

Supervisors

Section 7

In this Section, we are going to take a look at...

- Supervisors and tree
- Implementing supervisors
- Strategies

The Supervisor Behaviour

In this Video, we are going to take a look at...

- Recap linked processes
- What is the supervisor behaviour
- How supervisors work
- The supervision tree

Recap – Application and Linked Processes

Application

#PID<0.80.0>

#PID<0.81.0>

#PID<0.82.0>

Recap – Application and Linked Processes

Application



Recap – Application and Linked Processes

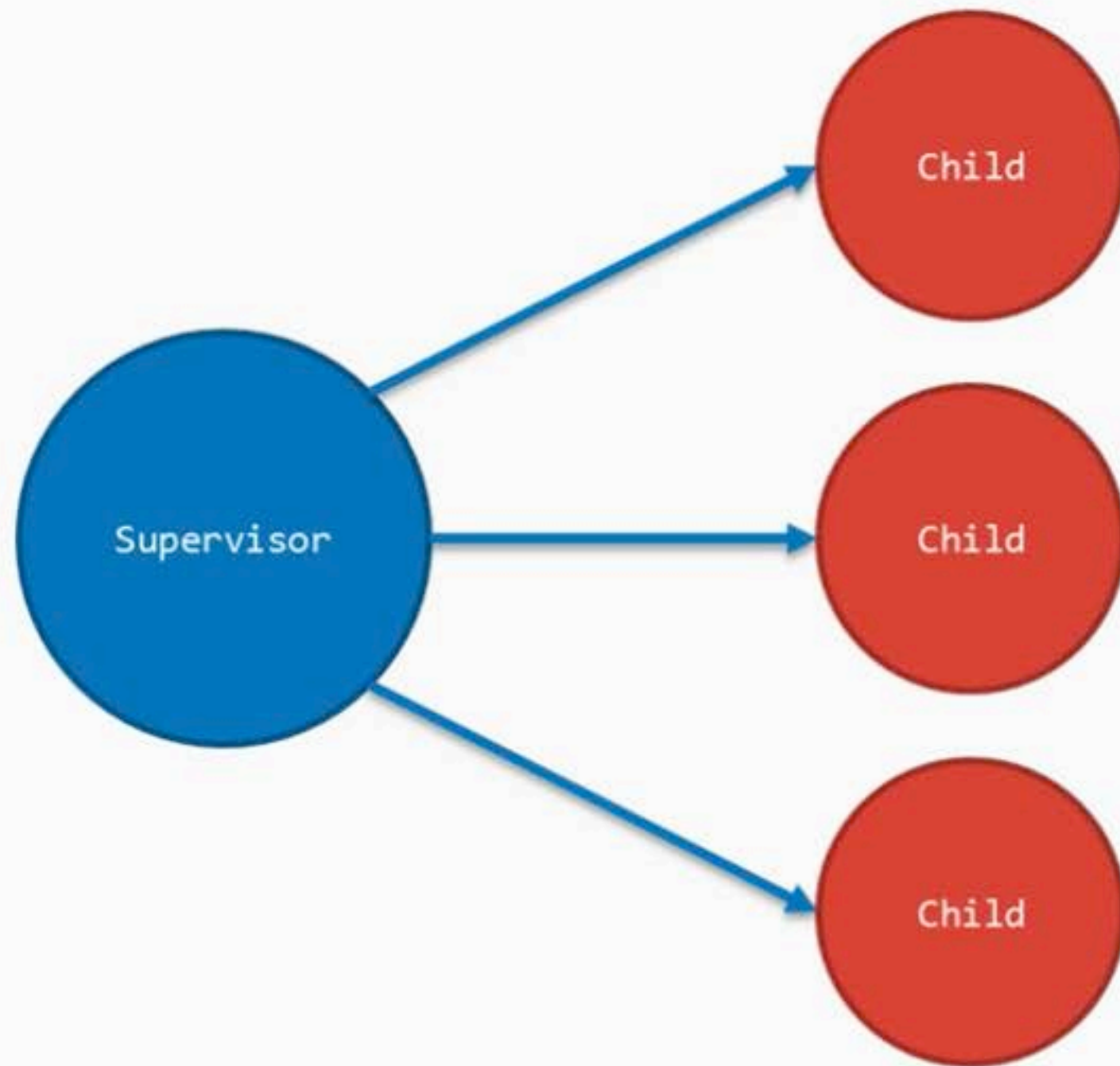


Recap – Application and Linked Processes

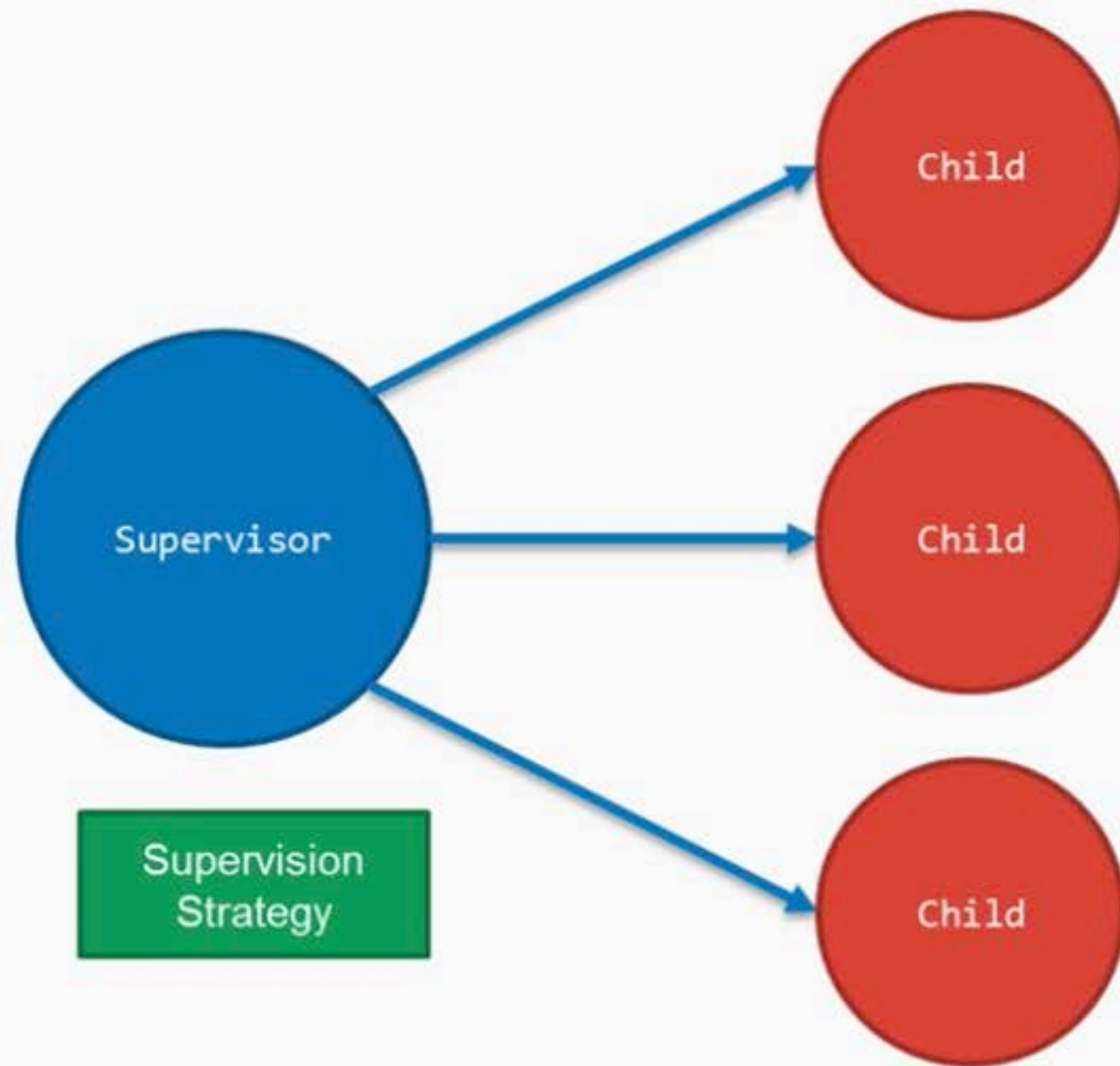


Supervisor

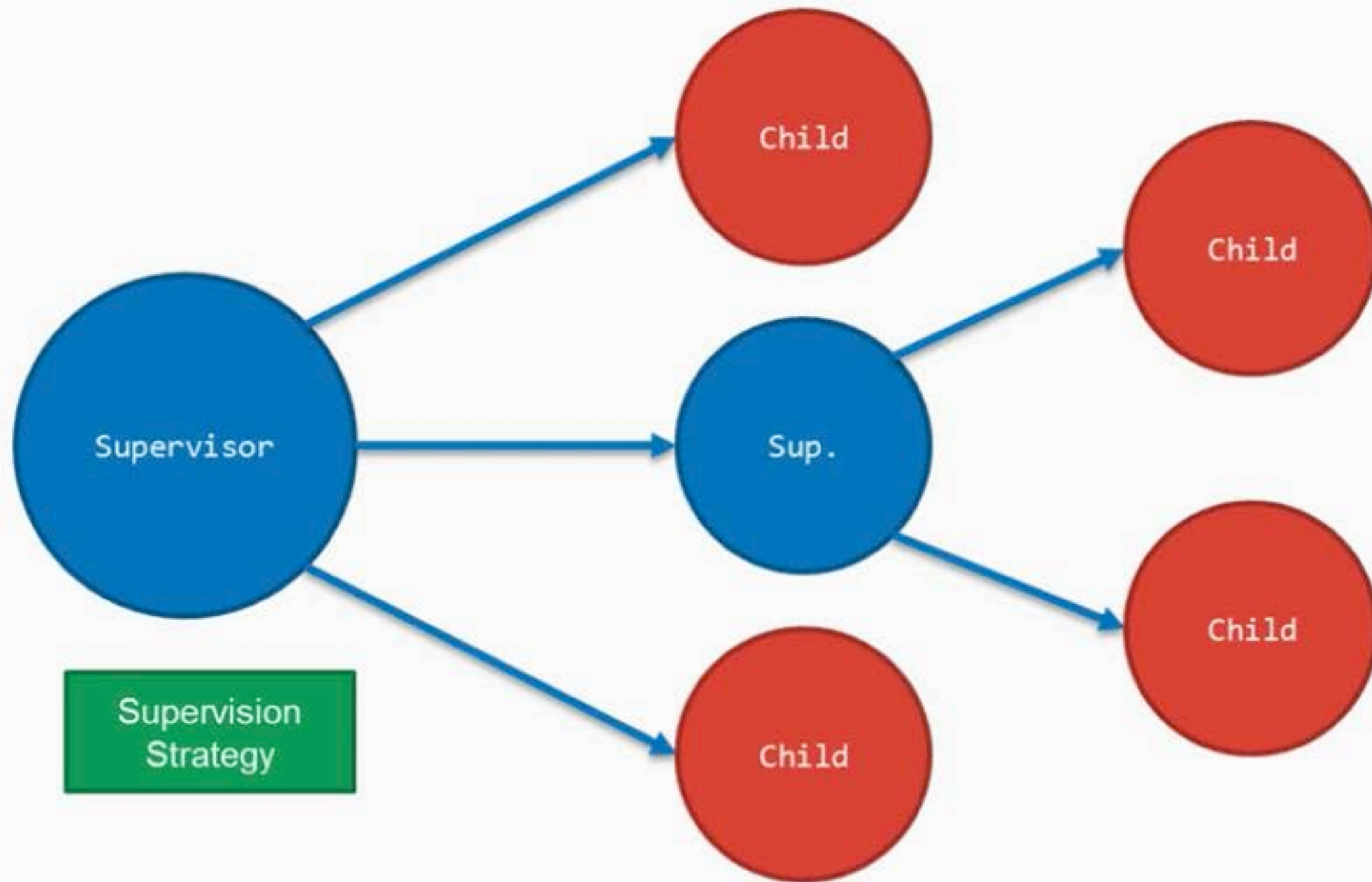
Supervisor



