CQRS with Erlang

presented by

Bryan Hunter

Firefly Logic

Twitter

#erlang #cqrs

#strangeloop

@bryan_hunter

@fireflylogic

?

What and why CQRS?

CQRS is
Command **Q**uery
Responsibility
Segregation.

CQRS is
an architectural
pattern.

Command Query Separation (CQS) applied to the “big”

```
void MutateSomeState(…);

- or -

MyResult GetWithoutTouching(…);
```

CQRS

is about

separating reads from writes

CQRS

is about

heresy

> What is CQRS?

Why CQRS?

Domain Driven Design

Your heart says

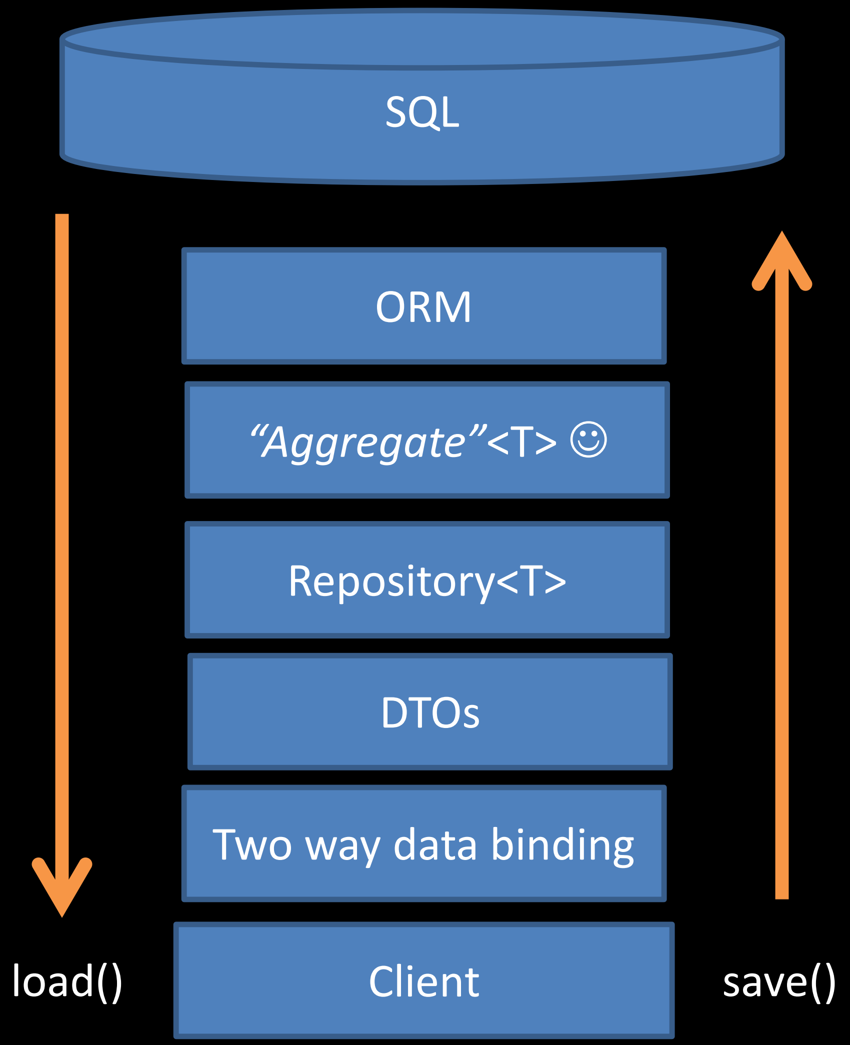
"it's the right thing"

…but your brain says

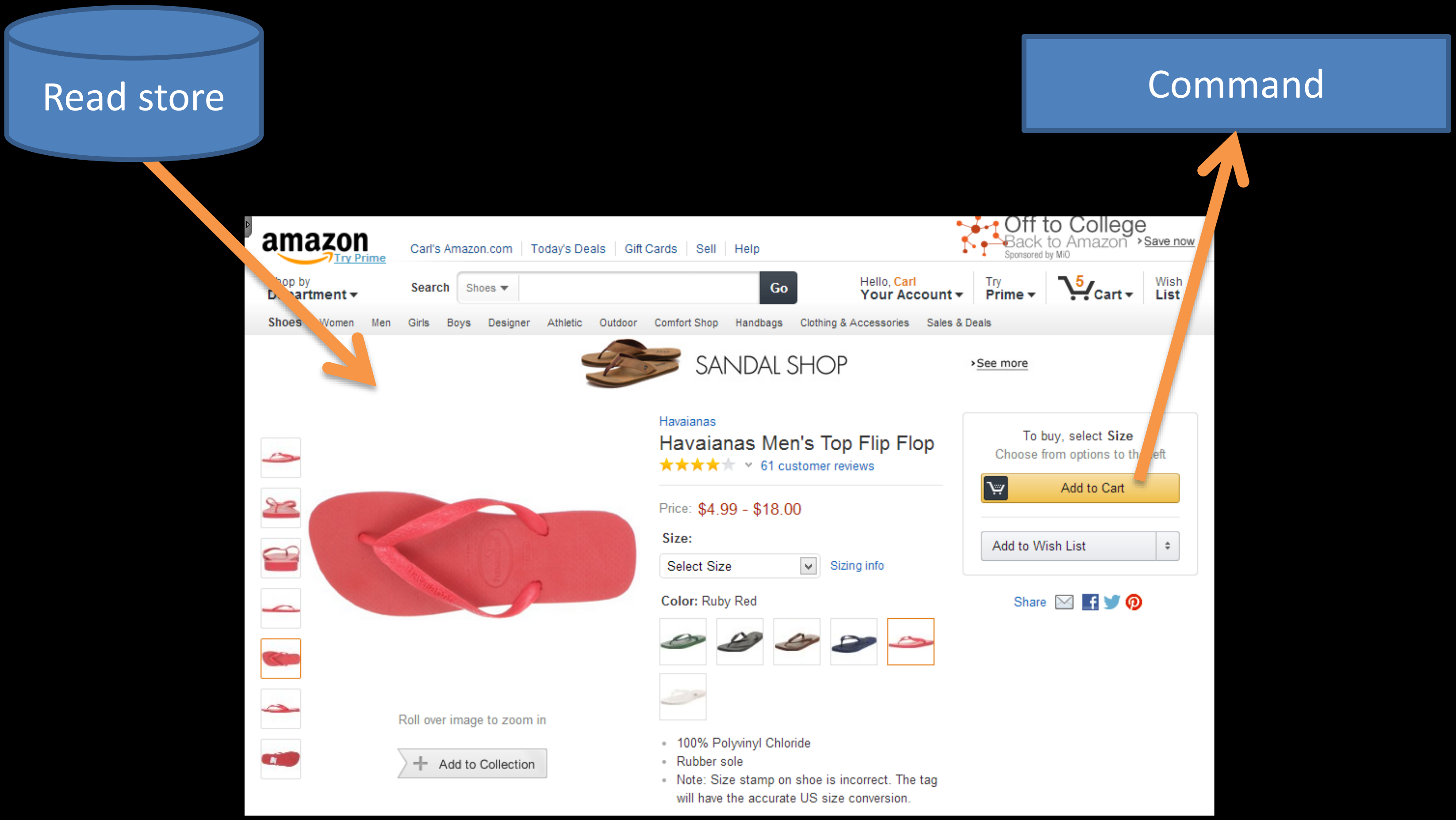
"you're a big stinkin' fraud"

