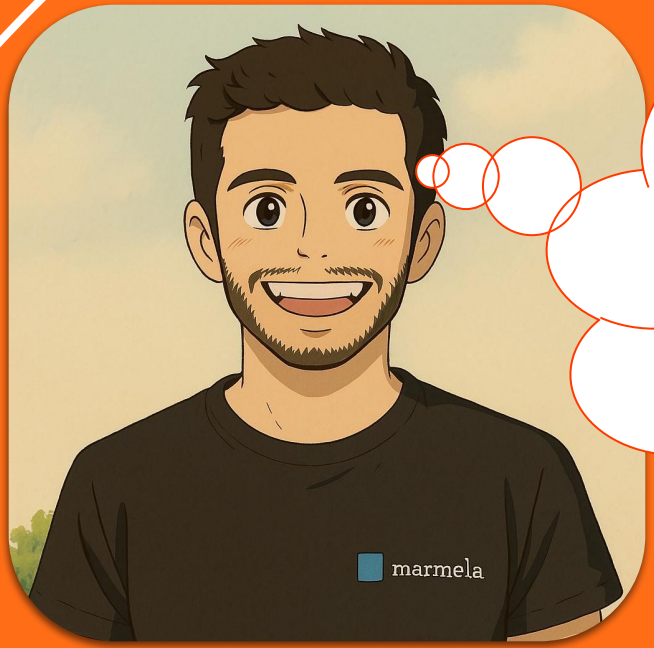
Functional Web Development with Phoenix LiveView

