Solid ground

@sasajuric

aircloak.com

OTP

metaprogramming

ExUnit

Distillery

Plug

Credo

community

pattern matching

Phoenix

GenStage

mix

Ecto

Cowboy

the pipe operator

functional programming

Nerves

Poolboy

Combine

Saša Jurić @sasajuric · 23 Aug 2015

Coincidentally, I recently said that "the most important aspect of Elixir is in fact Erlang VM" infoq.com/articles/elixi…

1 10 9

José Valim

@josevalim

Following

@sasajuric Precisely! If there was no Erlang VM, there would be no Elixir. Elixir is just a small drop. ;)

magic source

BEAM

magic beneficiaries

Erlang Elixir LFE

OTP GenStage

Cowboy Poolboy

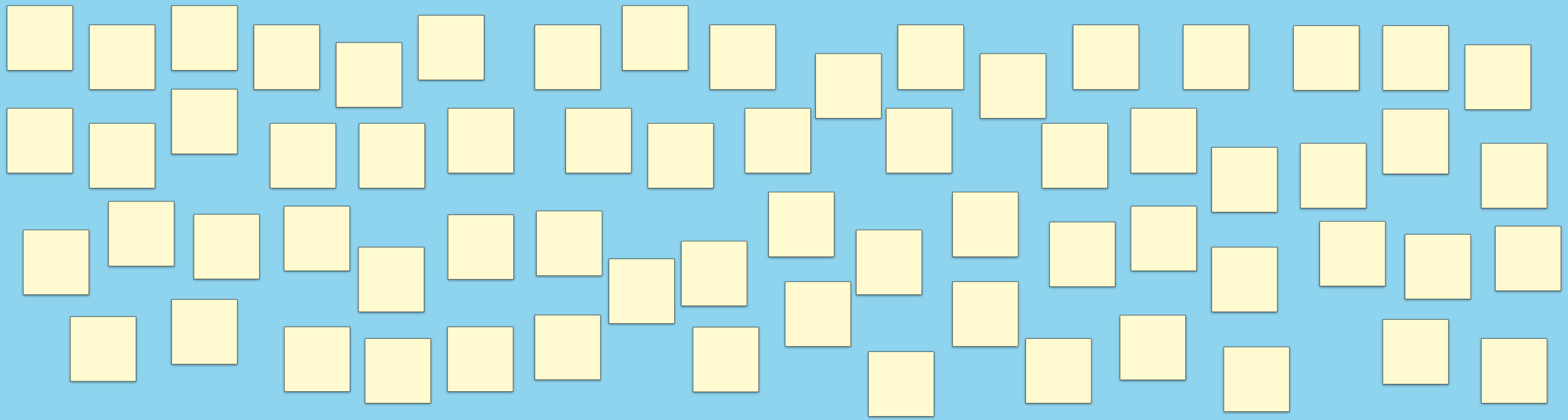
Plug Ecto

Phoenix

long-running system
many tasks
soft real-time

finite time program
all-or-nothing
hard real-time

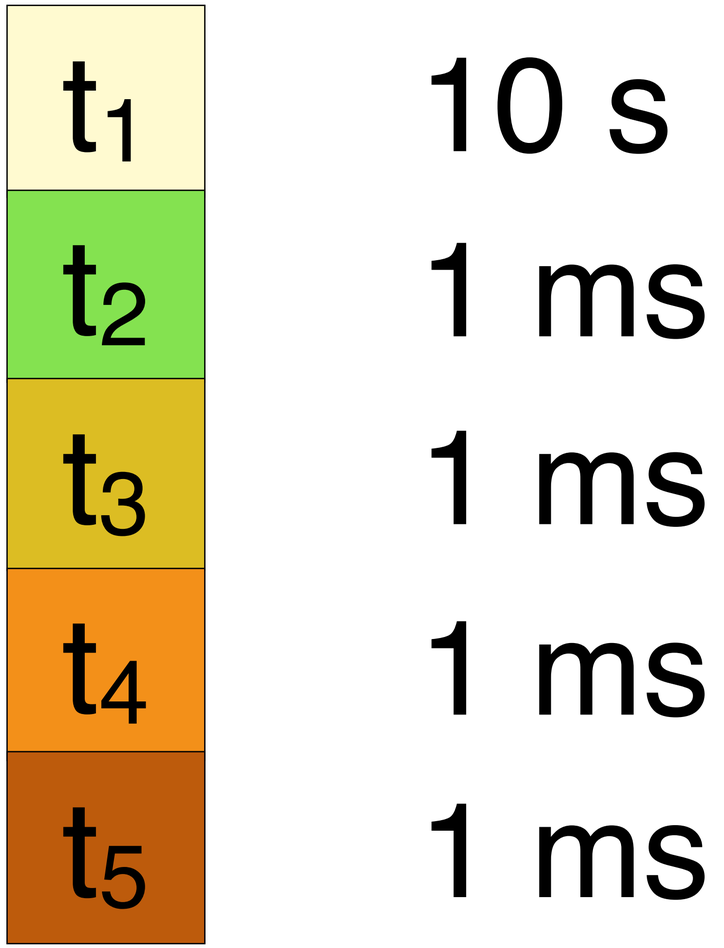
BEAM



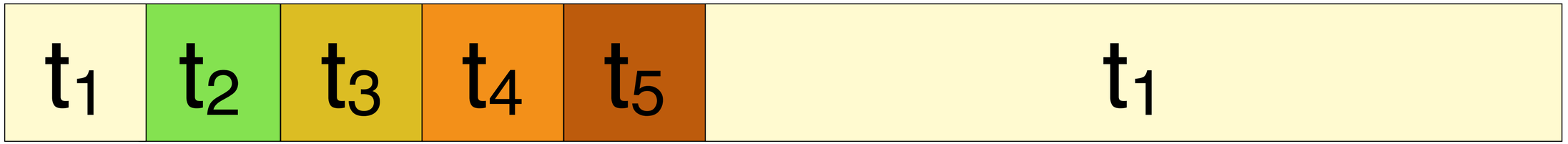
scheduler scheduler scheduler scheduler

CPU CPU CPU CPU

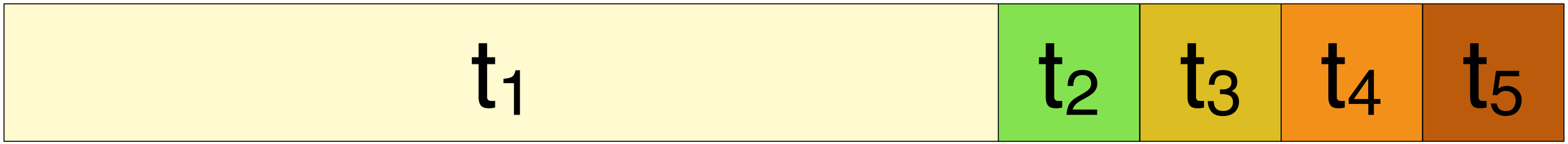
frequent context switches


t_1
t_2
t_3
t_4
t_5
10 s
1 ms
1 ms
1 ms
1 ms

BEAM scheduler



cooperative scheduler



cooperative scheduler

long running task

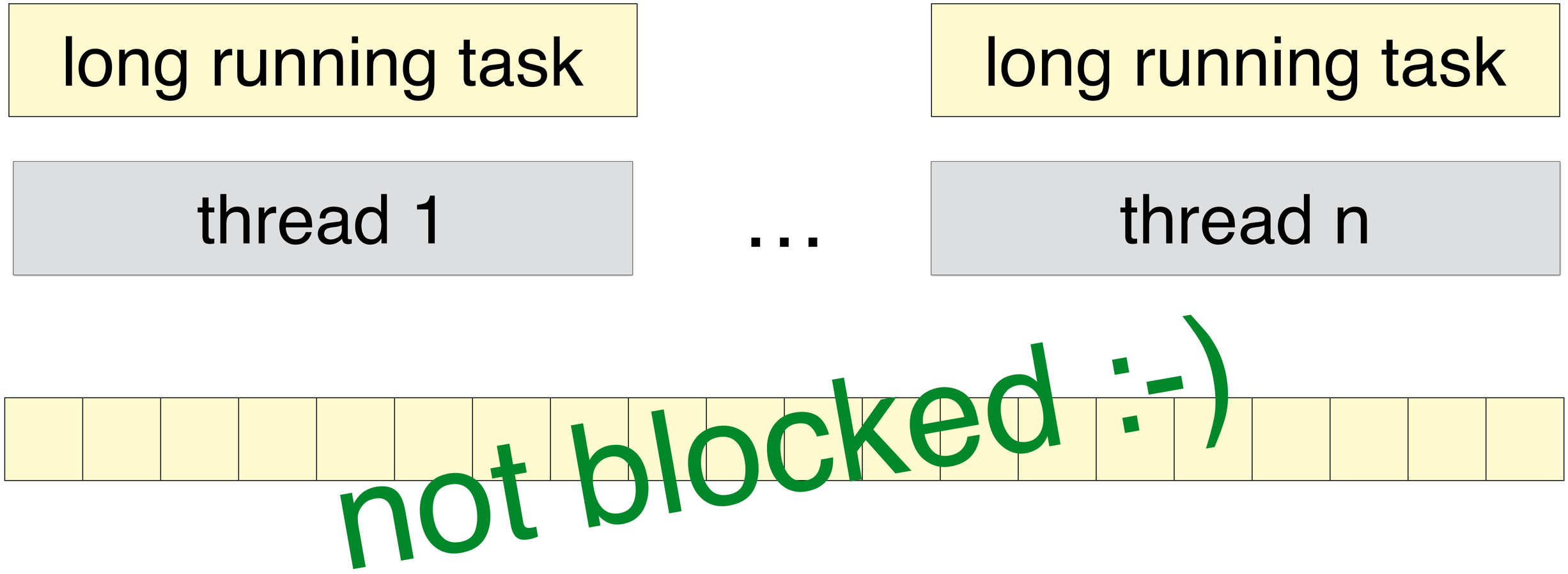
long running task

thread 1

…

thread n

BEAM scheduler



activities as runtime citizens

```
send(pid, :please_stop)
```

```elixir
mref = Process.monitor(pid)
send(pid, :please_stop)
receive do
  {:DOWN, ^mref, :process, ^pid, _reason} ->
    :ok
end
```

```elixir
mref = Process.monitor(pid)
send(pid, :please_stop)
receive do
  {:DOWN, ^mref, :process, ^pid, _reason} -> :ok
after :timer.seconds(5) ->
  Process.exit(pid, :kill)
  receive do
    {:DOWN, ^mref, :process, ^pid, _reason} -> :ok
  end
end
```

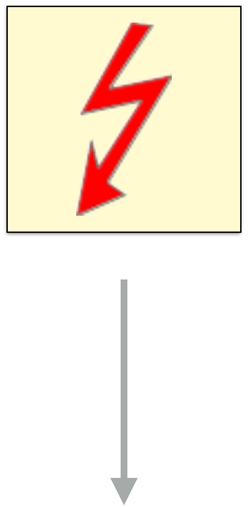
```elixir
task = Task.async(fn -> ... end)

case Task.yield(task, :timer.seconds(5)) do
  {:ok, result} -> do_something(result)
  nil -> Task.shutdown(task, :brutal_kill)
end
```