Bog standard Layered Architecture

(not CQRS)



Task-based UI



Read store

Command

amazon
Try Prime

Carl's Amazon.com | Today's Deals | Gift Cards | Sell | Help

Off to College
Back to Amazon ›Save now
Sponsored by MiO

Shop by Department

Search Shoes Go

Hello, Carl
Your Account

Try Prime

5 Cart

Wish List

Shoes Women Men Girls Boys Designer Athletic Outdoor Comfort Shop Handbags Clothing & Accessories Sales & Deals

SANDAL SHOP ›See more

Havaianas

Havaianas Men's Top Flip Flop

★★★★☆ 61 customer reviews

Price: $4.99 - $18.00

Size:

Select Size Sizing info

Color: Ruby Red

- 100% Polyvinyl Chloride
- Rubber sole
- Note: Size stamp on shoe is incorrect. The tag will have the accurate US size conversion.

Roll over image to zoom in

Add to Collection

To buy, select Size
Choose from options to the left

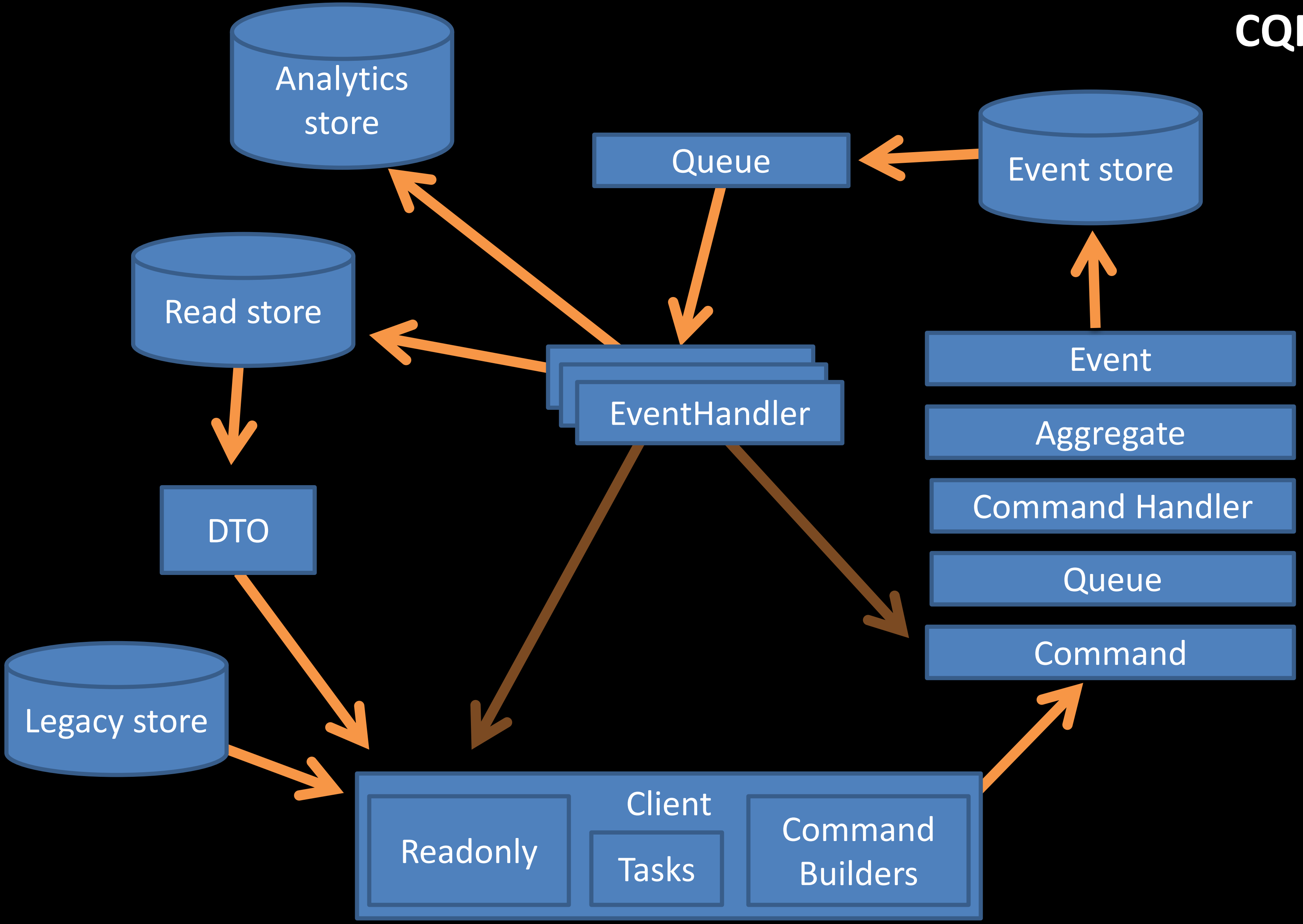
Add to Cart

Add to Wish List

Share

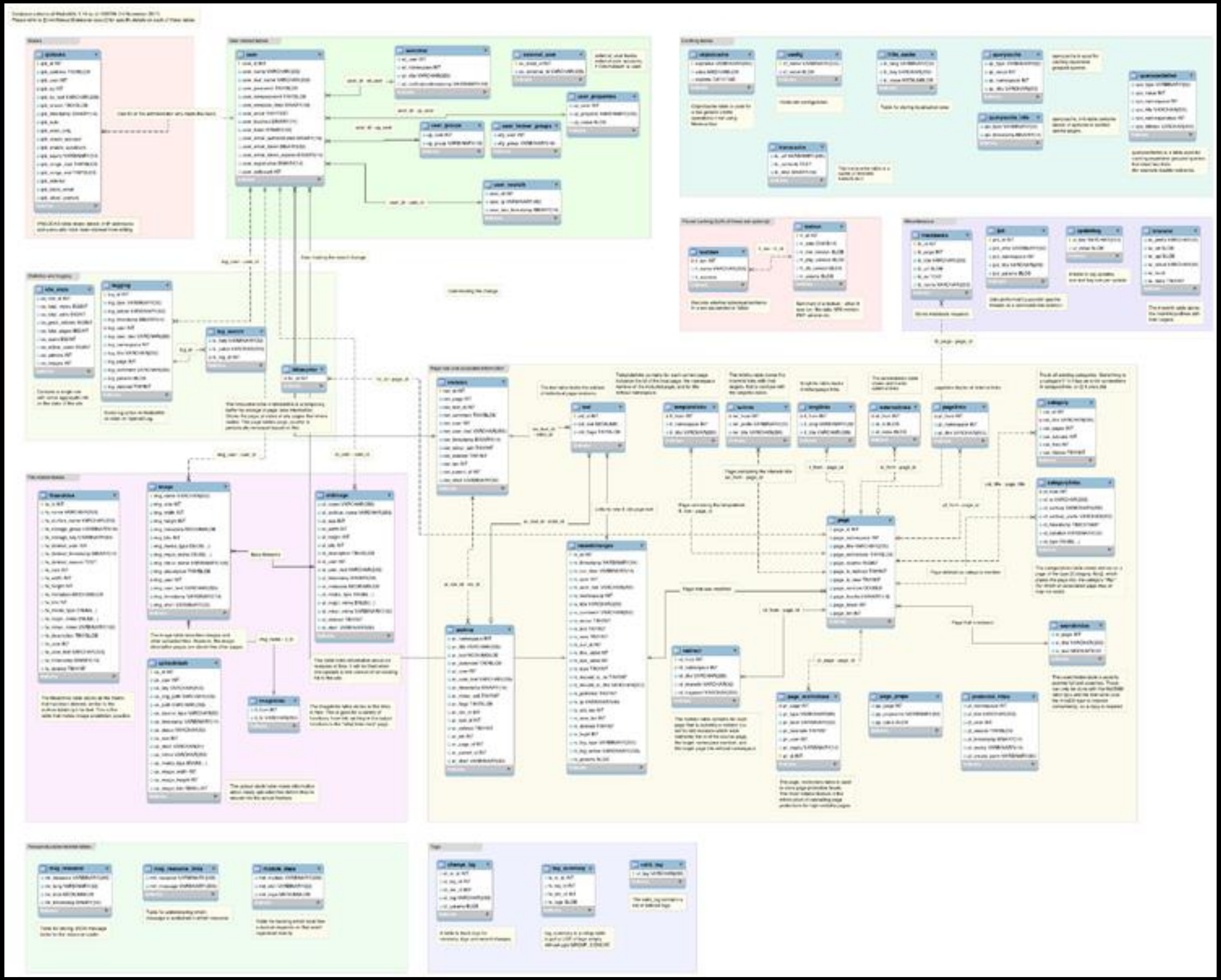
> How CQRS?