HELLO, I AM NELSON



neovim
inertia
elixir
git
Linux
SQLite
podman
Phoenix Framework

@nelsonmestevao

HELLO, I AM NELSON 👋

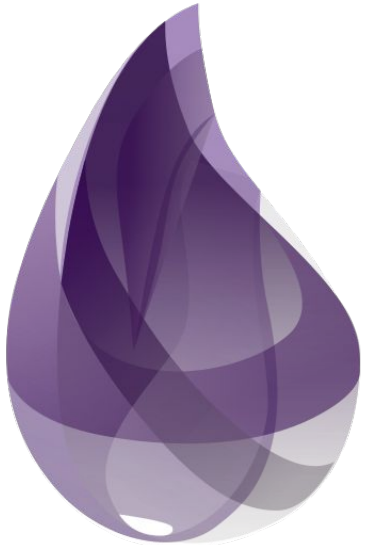
@nelsonmestevao

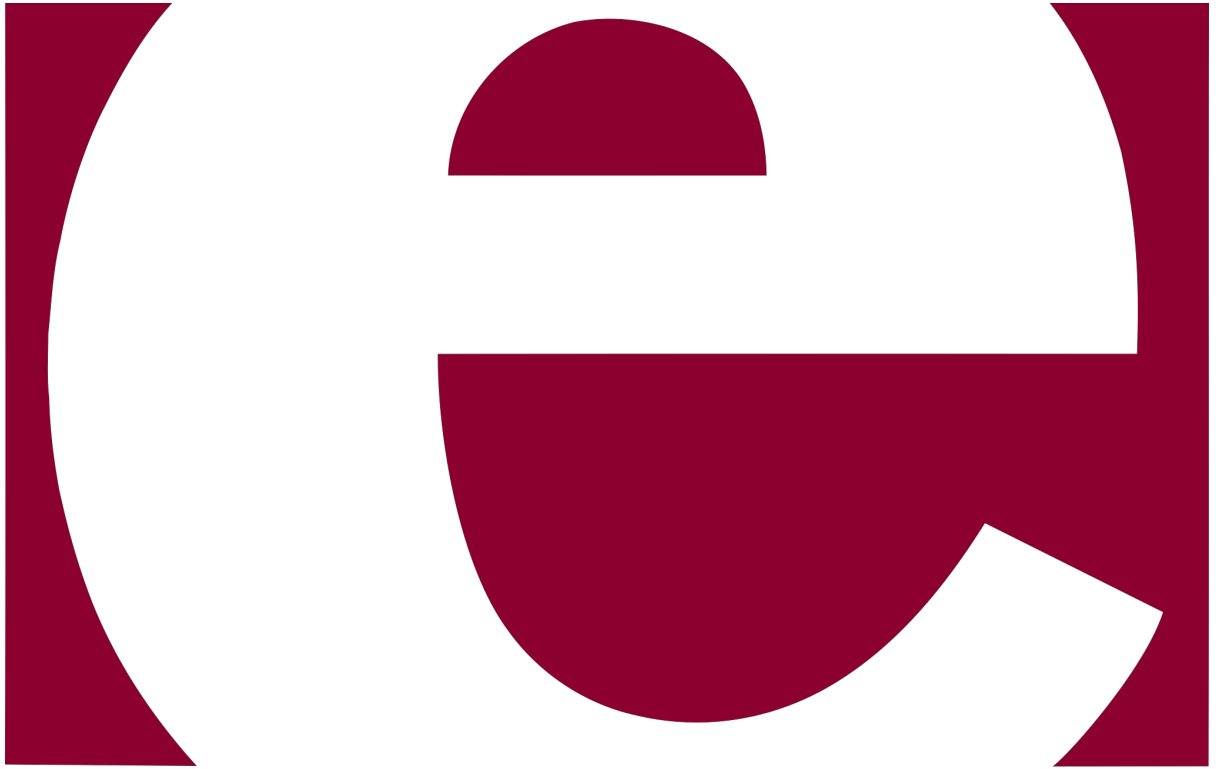
HELLO, I AM NELSON

neovim
inertia
elixir
git
Linux
SQLite
podman
Phoenix Framework

