

SOFTWARE DEVELOPMENT CONFERENCE

# YOW! LONDON 2022

## Concurrency Oriented Programming In a Modern World.

**Francesco Cesarini**

Founder & Technical Director



@FrancescoC

**Robert Virding**

Principal Language Expert



@rvirding

# Concurrency Oriented Programming

I WAS DOING  
CONCURRENCY  
THE RIGHT WAY  
BEFORE IT WAS A  
POPULAR HACKY  
KLUDGE ON EVERY  
OTHER LANGUAGE.



Joe Armstrong

# Concurrency Oriented Programming

- The world is concurrent
- Things in the world don't share data
- Things communicate with messages
- Things fail

I WAS DOING  
CONCURRENCY  
THE RIGHT WAY  
BEFORE IT WAS A  
POPULAR HACKY  
KLUDGE ON EVERY  
OTHER LANGUAGE.



Joe Armstrong's Tenets

**Let's create a Programming Language  
which builds on these beliefs!**

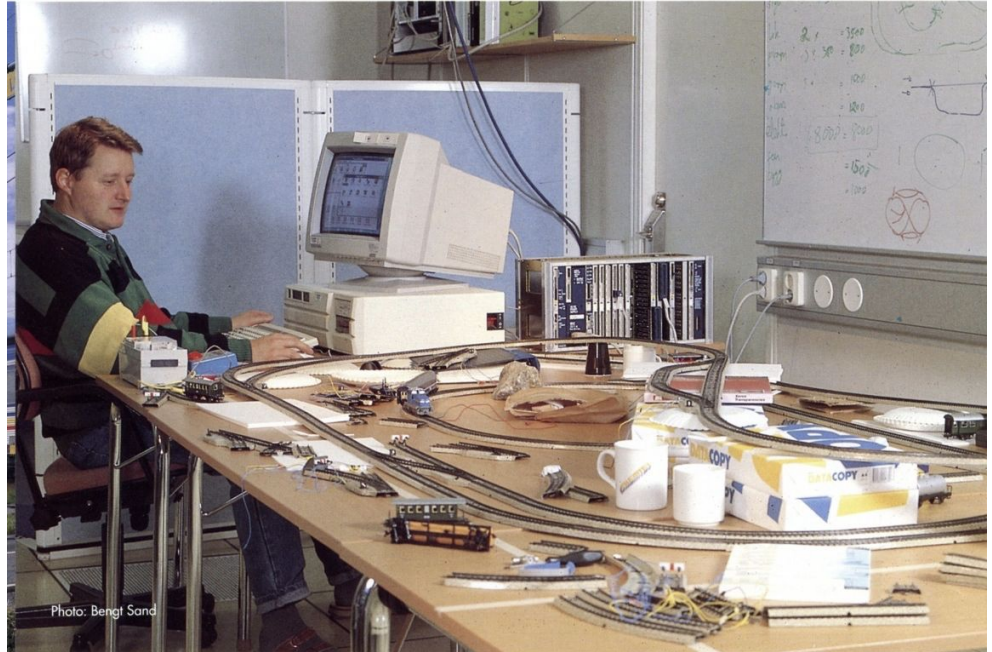


# Hard at work developing Erlang

Robert Virding preparing the demo of a Märklin toy railway for the IT show.

They demonstrated a working toy railway with 2 trains on the same track, ATC and balises.

(October 1993)



# The Main Ideas



pid1

```
pid1 = spawn(Mod, func, args)
```

# The Main Ideas



pid1

```
defmodule Foo do
  def print_add(a, b), do: IO.puts(a + b)
end

pid1 = spawn(Foo, :print_add, [1, 2])
```

# The Main Ideas



```
defmodule Foo do
  def print_add(a, b), do: IO.puts(a + b)
end

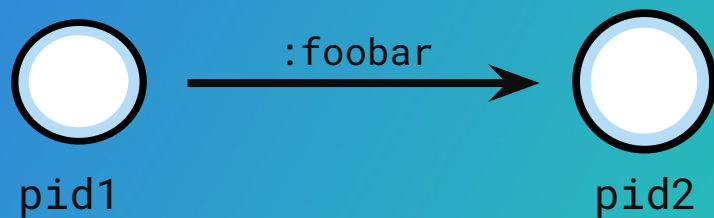
pid1 = spawn(Foo, :print_add, [1, 2])
```

```
defmodule Foo do
  def kaboom(a, b), do: a/b
end

pid1 = spawn(Foo, :kaboom, [1, 0])
```