CQRS

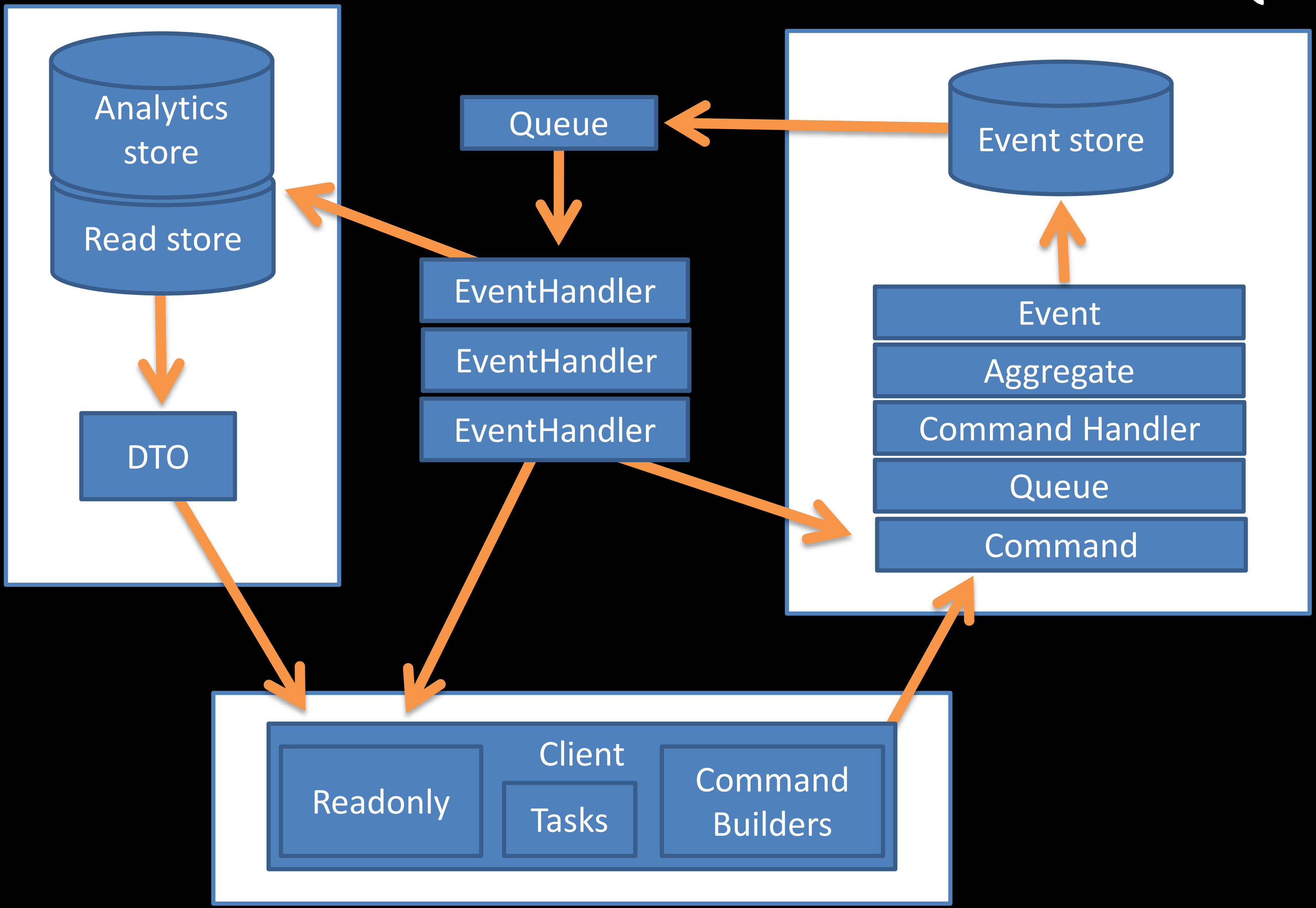


> How CQRS

keep your privates private



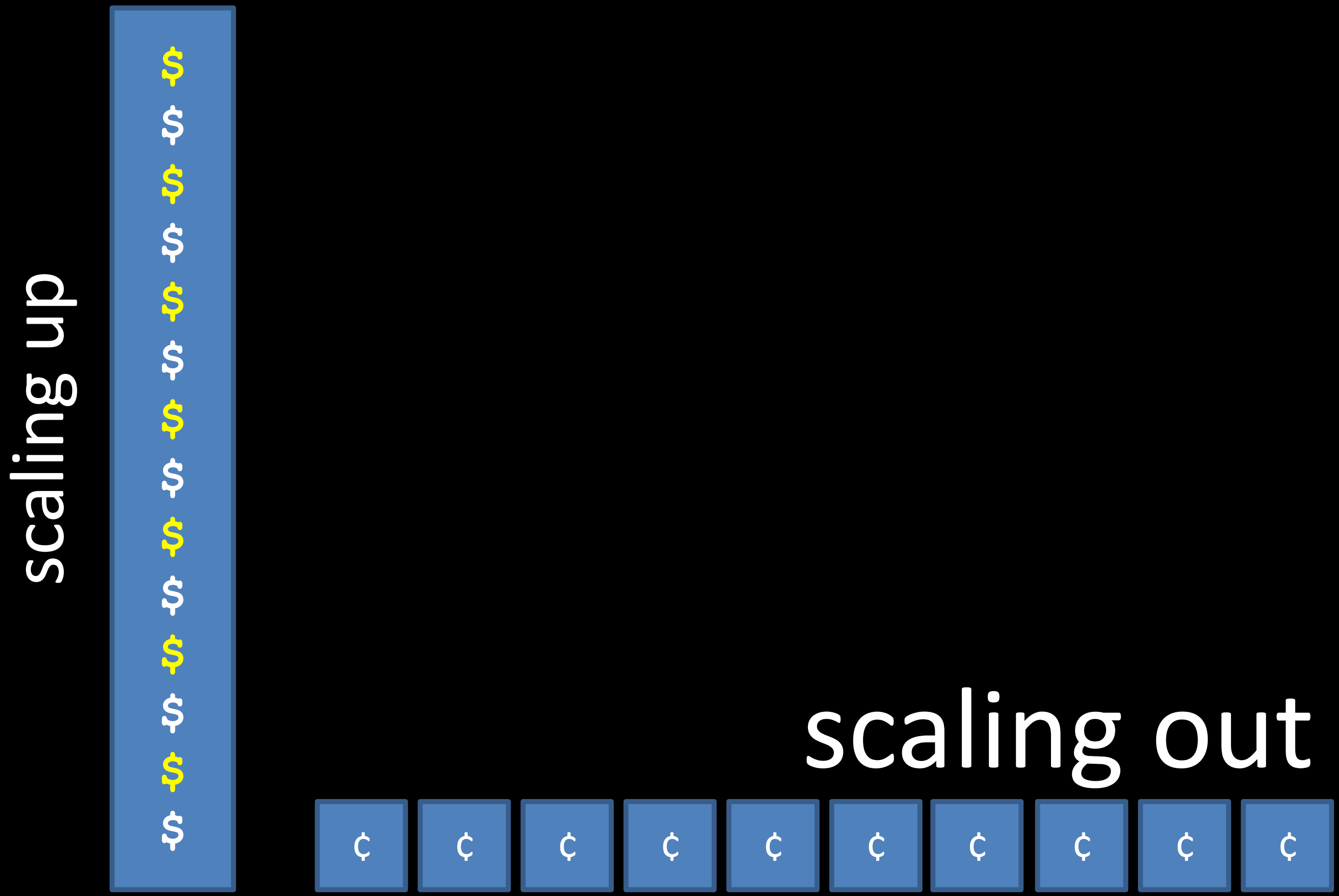
CQRS



> Why CQRS

returns focus to the business

Vendor relationships should be ***consensual***