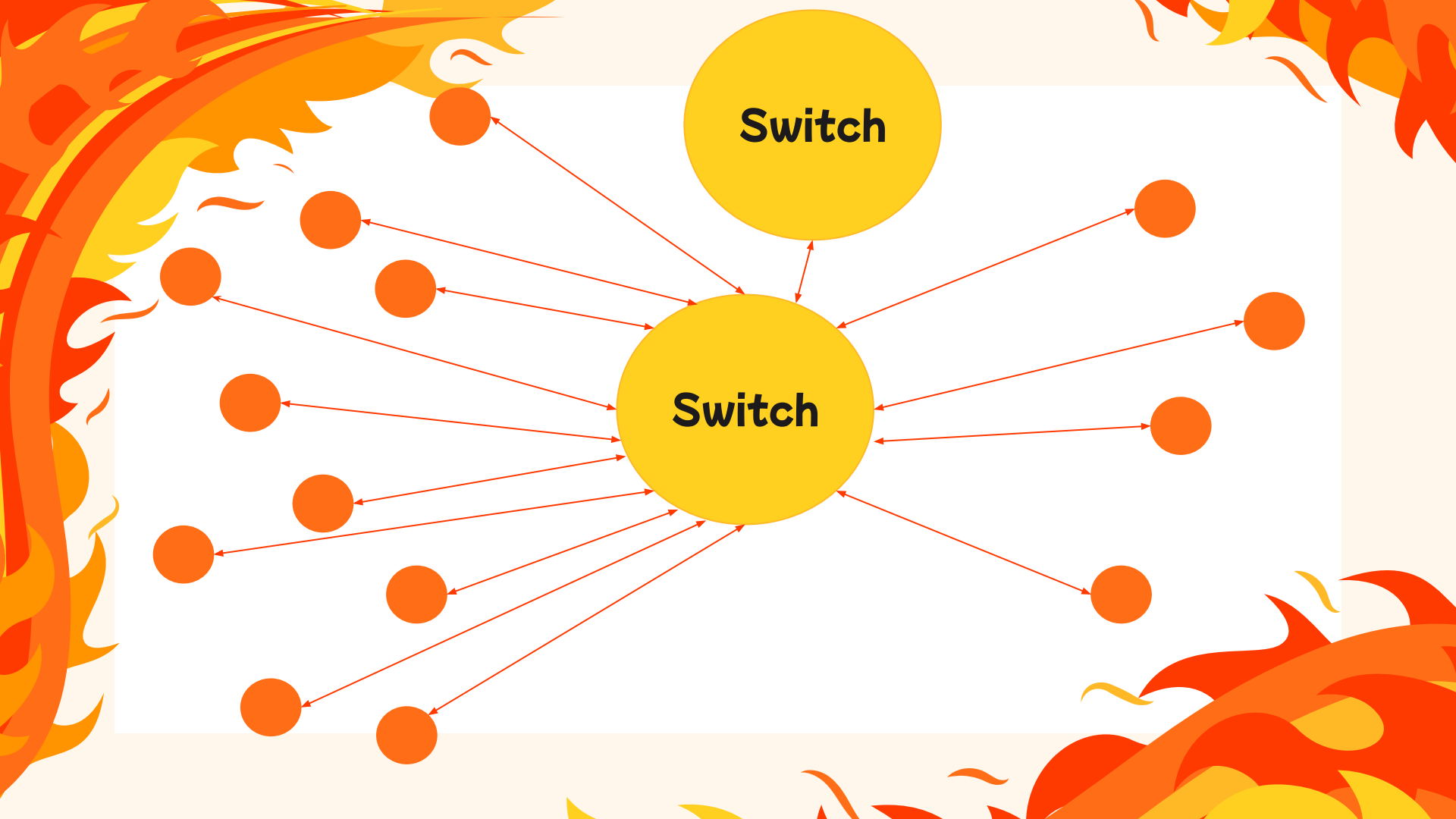
@nelsonmestevao

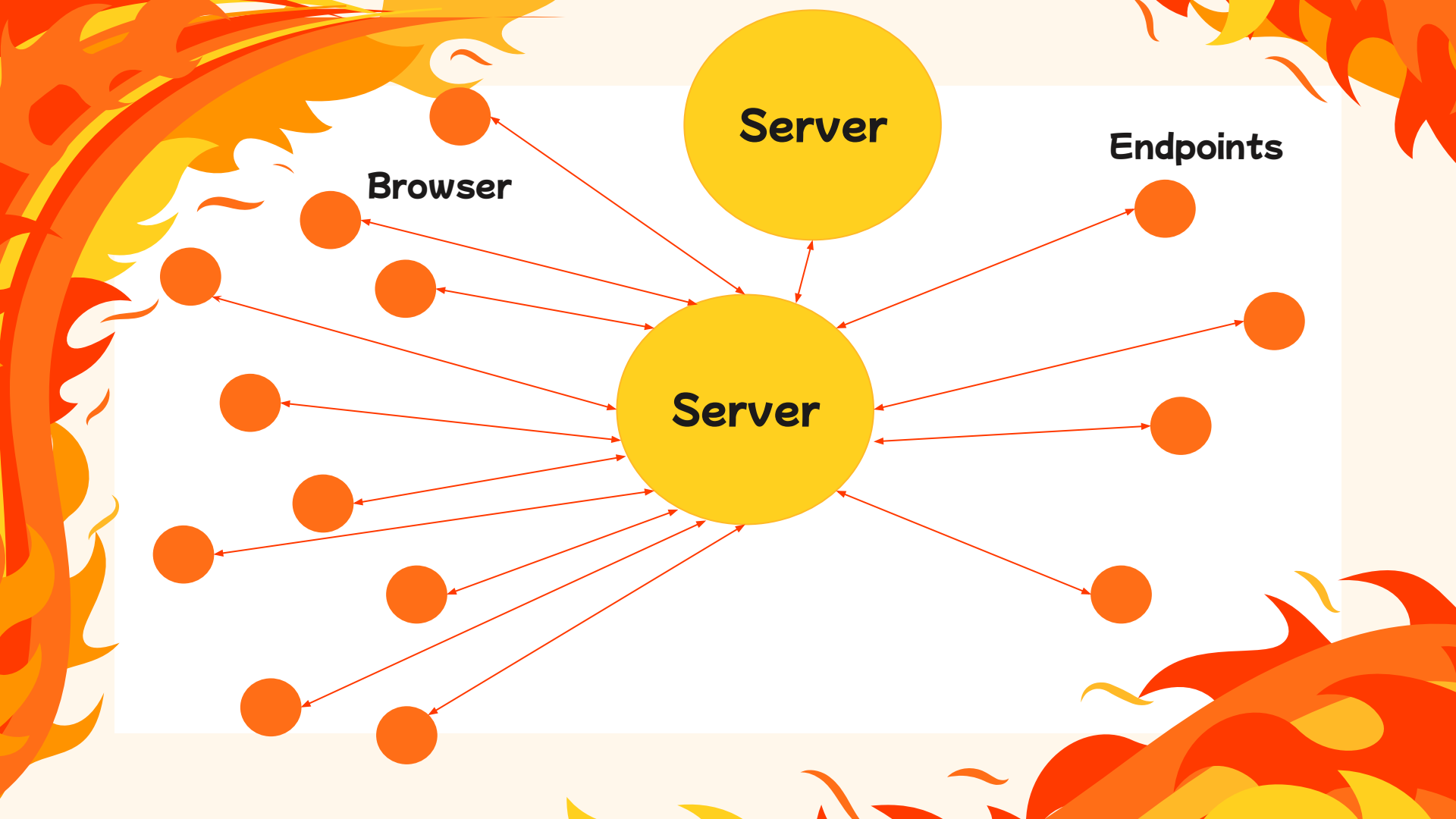
Elixir: A Programming Language for the Current Web