# The Main Ideas



```
send(pid2, :foobar)
```

# The Main Ideas



:foobar  
:start

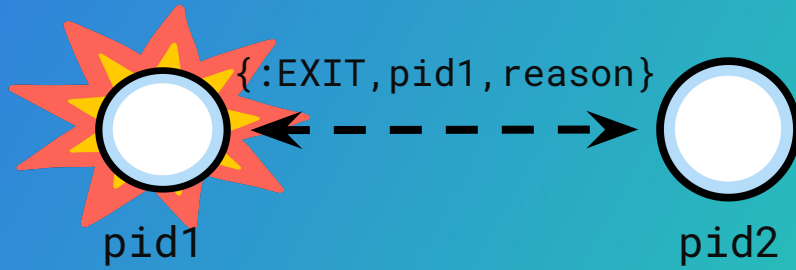
```
receive do
  :start    -> start_it()
  :stop     -> stop_it()
  {pid, :boot} -> boot(pid)
end
```

# The Main Ideas



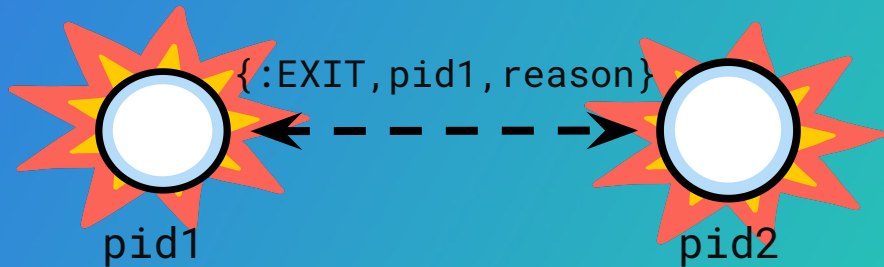
`Process.link(pid2)`

# The Main Ideas



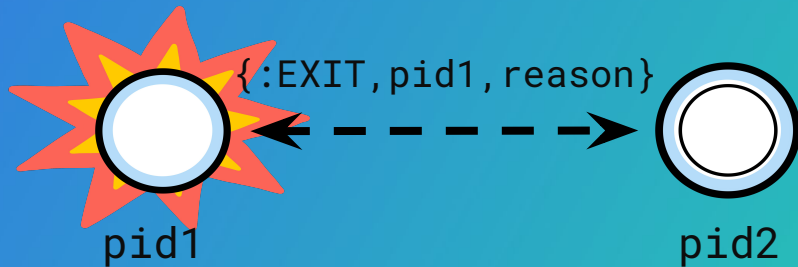
`Process.link(pid2)`

# The Main Ideas



`Process.link(pid2)`

# The Main Ideas

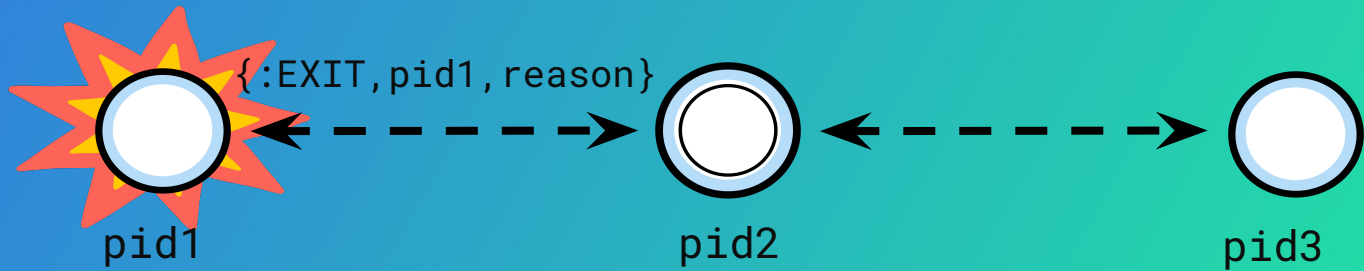


```
Process.flag(:trap_exit, true)
```