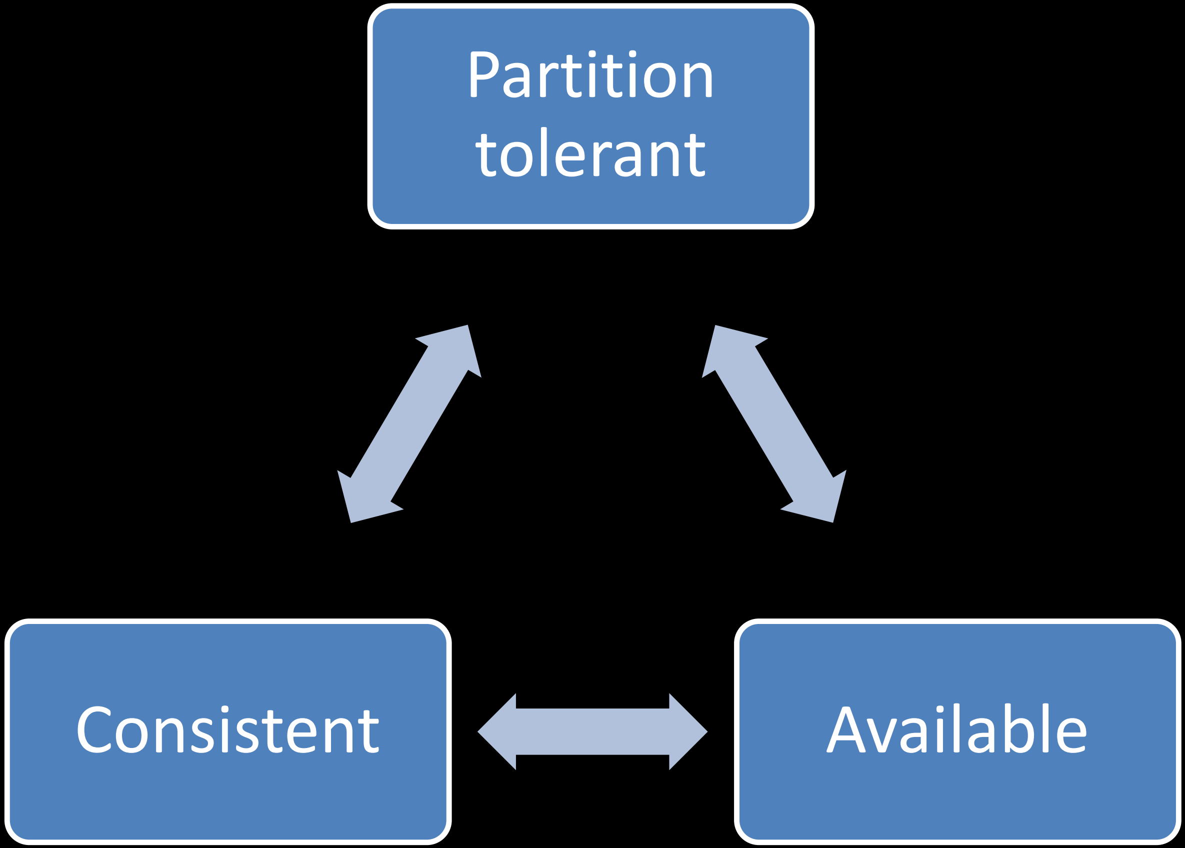
scaling up

scaling out

> Why CQRS?

explicit & tunable staleness

Let the business tweak CAP

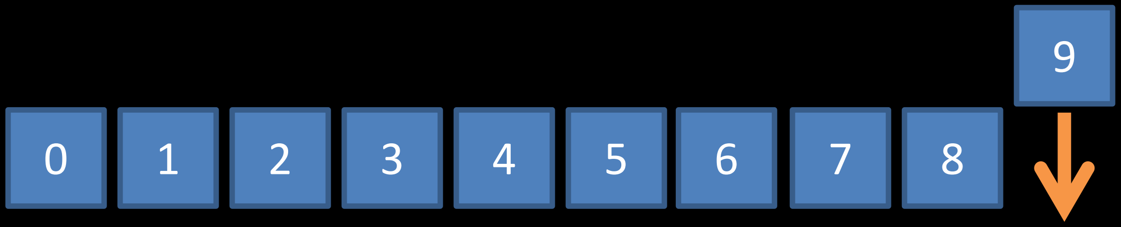


Partition tolerant

Consistent

Available

simplify conflicts, read-write contention, and concurrency