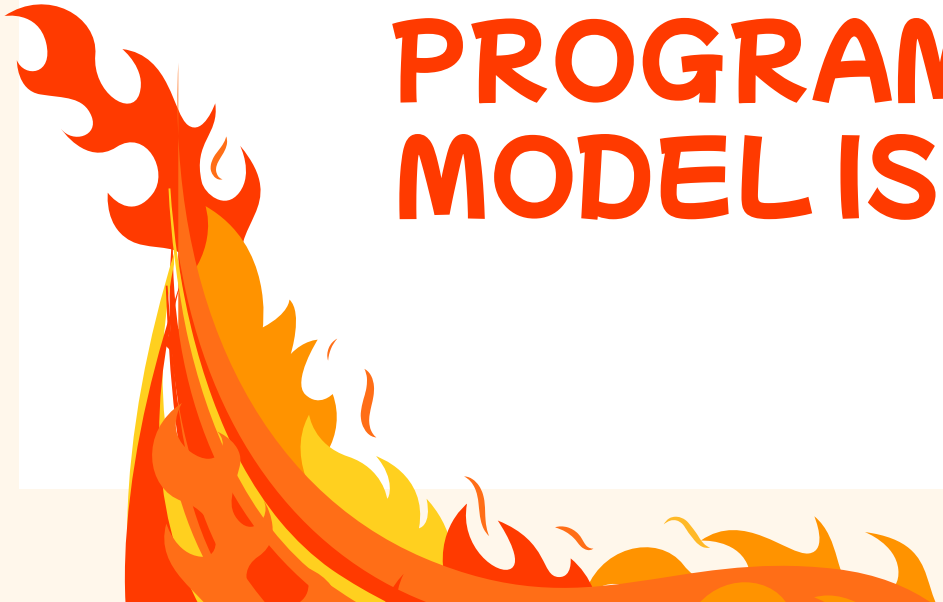


ERLANG

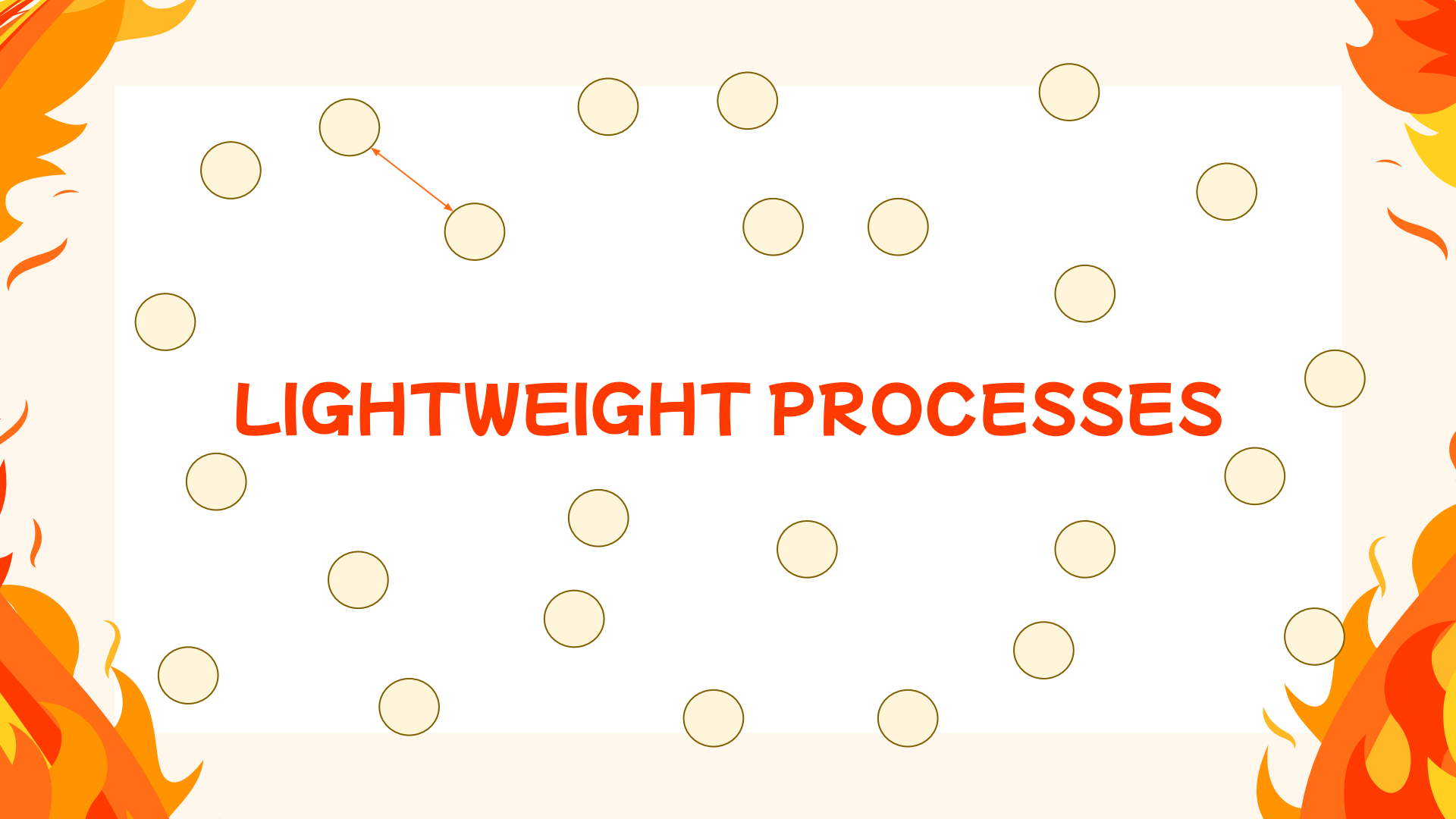

Switch
Switch

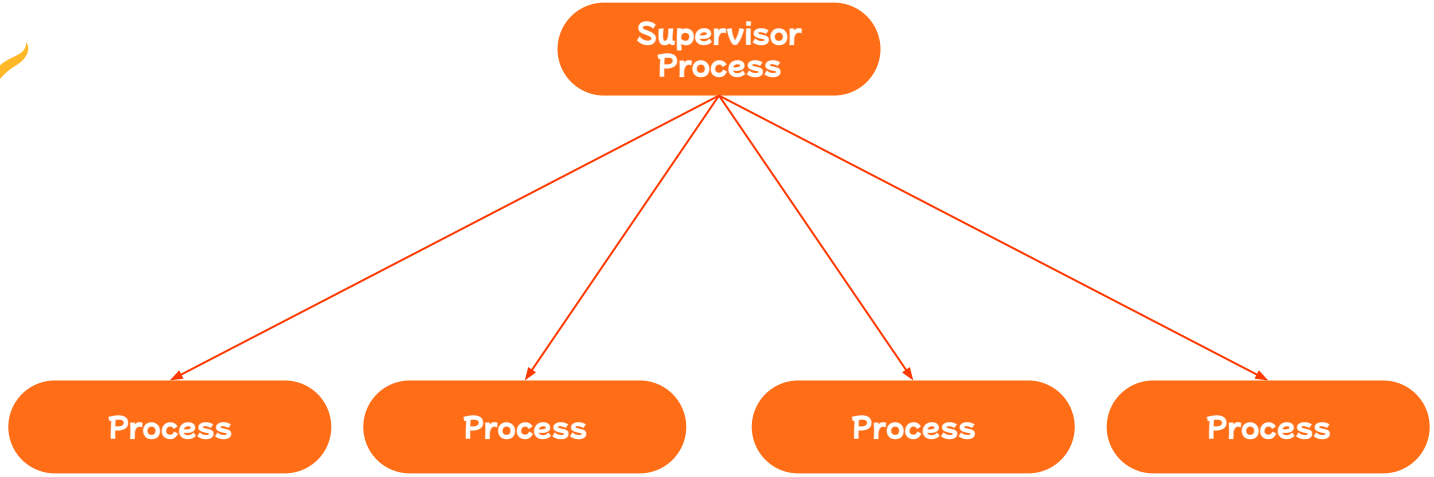

Server
Endpoints
Browser
Server

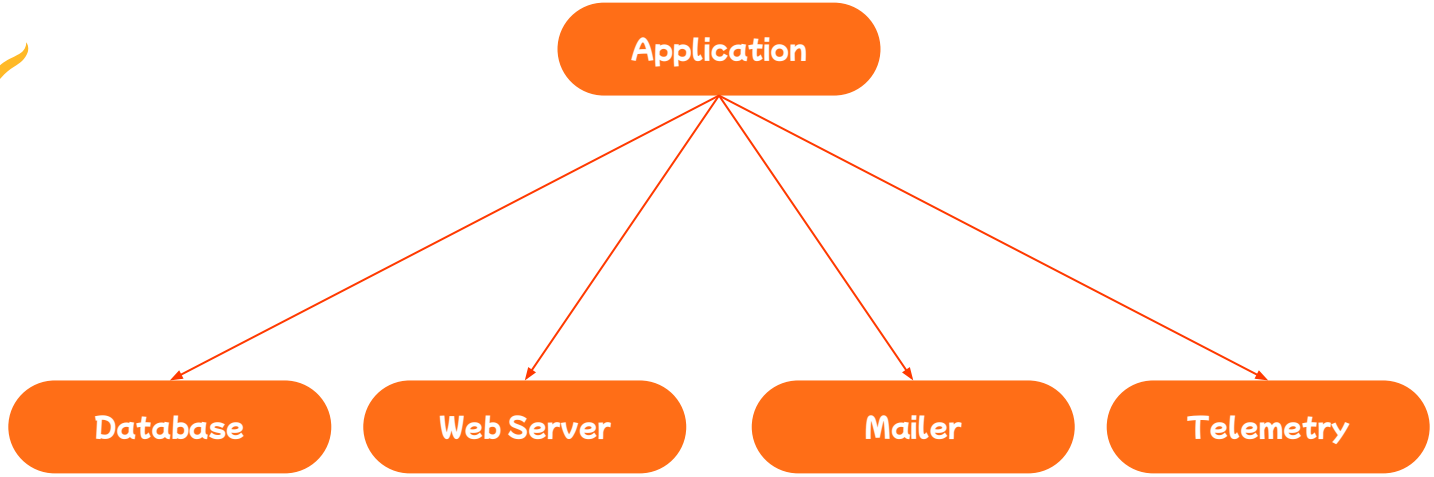
A NEW PROGRAMMING MODEL IS BORN



A NEW PROGRAMMING MODEL


LIGHTWEIGHT PROCESSES


Supervisor Process
Process
Process
Process
Process


Application
Database
Web Server
Mailer
Telemetry

is so great!

Then why bother with  elixir ?