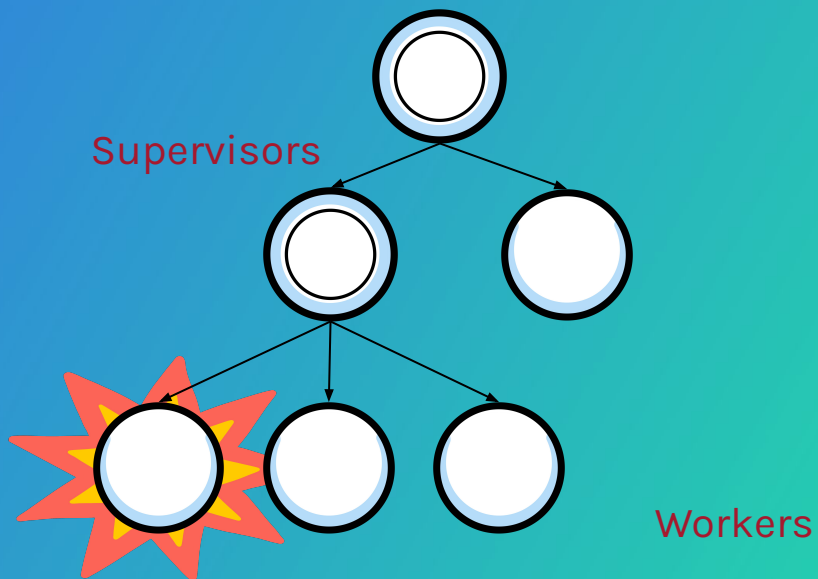
`{:EXIT, pid1, reason}`

# The Main Ideas



```
Process.flag(:trap_exit, true)
```

# The Main Ideas







“OOP to me means only messaging,  
local retention and protection and  
hiding of state-process, and extreme  
LateBinding of all things.”