> Why CQRS?

> Why CQRS?

http://memegenerator.net/Twin-Peaks-Log-Lady

OUR ONE CHANCE TO ~~TO~~ GET IT RIGHT

Big Design Upfront

YAGNI

Just-in-Time

Future-proof

Maintenance

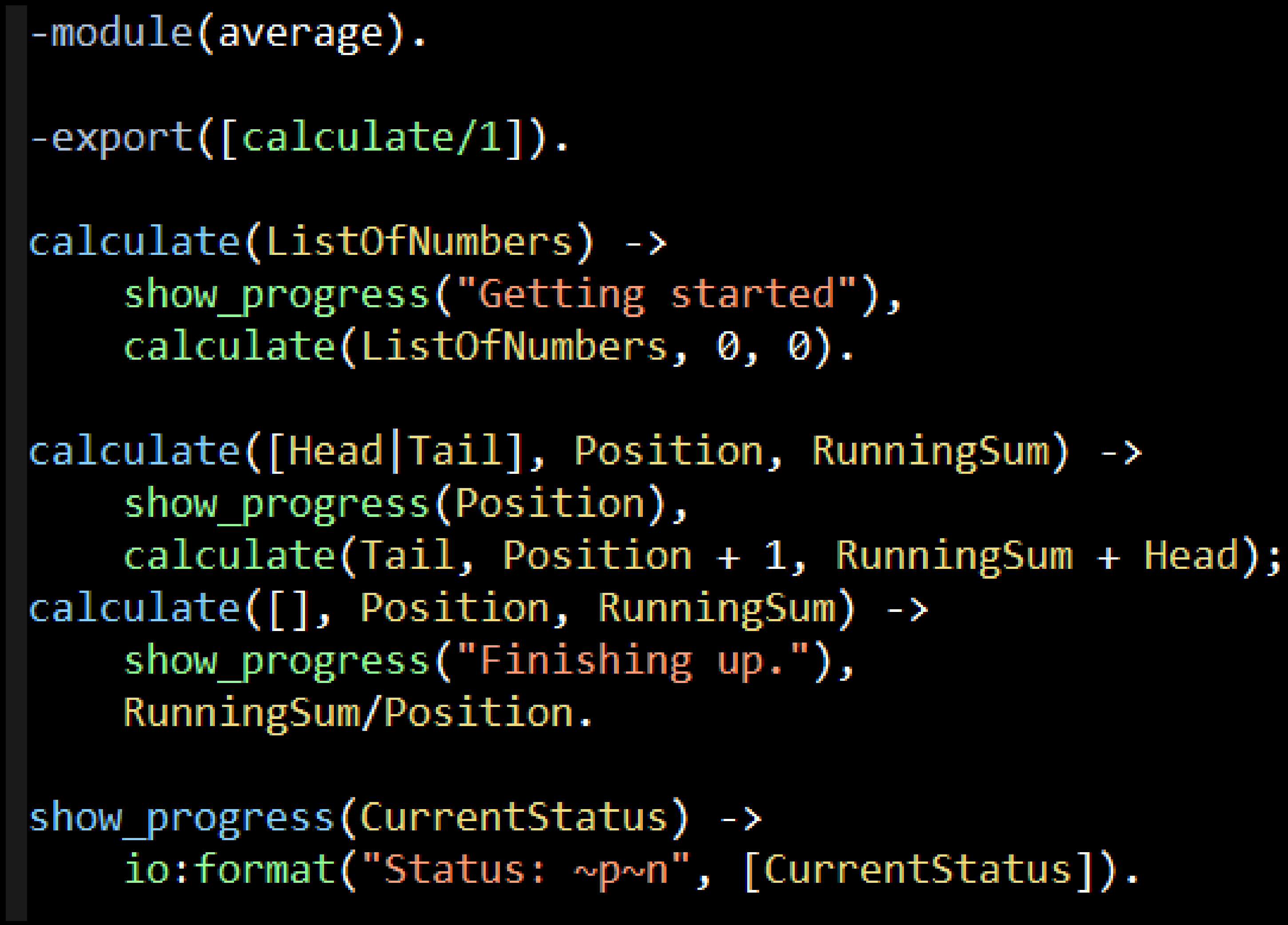
> Why CQRS?