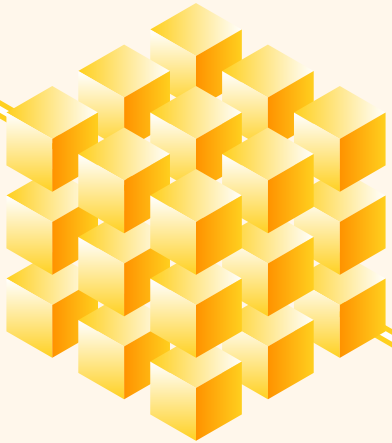
Key Features of Elixir

Functional
Immutable data structures

Scalability
Lightweight and isolated processes

Fault-Tolerance
hierarchical process structure with a supervision tree

Concurrent
Communicate by sending and receiving messages



Dynamic
Hot swapping ability

Distributed
Across multiple machines with same programming model

Metaprogramming
Flexibility and extensibility

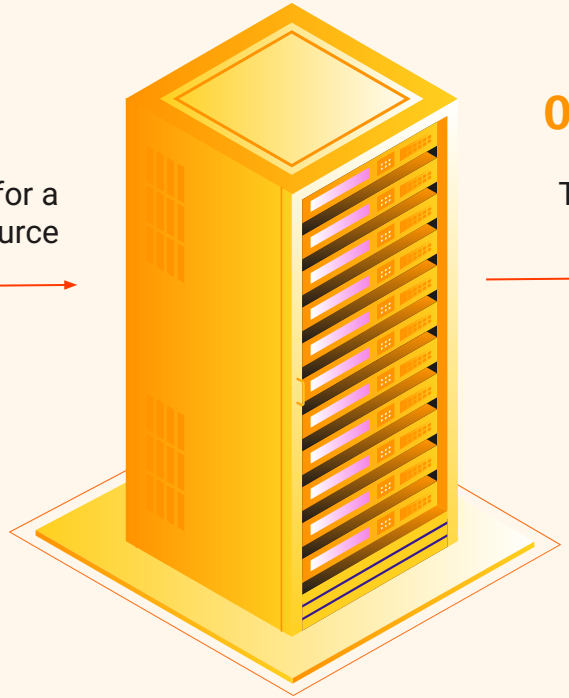
Tooling
Great developer tools and a growing ecosystem of libraries

What is Functional Programming?

HTTP: The Heart of Web Communication

Incoming Request

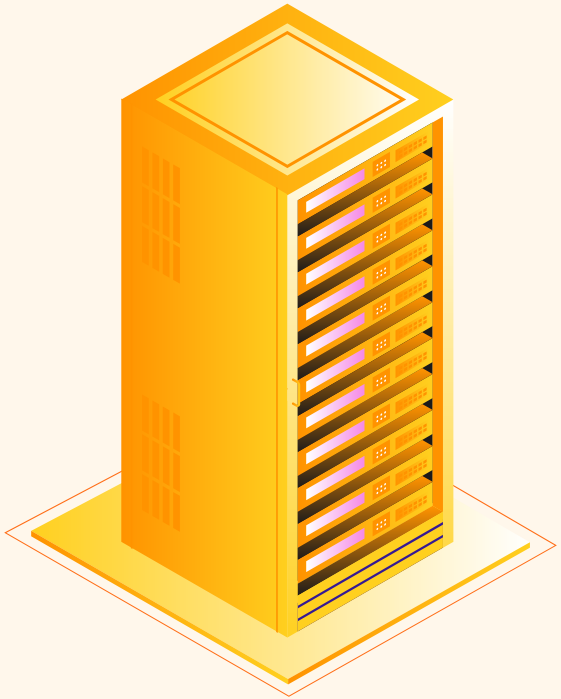
The server receives a HTTP request for a given resource



Outcoming Response

The server builds the HTTP response and sends it back to the client

LiveView: Leveraging WebSockets



After first render, upgrade connection

The WebSocket allows the server to push updates to the client, and the client can also send events to the server. This enables real-time interactions without needing to refresh the page.

Key Concepts of Functional Programming

Immutability

Once a variable is declared, it cannot be changed or mutated

Function Composition

The process of using the output of one function as the input to another function

Recursion

Functions call themselves until they reach a base case

High-Order Functions

A high-order function is a function that can take other functions as arguments, or return a function as a result

Immutability

```
animals = [:bear, :elephant]

Enum.reject(animals, fn animal -> animal == :bear)
IO.inspect(animals)
#> [:bear, :elephant]

animals = Enum.reject(animals, fn animal -> animal == :bear)
IO.inspect(animals)
#> [:elephant]
```

Immutability

```
animals = [:bear, :elephant]

Enum.reject(animals, fn animal -> animal == :bear)
IO.inspect(animals)
#> [:bear, :elephant]

animals = Enum.reject(animals, fn animal -> animal == :bear)
IO.inspect(animals)
#> [:elephant]
```

Immutability

```
animals = [:bear, :elephant]

Enum.reject(animals, fn animal -> animal == :bear)
IO.inspect(animals)
#> [:bear, :elephant]

animals = Enum.reject(animals, fn animal -> animal == :bear)
IO.inspect(animals)
#> [:elephant]
```

Function Composition

```
defmodule Math do
  def add(a, b), do: a + b

  def square(x), do: x * x
end
```

Function Composition

```
defmodule Math do
  def add(a, b), do: a + b

  def square(x), do: x * x
end

2
|> Math.add(3)
|> Math.square()
#> 25
```