Alan Kay

**Why is this  
relevant today?**

# #TalkConcurrency Interviews & Panel

WITH SIR TONY HOARE, JOE ARMSTRONG, CARL HEWITT

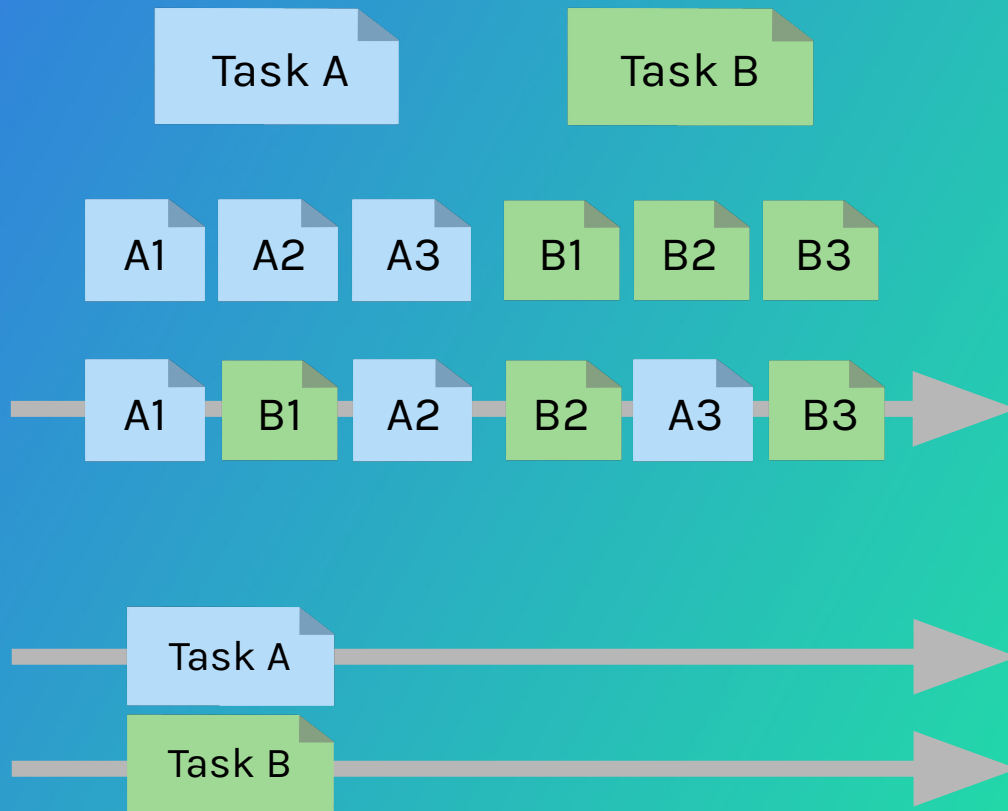


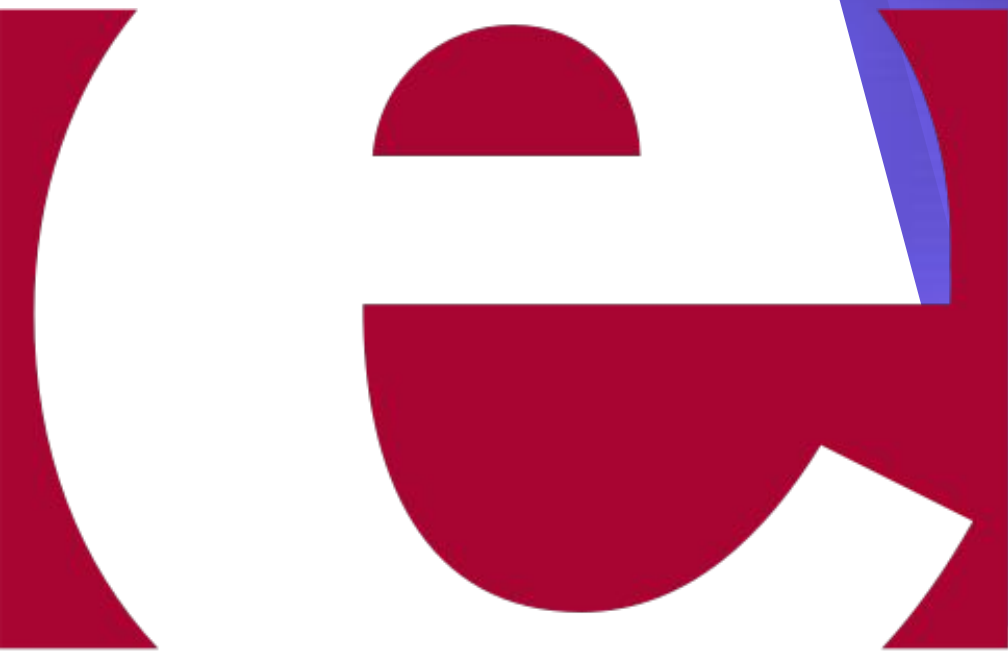


“Concurrency is about dealing with lots of things at once. Parallelism is about doing lots of things at once”