Why Erlang?

Erlang is **open**, **proven** and **cross-platform**. It simplifies writing **reliable**, **concurrent,** **distributed** systems. Its **pattern matching** is A-OK.

> Why Erlang?

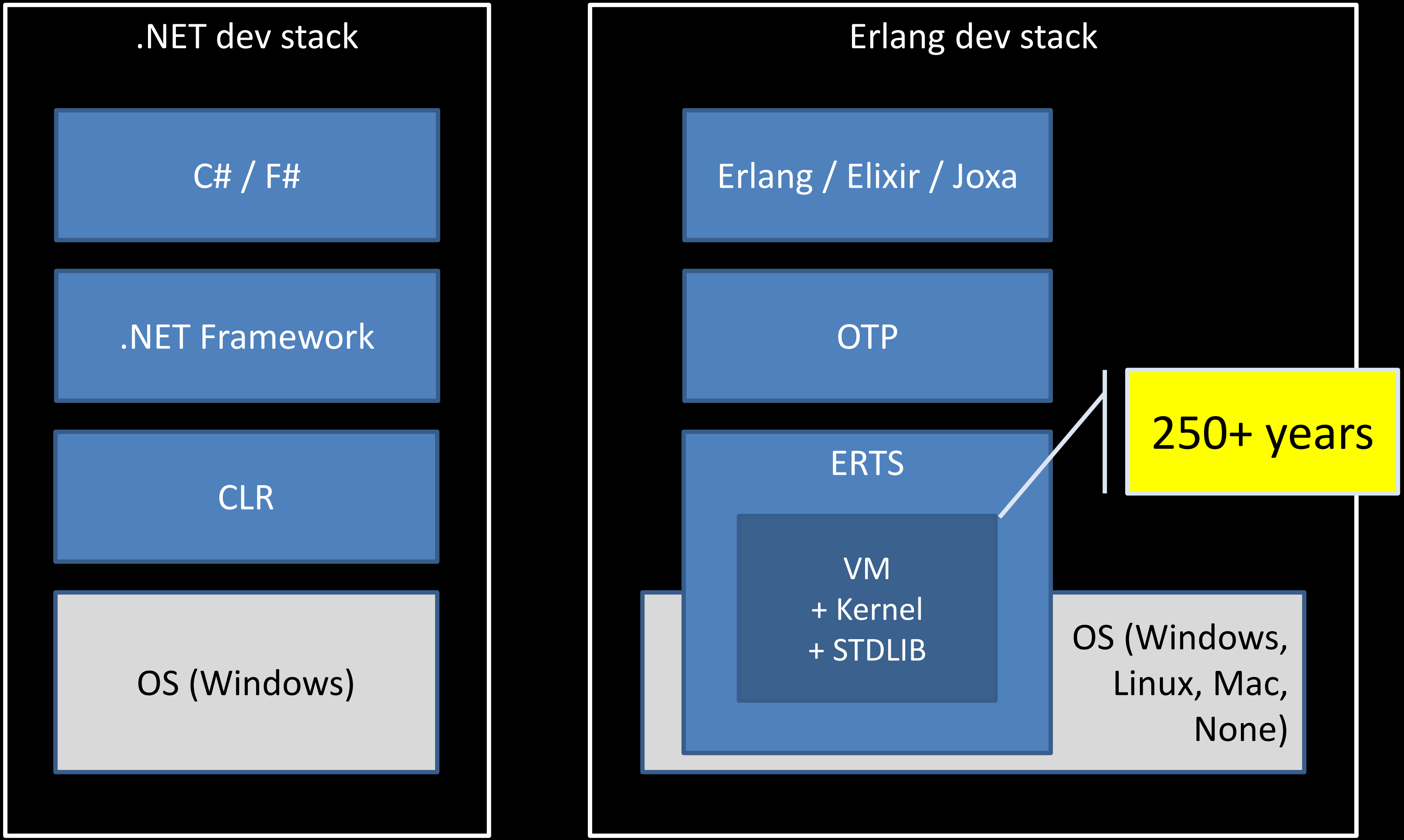
Erlang (the Language)



```erlang
-module(average).

-export([calculate/1]).

calculate(ListOfNumbers) ->
    show_progress("Getting started"),
    calculate(ListOfNumbers, 0, 0).

calculate([Head|Tail], Position, RunningSum) ->
    show_progress(Position),
    calculate(Tail, Position + 1, RunningSum + Head);
calculate([], Position, RunningSum) ->
    show_progress("Finishing up."),
    RunningSum/Position.

show_progress(CurrentStatus) ->
    io:format("Status: ~p~n", [CurrentStatus]).
```

File: average.erl

Erlang (The VM)



.NET dev stack

C# / F#

.NET Framework

CLR

OS (Windows)