Recursion

```
defmodule Math do
  def factorial(0), do: 1
  def factorial(n), do: n * factorial(n - 1)
end
```

High-Order Functions

```elixir
defmodule MyAppWeb.PageController do
  def index(conn, _params) do
    products = [
      %{name: "Product 1", price: 10, on_sale: true},
      %{name: "Product 2", price: 20, on_sale: false},
      %{name: "Product 3", price: 30, on_sale: true}
    ]
    |> Enum.filter(products, fn product -> product.on_sale end)

    render(conn, "index.html", title: "My App", items: sale_products)
  end
end
```

LET'S GET PRACTICAL

Elixir is OP

Reliable distributed systems

IoT Communication Hubs

Media Pipelines and Streaming

Machine Learning And Data Analysis

Real-Time WEB APPS

Collaborative Tools

Success stories of Elixir

LEARN MORE

Elixir is a dynamic, functional language for building scalable and maintainable applications.

Elixir runs on the Erlang VM, known for creating low-latency, distributed, and fault-tolerant systems. These capabilities and Elixir tooling allow developers to be productive in several domains, such as web development, embedded software, machine learning, data pipelines, and multimedia processing, across a wide range of industries.

Here is a peek:

```
iex> "Elixir" |> String.graphemes() |> Enum.frequencies()
%{"E" => 1, "i" => 2, "l" => 1, "r" => 1, "x" => 1}
```

Check our Getting Started guide and our Learning page to begin your journey with Elixir. Or keep scrolling for an overview of the platform, language, and tools.

Companies using Elixir in production

See more cases →

VEEPS

SPARKMETER

MINISTÈRE DE LA TRANSITION ÉCOLOGIQUE
Liberté
Égalité
Fraternité

News: Elixir v1.17 released

Search...

IMPORTANT LINKS

- Development & Team
- Source code & issues tracker

Watch the Elixir mini-documentary!

UPCOMING EVENTS

Elixir Stream Week

- **Oct 21-25, 2024** - Remote
- **5 days, 5 streams, 5 Elixir experts**

JOIN THE COMMUNITY

- Hex.pm package manager
- @elixirlang on Twitter
- #elixir on irc.libera.chat


goto;
SAŠA JURIĆ
The Soul of
Erlang & Elixir

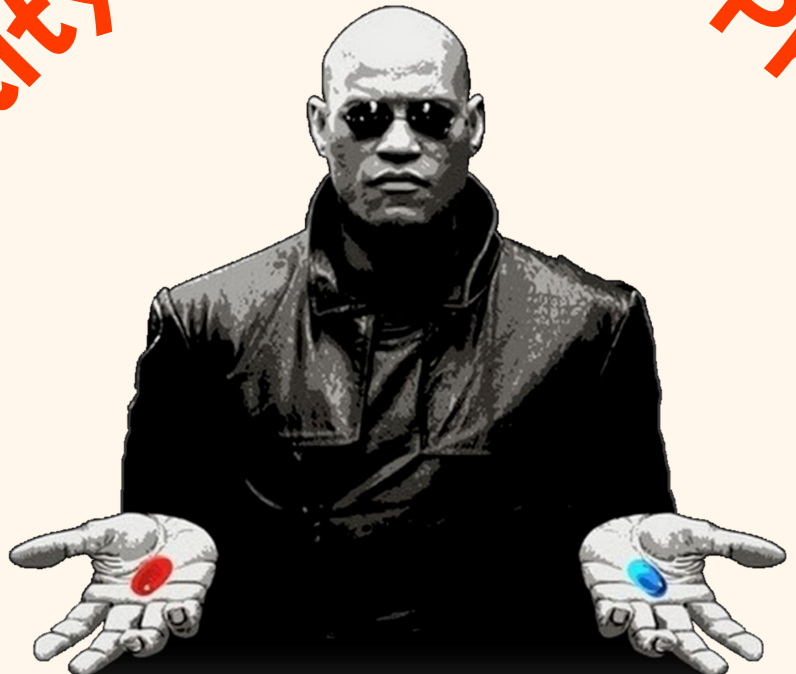
ALCHEMIST CAMP

METACAST

ElixirCasts


exercism

Scalability



Productivity

Functional Web Development with Phoenix LiveView