

μSvcs with Elixir

Rodrigo Nonose

μSvcs with Elixir

Rodrigo Nonose

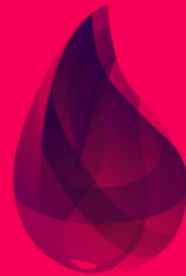
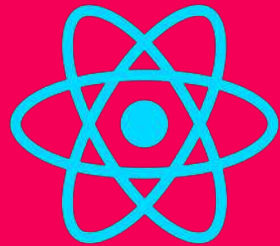








emcasa.com

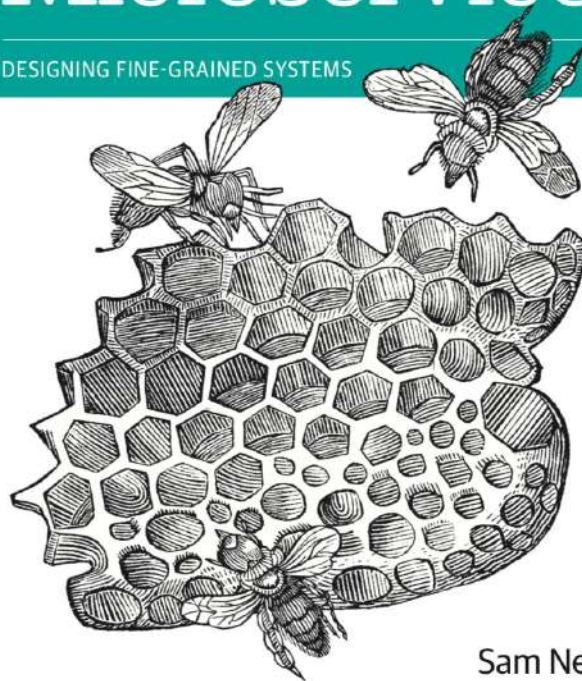


μSvcs



Building Microservices

DESIGNING FINE-GRAINED SYSTEMS



Sam Newman



Replaceability

Single Responsibility Principle

Composability

Organizational Alignment

Ease of Deployment

Scaling

Autonomy

Resilience

Technological Heterogeneity

Elixir



```
def sluggify(string) do
  string
  |> String.split(" ")
  |> Enum.map(&String.normalize(&1, :nfd))
  |> Enum.map(&String.replace(&1, ~r/\W/u, ""))
  |> Enum.join("-")
  |> String.downcase()
end
```