Erlang dev stack

Erlang / Elixir / Joxa

OTP

ERTS

VM
+ Kernel
+ STDLIB

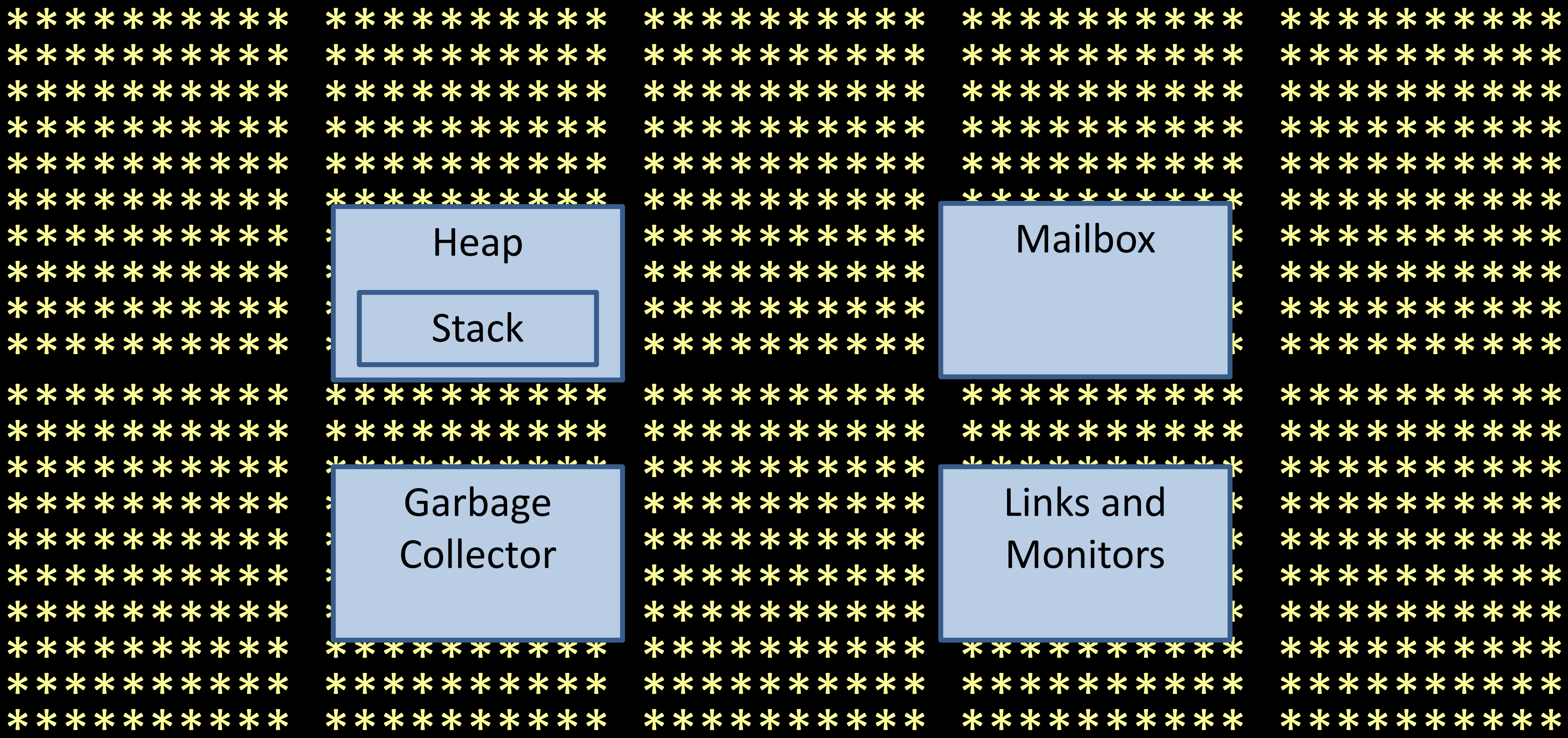
OS (Windows,
Linux, Mac,
None)

250+ years

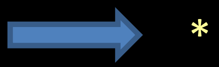
“The performance
of a concurrent language
is predicated by three things:
the **context switching time,**
The **message passing time**,
and the **time to create a process**.”

–Mike Williams

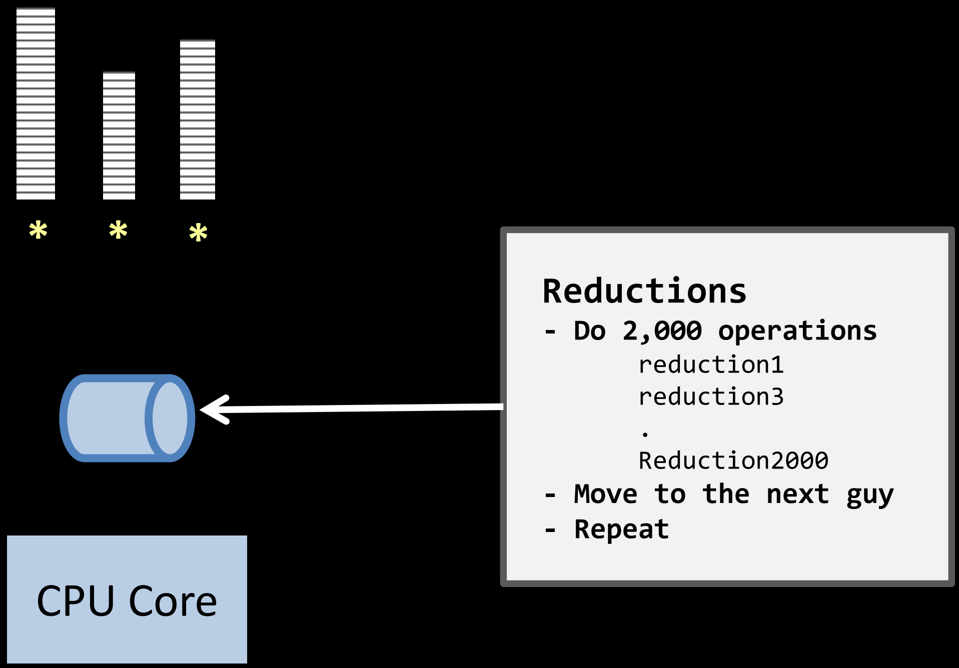
One Java 6 or .NET 4.0 Thread
(allocates one megabyte)



One Erlang Process (allocates one kilobyte)