shared-nothing concurrency

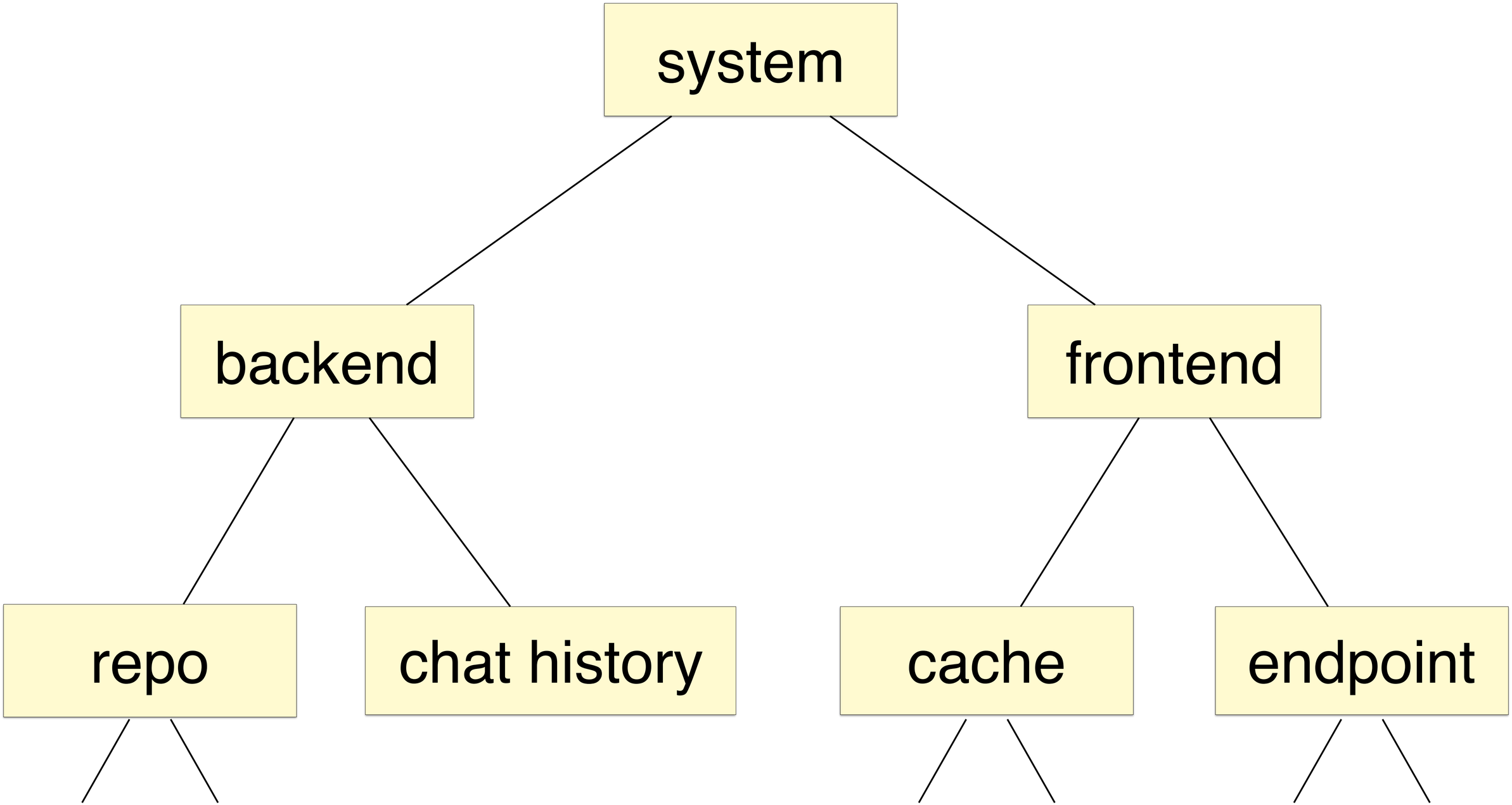
Process

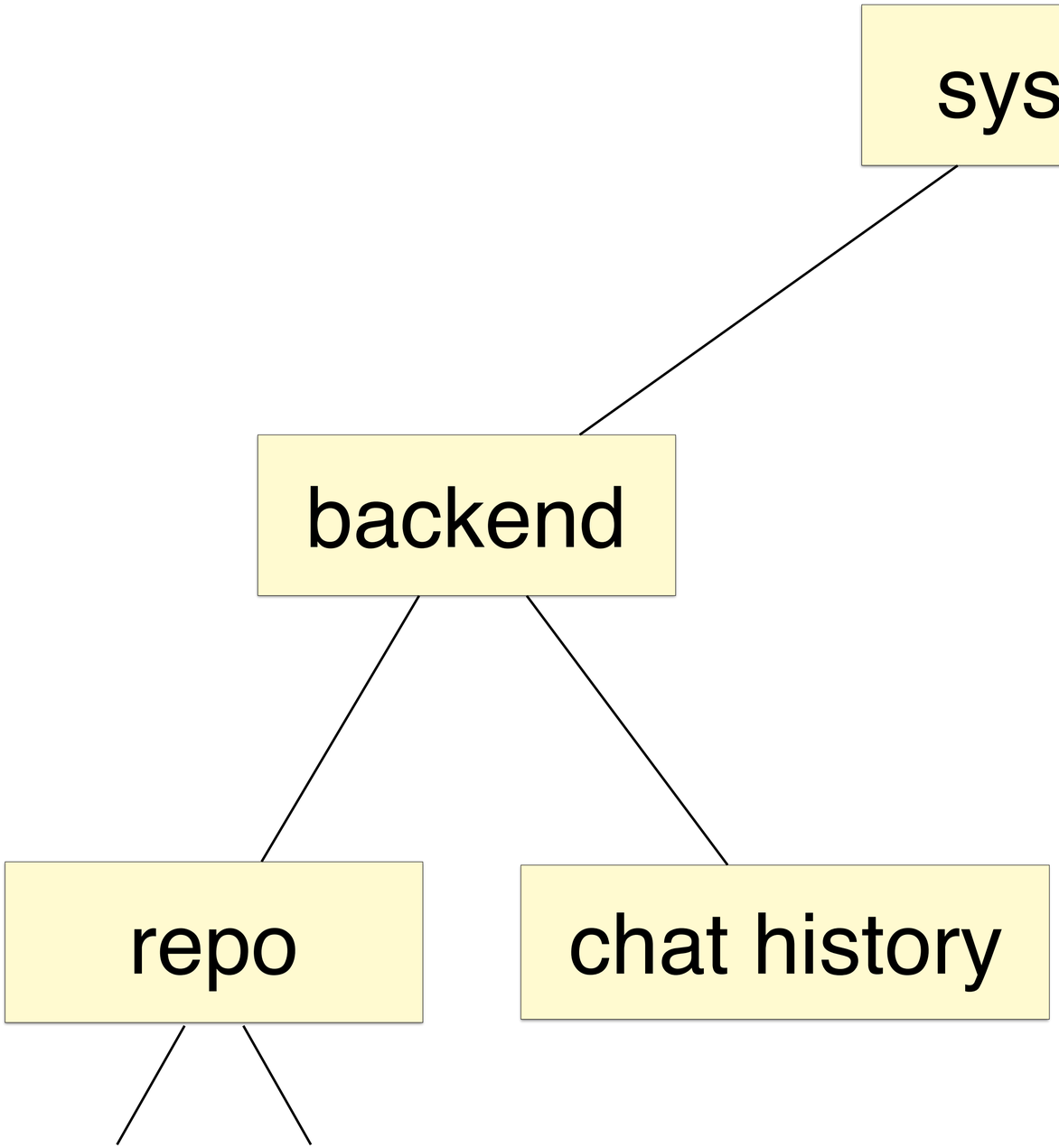
heap
resources



process-owned “stuff” is released

supervision tree

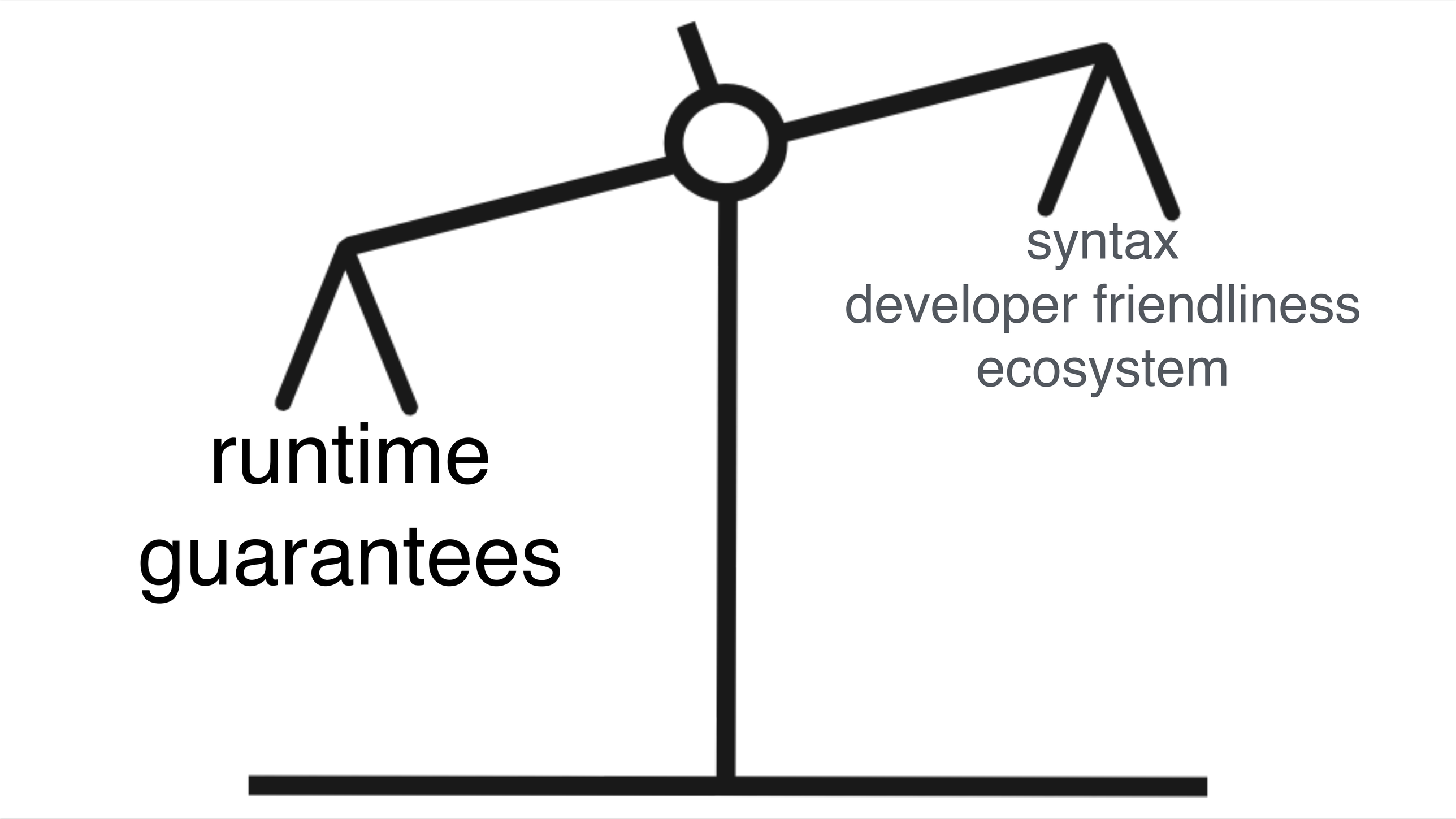

system
backend
frontend
repo
chat history
cache
endpoint


system
backend
repo
chat history

supervisor foundations

frequent context switching
activities as runtime citizens
observable process termination
stoppable processes
shared-nothing concurrency

syntax
developer friendliness
ecosystem


runtime
guarantees
syntax
developer friendliness
ecosystem

How Erlang does scheduling	http://jlouisramblings.blogspot.hr/2013/01/how-erlang-does-scheduling.html
Erlang scheduler details	https://hamidreza-s.github.io/erlang/scheduling/real-time/preemptive/migration/2016/02/09/erlang-scheduler-details.html
The BEAM book	https://github.com/happi/theBeamBook
BEAM wisdoms	http://beam-wisdoms.clau.se/en/latest/
Web based observer	https://github.com/shinyscorpion/wobserver
Recon	http://ferd.github.io/recon/
Erlang performance lab	http://www.erlang.pl/

Scales of Justice image taken from clipartfox.com