Rob Pike

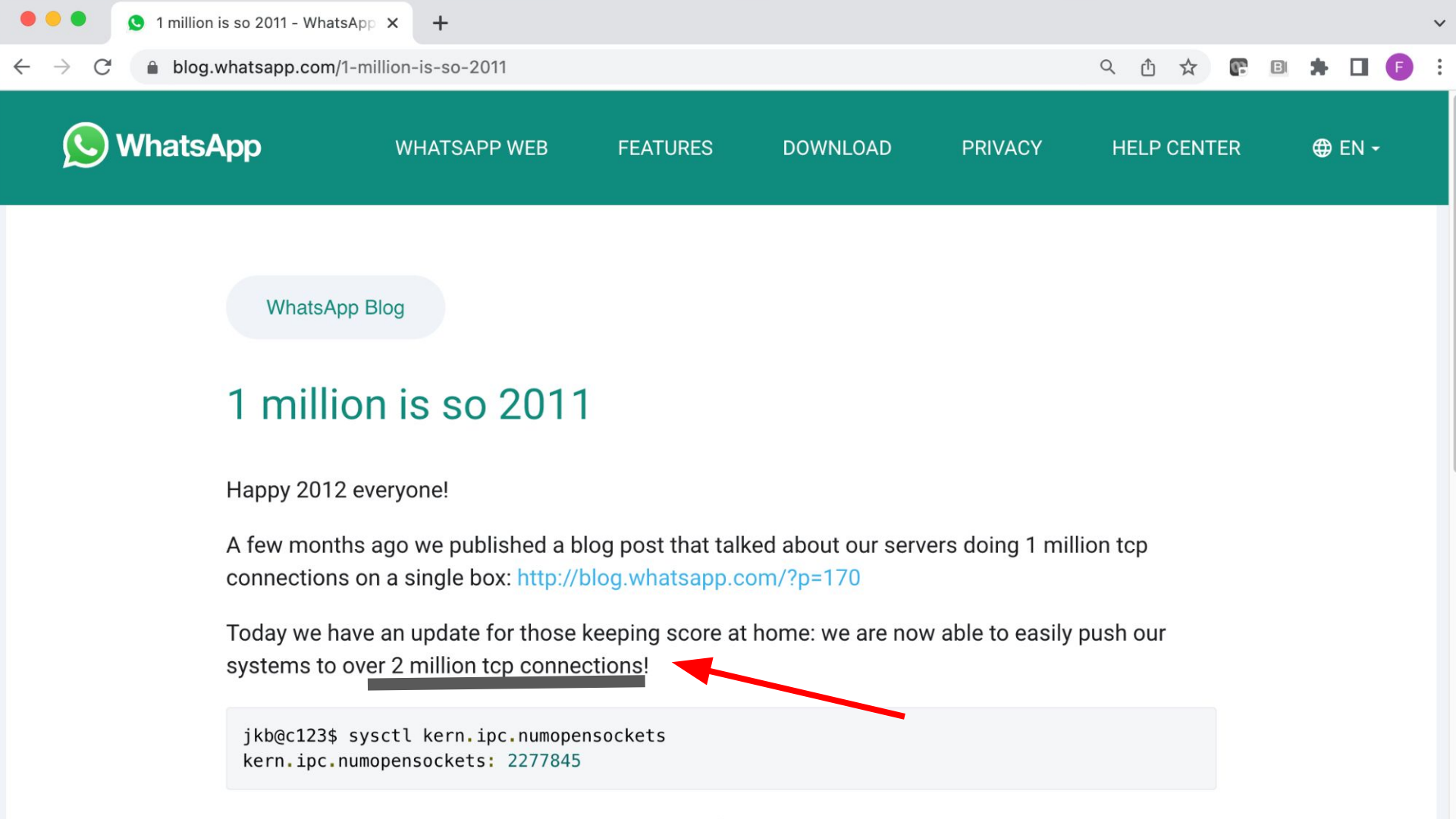
# Concurrency vs Parallelism





**ERLANG**





## WhatsApp Blog

# 1 million is so 2011

Happy 2012 everyone!

A few months ago we published a blog post that talked about our servers doing 1 million tcp connections on a single box: <http://blog.whatsapp.com/?p=170>

Today we have an update for those keeping score at home: we are now able to easily push our systems to over 2 million tcp connections!

```
jkb@c123$ sysctl kern.ipc.numopensockets  
kern.ipc.numopensockets: 2277845
```



# The Road to 2 Million Websocket Connections in Phoenix

Posted on November 3rd, 2015 by Gary Rennie

Search

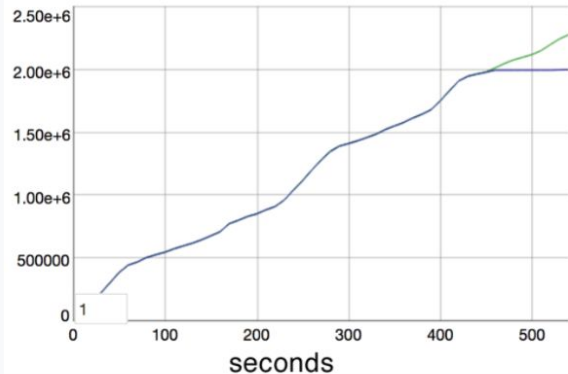
Search for...

Search

Tags

[CI](#)  
[channels](#)  
[elixirconf](#)  
[generators](#)  
[guides](#)  
[liveview](#)  
[phoenix](#)  
[pubsub](#)  
[releases](#)  
[testing](#)  
[uploads](#)