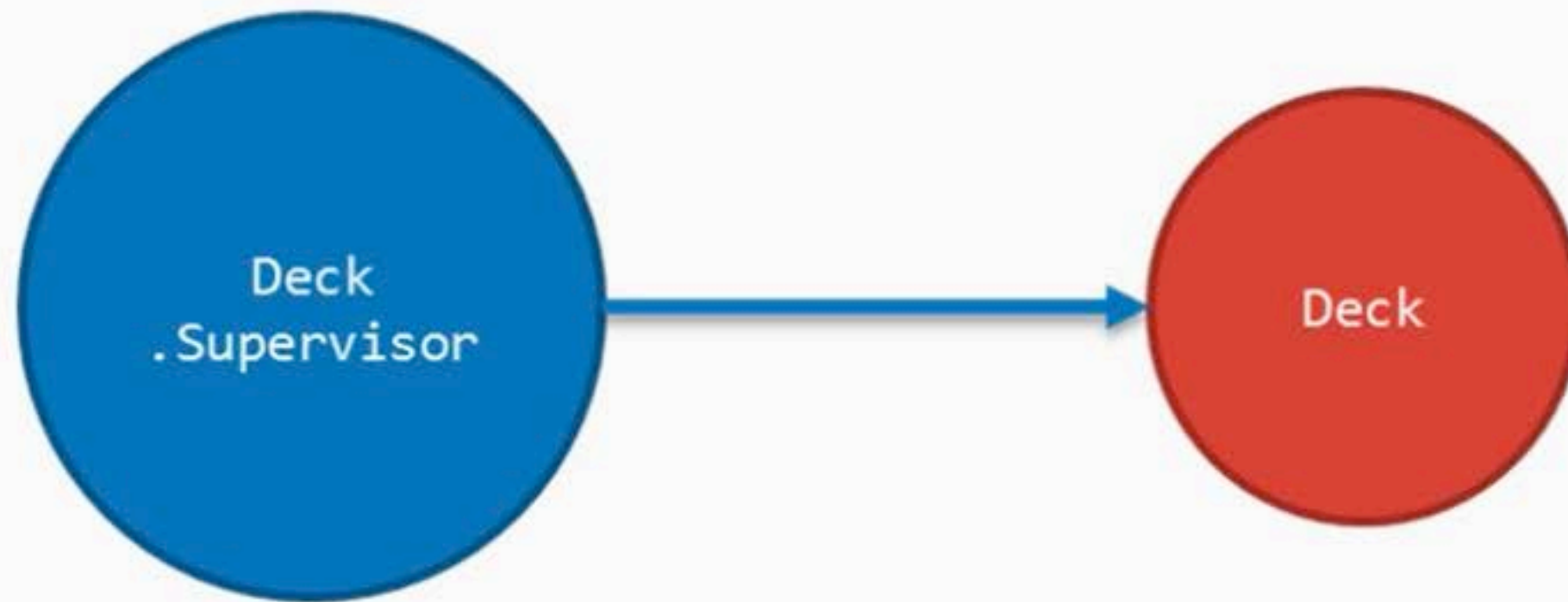
Supervisor



Supervisor



Supervisor



Press ? for neotree help

<r/Projects/Section6/card_deck/

+_build/

+config/

-lib/

application.ex

deck.ex

+test/

README.md

mix.exs

[3/6] card_deck (D:4 F:2)



- 0

scratch

Text

utf-8 | 1: 0

All

Filename: /Users/jpoverclock/Development/Elixir/Projects/Section6/card_deck/lib/

neotree-create-node (C-h: Go up one level)

/Users/jpoverclock/Development/Elixir/Projects/Section6/card_deck/lib/.

/Users/jpoverclock/Development/Elixir/Projects/Section6/card_deck/lib/..

application.ex

deck.ex

Press ? for neotree help

<r/Projects/Section6/card_deck/

+_build/

+config/

-lib/

application.ex

deck.ex

+test/

README.md

mix.exs