Replaceability

Single Responsibility Principle

Composability

Organizational Alignment

Ease of Deployment

Scaling

Autonomy

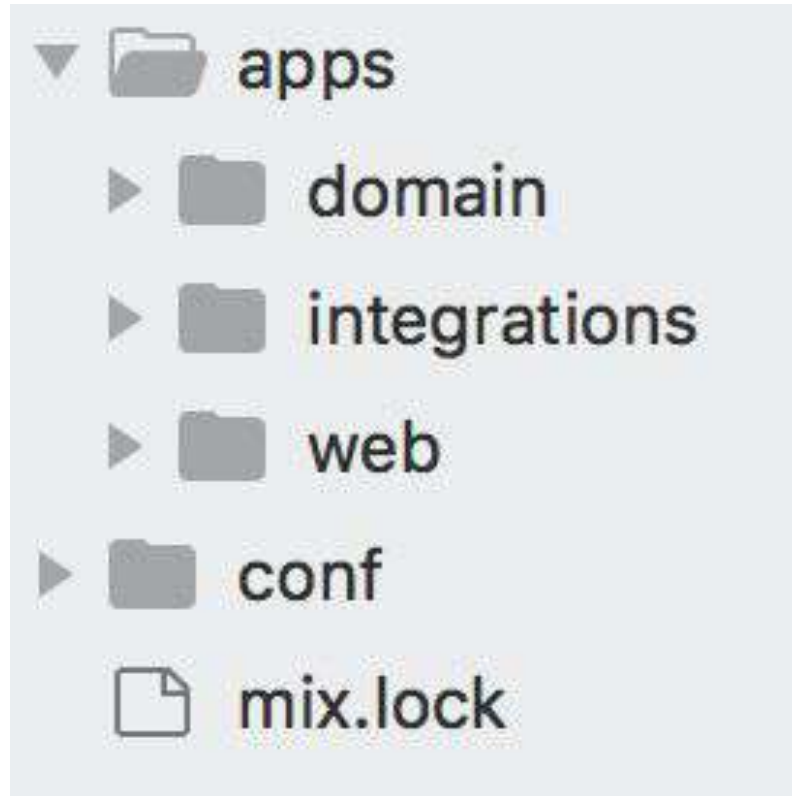
Resilience

Technological Heterogeneity

Umbrella Applications



Umbrella applications





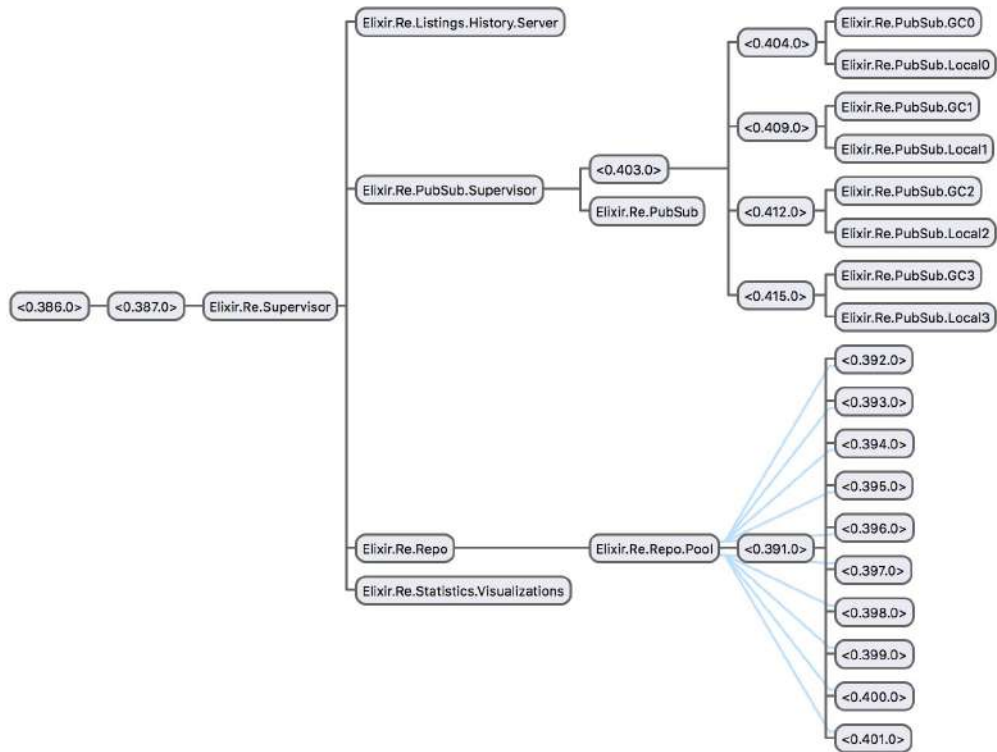
```
defmodule Domain.Application do
  use Application

  def start(_type, _args) do
    children =
      [
        supervisor(Domain.Repo, []),
      ]

    opts = [strategy: :one_for_one, name: Domain.Supervisor]
    Supervisor.start_link(children, opts)
  end
end
```

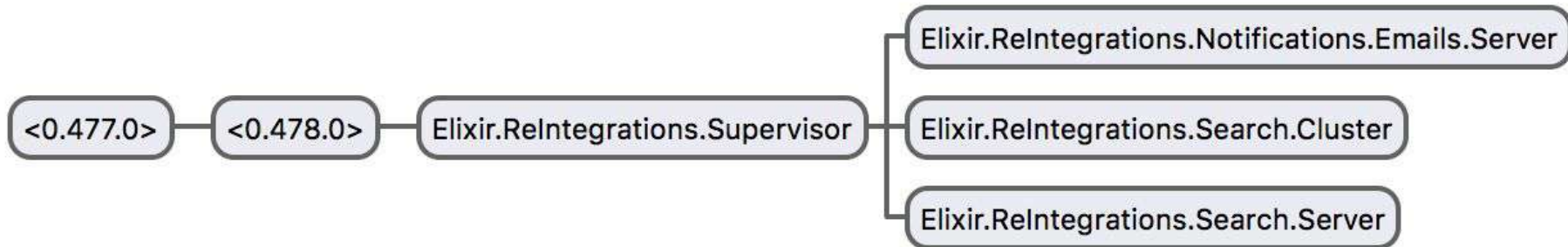



Umbrella applications





Umbrella applications





```
Repo.all(Listing)
```

```
:rpc.call(node, :Repo, :listing, [:"Listing"])
```



Replaceability

Single Responsibility Principle

Composability

Organizational Alignment

Ease of Deployment

Scaling

Autonomy

Resilience

Technological Heterogeneity

Clustering



```
iex(b1@Rodrigos-MacBook-Pro-2)3> Node.self()  
:"b1@Rodrigos-MacBook-Pro-2"  
iex(b1@Rodrigos-MacBook-Pro-2)4> Node.list()  
[:"b2@Rodrigos-MacBook-Pro-2", : "b3@Rodrigos-MacBook-Pro-2"]  
iex(b1@Rodrigos-MacBook-Pro-2)5>
```



Clustering

- Node takeover
- Node failover
- Load-balancing and process distribution
- Stateful process hand-off
- Work stealing
- Cluster healing



Replaceability

Single Responsibility Principle

Composability

Organizational Alignment

Ease of Deployment

Sealing

Autonomy

Resilience

Technological Heterogeneity

Ports



Replaceability

Single Responsibility Principle

Composability

Organizational Alignment

Ease of Deployment

Scaling

Autonomy

Resilience

Technological Heterogeneity

Domain-Driven Design



By Frits Ahlefeldt

Thank you

 rhnonose

 backendwizard

 EmCasa