Simultaneous Users



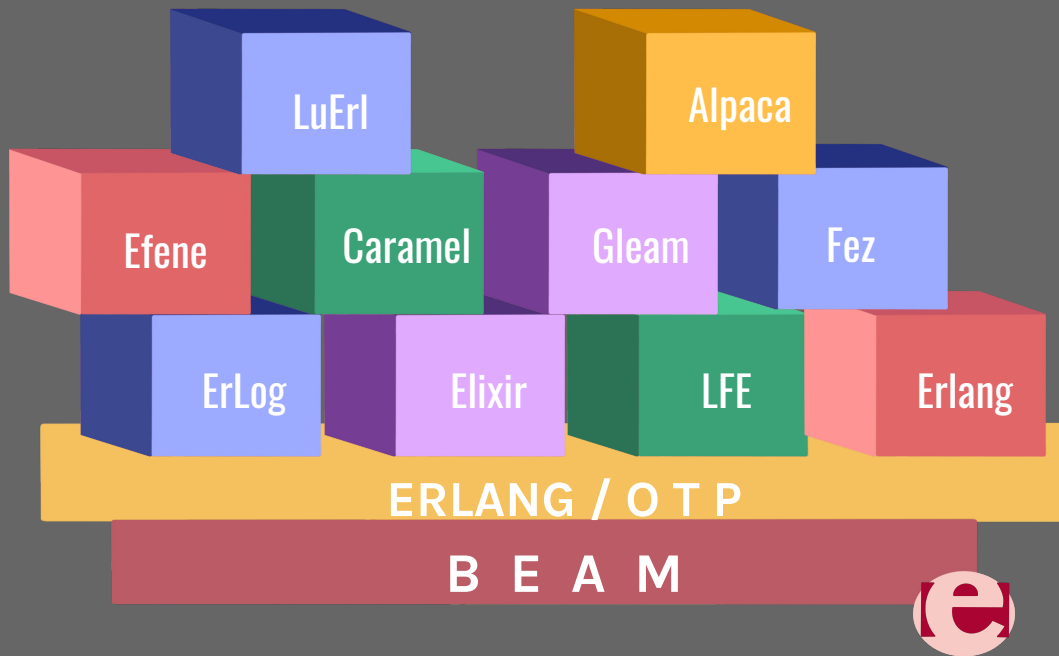
1700045  
1763630  
1999975  
1999984

subscribers

```
1 [ 0.0% ] 11 [ 0.5% ] 21 [ 0.0% ] 31 [ 0.0% ]
2 [ 0.0% ] 12 [ 0.5% ] 22 [ 0.0% ] 32 [ 0.0% ]
3 [ 0.0% ] 13 [ 0.0% ] 23 [ 0.0% ] 33 [ 0.0% ]
4 [ 1.0% ] 14 [ 0.0% ] 24 [ 0.5% ] 34 [ 0.0% ]
5 [ 0.5% ] 15 [ 0.0% ] 25 [ 0.0% ] 35 [ 0.0% ]
6 [ 0.5% ] 16 [ 0.0% ] 26 [ 0.0% ] 36 [ 0.0% ]
7 [ 0.0% ] 17 [ 0.0% ] 27 [ 0.0% ] 37 [ 0.0% ]
8 [ 1.0% ] 18 [ 0.0% ] 28 [ 0.5% ] 38 [ 0.0% ]
9 [ 0.0% ] 19 [ 0.0% ] 29 [ 0.0% ] 39 [ 0.0% ]
10 [ 0.0% ] 20 [ 0.0% ] 30 [ 0.0% ] 40 [ 0.0% ]
Mem [|||||||] 83765/128906MB Tasks: 22, 150 thr; 2 running
Swp [ 0/0MB ] Load average: 5.98 5.45 3.98
Uptime: 5 days, 11:17:13
```



# A Language Becomes an Ecosystem



# Elixir

## Functional Languages

ML, Miranda

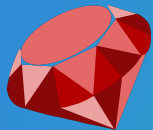
## Concurrent Languages

SmallTalk, Ada,  
Modula, Chill

## Logical Languages

Prolog

E R L A N G



R U B Y



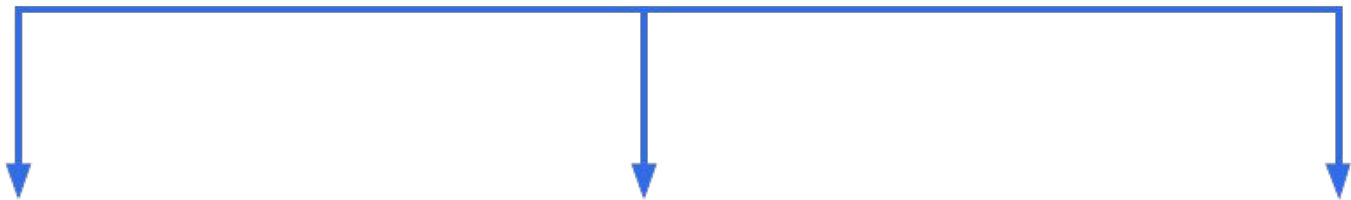
E L I X I R



C L O J U R E



**kubernetes**



Google Cloud Platform



## Virding's First Rule of Programming

“Any sufficiently complicated concurrent program in another language contains an ad hoc informally-specified bug-ridden slow implementation of half of Erlang.

- Robert Virding



# Questions?

Concurrency Oriented Programming in a Modern World.

**Francesco Cesarini**

Founder & Technical Director



@FrancescoC

**Robert Virding**

Principal Language Expert



@rvirding