[3/6] card_deck (D:4 F:2)



- 0

scratch

Text

utf-8 | 1: 0

All

->_supervisor.ex

neotree-create-node (C-h: Go up one level)

[?] deck_supervisor.ex

HELM Neotree Create Node

1/1 (1 total)

C-c ? (help)

C-z (actions)

RET/F1/F2...

Press ? for neotree help

<r/Projects/Section6/card_deck/

+_build/

+config/

-lib/

application.ex

deck.ex

deck_supervisor.ex

+test/

README.md

mix.exs

[3/3] lib (F:3)

1

- 0 deck_supervisor.ex

Elixir

alchemist

Y

Wrote /Users/jpoverclock/Development/Elixir/Projects/Section6/card_deck/lib/deck_supervisor.ex

```
defmodule Deck.Supervisor do
  use Supervisor

  def start_link() do
    Supervisor.start_link(__MODULE__, [], name: __MODULE__)
  end

  def init(_) do
    children = [
      worker(Deck, [])
    ]

    Supervisor.init(children, strategy: :one_for_one)
  end
end
```

```
Press ? for neotree help
<r/Projects/Section6/card_deck/
+_build/
+config/
-lib/
  application.ex
  deck.ex
  deck_supervisor.ex
+test/
README.md
mix.exs
```

[1/3] lib (F:3)

```
defmodule Deck.Application do
  use Application

  def start(_type, _args) do
    Deck.start_link()
  end
end
```

```
Press ? for neotree help
<r/Projects/Section6/card_deck/
+_build/
+config/
-lib/
  application.ex
  deck.ex
  deck_supervisor.ex
+test/
README.md
mix.exs
```

[2/4] lib (F:4)

:w

```
defmodule Deck.Application do
  use Application

  def start(_type, _args) do
    Deck.Supervisor.start_link()
  end
end
```

