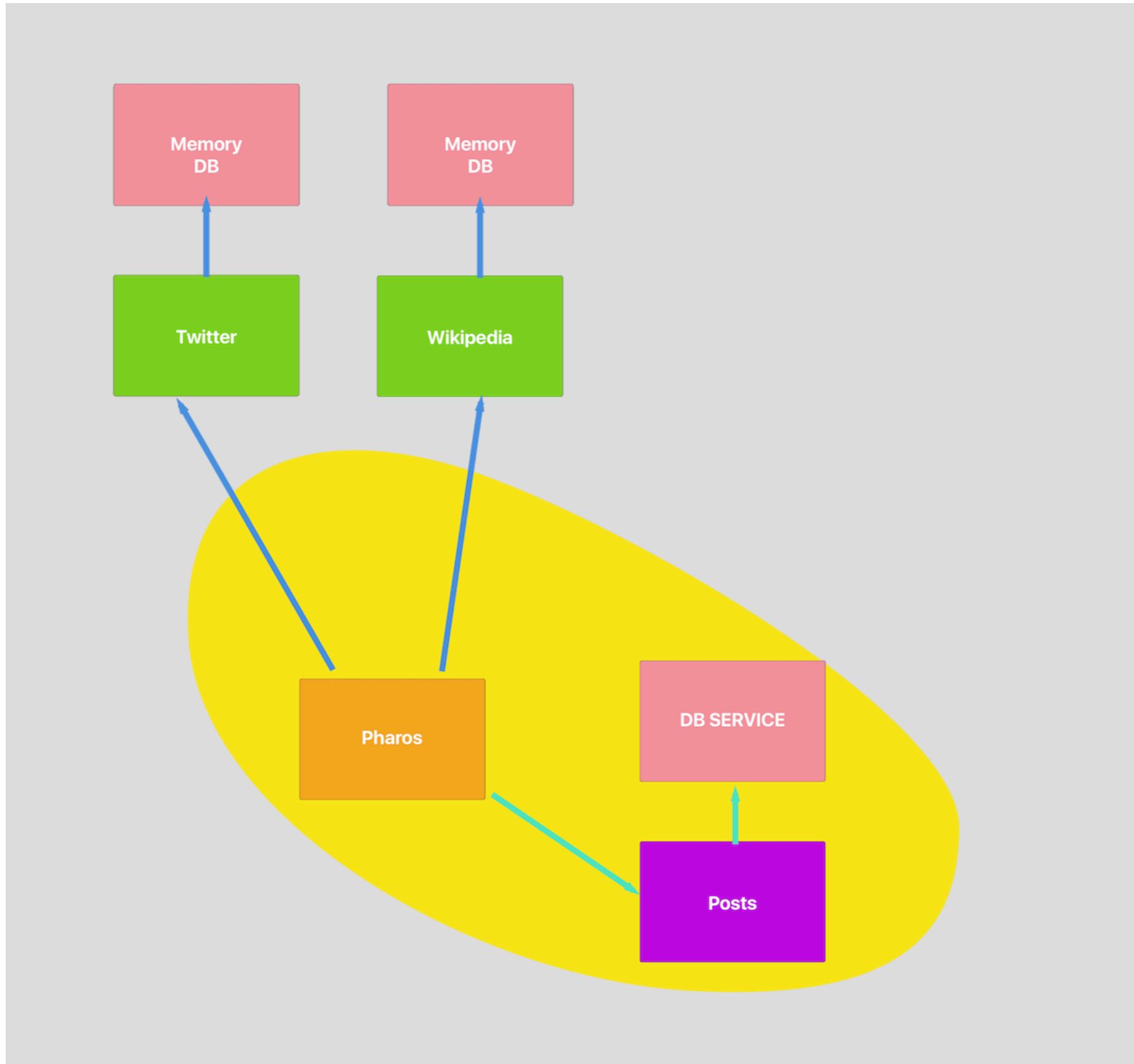
Internal Dependencies

```
# inside web_app/mix.exs
defp deps do
  [
    # dependencies listed here...
    {:app_1, in_umbrella: true}
  ]
end
```



Architectural Roadmap





Technical Setup



Local Environment

Follow the README at:

github.com/Maikon/elixir_setup

Build Search



Posts

Twitter

MemoryDb

DbService

PharosWeb

Wikipedia

Pharos Demo

Objective

Use an API (Twitter, Wikipedia etc) to search for a topic and return the requested number of results.

Starter code:

<https://github.com/Maikon/Pharos>

Contract

Your Search module needs to conform to the following interface:

`YourApi.Search.for_topic(%{topic: “topic”, amount: 3})`

Contract

Your Search module will return the following:

```
=> [%Result{  
    title: "the title",  
    link: "link-to-tweet",  
    description: "tweet content",  
    source: "twitter"  
}]
```

Public API

<https://github.com/toddmotto/public-apis>

OAuth

- Twitter
- Medium
- Guardian
- New York Times
- Reddit
- Vimeo

No Auth

- Wikipedia
- Github-UserData
- GoodReads
- APIs.guru

Objective

Use an API (Twitter, Wikipedia etc) to search for a topic and return the requested number of results.

Starter code:

<https://github.com/Maikon/Pharos>

Solution at: [exercise-i-wiki-search](#)

Add Persistence

Gen Server

Gen Server

Elixir Process abstracting client-server interactions

Keeps State in memory

Defers decisions about using a database until later

Provides a set of interface methods

Gen Server

start_link/3

start_link(module, initial_state, options \\ [])

Gen Server

use GenServer

Client-Server

```
GenServer.call(server, request, timeout)  
GenServer.handle_call(request, from, state)
```

Client-Server

`GenServer.cast(server, request)`

`GenServer.handle_cast(request, state)`

GenServer Demo

Checkout branch [demo-of-gen-server](#)

Agents

Agent Demo

Checkout branch demo-of-agent

GenServer vs Agent

Agents store state

Agents require less code

GenServer can be used as an abstraction for building a server

GenServer methods are split between client/server

Objective

Introduce an in memory database for the search results using a GenServer or Agent

Solution at: [exercise-2-in-memory-db](#)

Testing with Fakes

External Dependencies

External Systems are unpredictable

Api Limits

Unexpected Down time

Test Double Demo

Checkout branch demo-use-fake-search-for-testing-purposes

Objective

Introduce a fake search for the test environment,
ensuring the real search is used in all other
environments



Deployment



Build & Deploy

Follow “Deployment” wiki page at:

github.com/Maikon/elixir_setup/wiki/Deployment

Checkout branch **exercise-3-build-pharos-release**

Distributed Erlang

github.com/Maikon/pharos

Checkout branch exercise-4-distributed-umbrella

“vm.args” File

```
-name <node-name>@<host>
-setcookie <cookie>
-kernel inet_dist_listen_min 9100
inet_dist_listen_max 9155
```

Example location: pharos/apps/<your-app>/vm.args

- * node name example: ec2-54-197-0-118.compute-1.amazonaws.com
- * setcookie sets the cookie to be used for verifying & connecting nodes
 - * Ports 9100-9155 are used by the Erlang distribution protocol

Summary

Umbrella Pros

Single Repository

Modular Approach

Loose Coupling

Flexible Deployment

Strategies

Umbrella Cons

Dependency Conflicts

Process Ownership

Namespace Clashes

Tooling not fully evolved

Opportunity to improve
resources



Thank-You!

@gemcfadyen

ElixirConf EU

@MakisOtman

Resources