① * 121 application.ex Elixir alchemist

➔ **card_deck** iex -S mix

Erlang/OTP 20 [erts-9.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:10]
] [hipe] [kernel-poll:false] [dtrace]

Compiling 3 files (.ex)

Generated card_deck app

Interactive Elixir (1.5.1) - press Ctrl+C to exit (type h() ENTER for help)

iex(1)> Deck.take_card

10

iex(2)> Supervisor.which_children(Deck.Supervisor)

[{Deck, #PID<0.155.0>, :worker, [Deck]}]

iex(3)> :█

```
➔ card_deck iex -S mix  
Erlang/OTP 20 [erts-9.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:10  
] [hipe] [kernel-poll:false] [dtrace]
```

```
Compiling 3 files (.ex)
```

```
Generated card_deck app
```

```
Interactive Elixir (1.5.1) - press Ctrl+C to exit (type h() ENTER for help)
```

```
iex(1)> Deck.take_card
```

```
10
```

```
iex(2)> Supervisor.which_children(Deck.Supervisor)
```

```
[{Deck, #PID<0.155.0>, :worker, [Deck]}]
```

```
iex(3)> :observer.start
```

```
█
```

card_deck
elixir
iex
kernel
logger
mix




```
➔ card_deck iex -S mix  
Erlang/OTP 20 [erts-9.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:10]  
] [hipe] [kernel-poll:false] [dtrace]
```

```
Compiling 3 files (.ex)  
Generated card_deck app  
Interactive Elixir (1.5.1) - press Ctrl+C to exit (type h() ENTER for help)  
iex(1)> Deck.take_card  
10  
iex(2)> Supervisor.which_children(Deck.Supervisor)  
[{Deck, #PID<0.155.0>, :worker, [Deck]}]  
iex(3)> :observer.start  
:ok  
iex(4)> 1..52 |> Enum.each(fn(_) -> IO.puts(Deck.take_card) end)
```

```
(stdlib) erl_eval.erl:404: :erl_eval.expr/5
(elixir) lib/enum.ex:681: anonymous fn/3 in Enum.each/2
(elixir) lib/enum.ex:1816: Enum.each/2
iex(4)>
16:42:27.015 [error] GenServer Deck terminating
** (FunctionClauseError) no function clause matching in Deck.handle_call/3
    (card_deck) lib/deck.ex:16: Deck.handle_call({:take_card}, {#PID<0.156.0>, #Reference<0.2636322940.2463891462.24876>}, [])
    (stdlib) gen_server.erl:636: :gen_server.try_handle_call/4
    (stdlib) gen_server.erl:665: :gen_server.handle_msg/6
    (stdlib) proc_lib.erl:247: :proc_lib.init_p_do_apply/3
Last message (from #PID<0.156.0>): {:take_card}
State: []
Client #PID<0.156.0> is alive
    (stdlib) gen.erl:169: :gen.do_call/4
    (elixir) lib/gen_server.ex:771: GenServer.call/3
    (stdlib) erl_eval.erl:670: :erl_eval.do_apply/6
    (stdlib) erl_eval.erl:878: :erl_eval.expr_list/6
    (stdlib) erl_eval.erl:404: :erl_eval.expr/5
    (elixir) lib/enum.ex:681: anonymous fn/3 in Enum.each/2
    (elixir) lib/enum.ex:1816: Enum.each/2
    (stdlib) erl_eval.erl:670: :erl_eval.do_apply/6
```




```
(stdlib) gen_server.erl:636: :gen_server.try_handle_call/4
(stdlib) gen_server.erl:665: :gen_server.handle_msg/6
(stdlib) proc_lib.erl:247: :proc_lib.init_p_do_apply/3
Last message (from #PID<0.156.0>): {:take_card}
State: []
Client #PID<0.156.0> is alive
(stdlib) gen.erl:169: :gen.do_call/4
(elixir) lib/gen_server.ex:771: GenServer.call/3
(stdlib) erl_eval.erl:670: :erl_eval.do_apply/6
(stdlib) erl_eval.erl:878: :erl_eval.expr_list/6
(stdlib) erl_eval.erl:404: :erl_eval.expr/5
(elixir) lib/enum.ex:681: anonymous fn/3 in Enum.each/2
(elixir) lib/enum.ex:1816: Enum.each/2
(stdlib) erl_eval.erl:670: :erl_eval.do_apply/6
```

nil

```
iex(5)> Deck.take_card
```

37

```
iex(6)> Deck.take_card
```

39

```
iex(7)> Deck.take_card
```

3

```
iex(8)> █
```