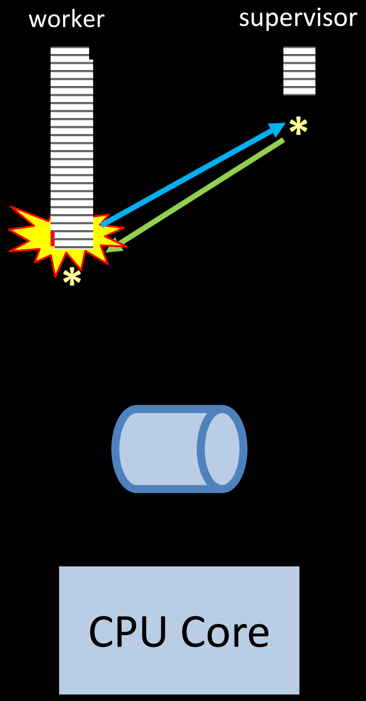


Why -> elsewhere concurrency is painful

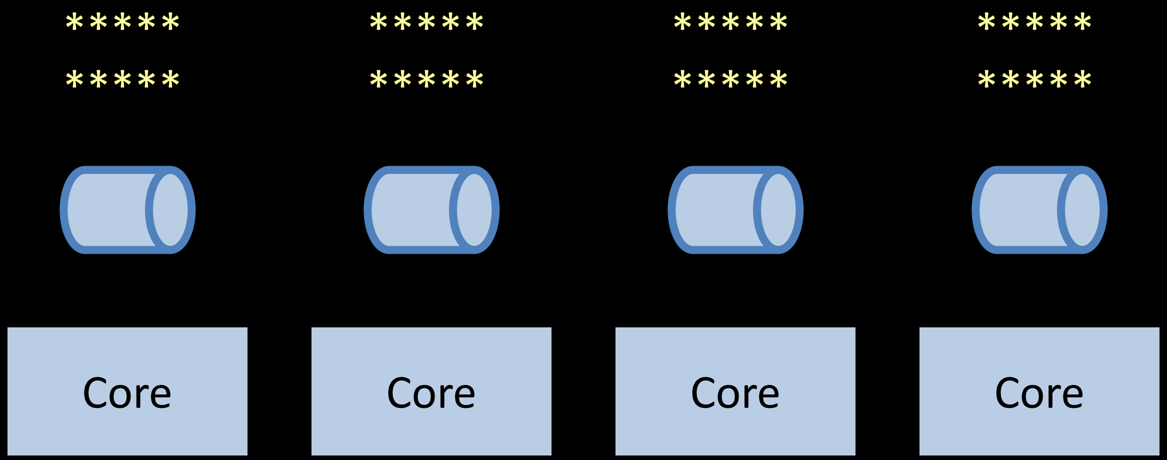


```
Reductions
- Do 2,000 operations
        reduction1
        reduction3
        .
        Reduction2000
- Move to the next guy
- Repeat
```

Why -> near zero context switching cost



Why -> supervision



***** ***** ***** *****

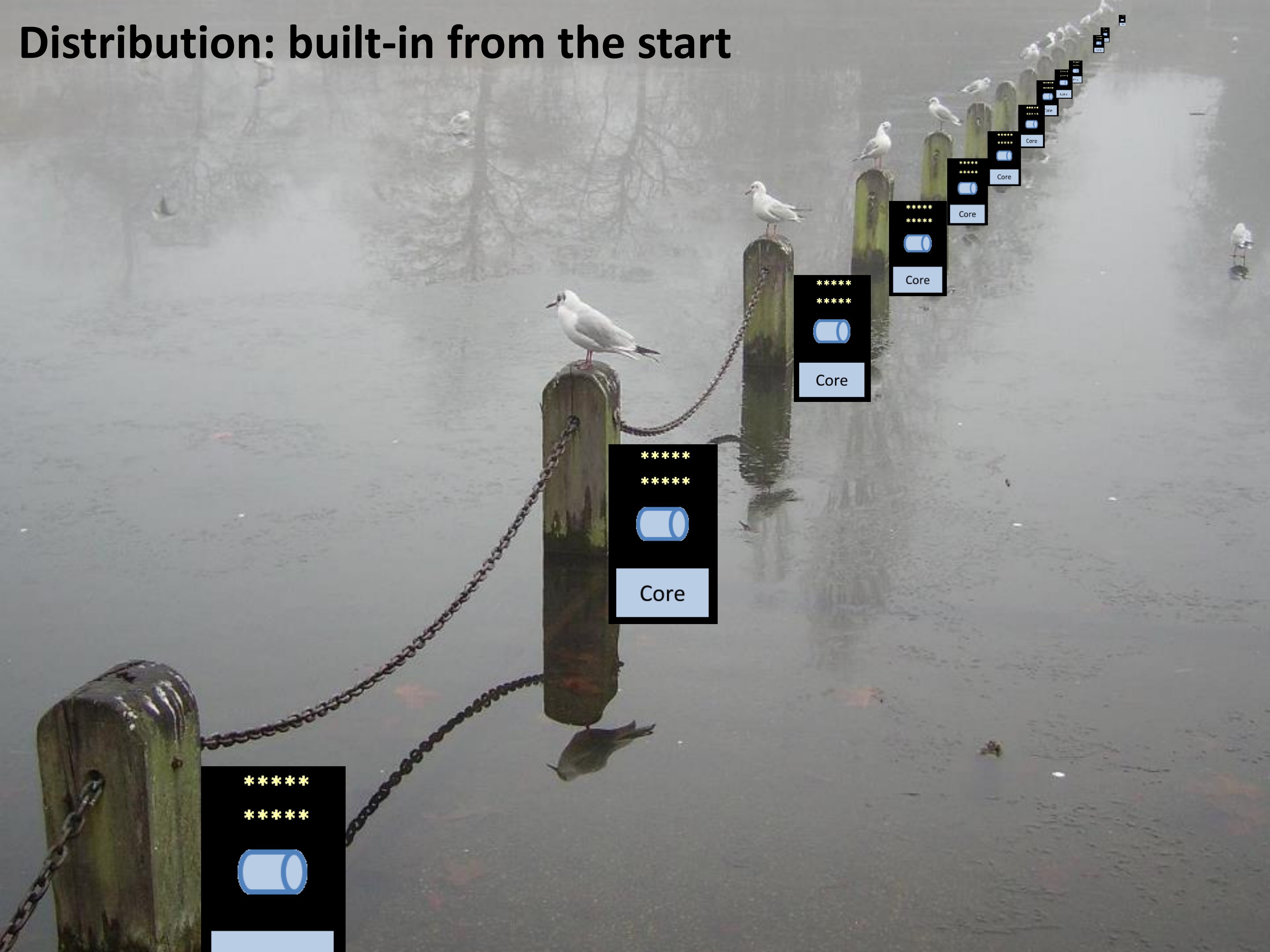
***** ***** ***** *****

Core Core Core Core

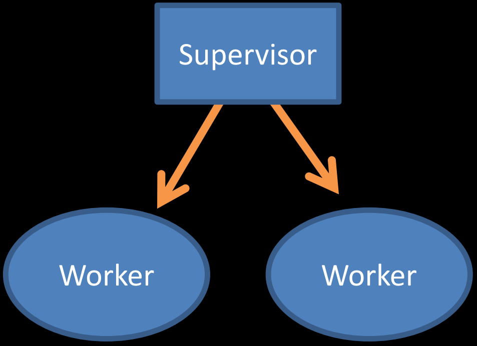
Why -> elsewhere concurrency is painful

Distribution: built-in from the start

Distribution: built-in from the start



Nines	Uptime	Annual Downtime
9	99.9999999%	30 milliseconds
8	99.9999990%	300 milliseconds
7	99.9999900%	3 seconds
6	99.9999000%	32 seconds
5	99.9990000%	5 minutes
4	99.9900000%	53 minutes
3	99.9000000%	8 hours, 46 minutes
2	99.0000000%	3 days, 15 hours, 36 minutes



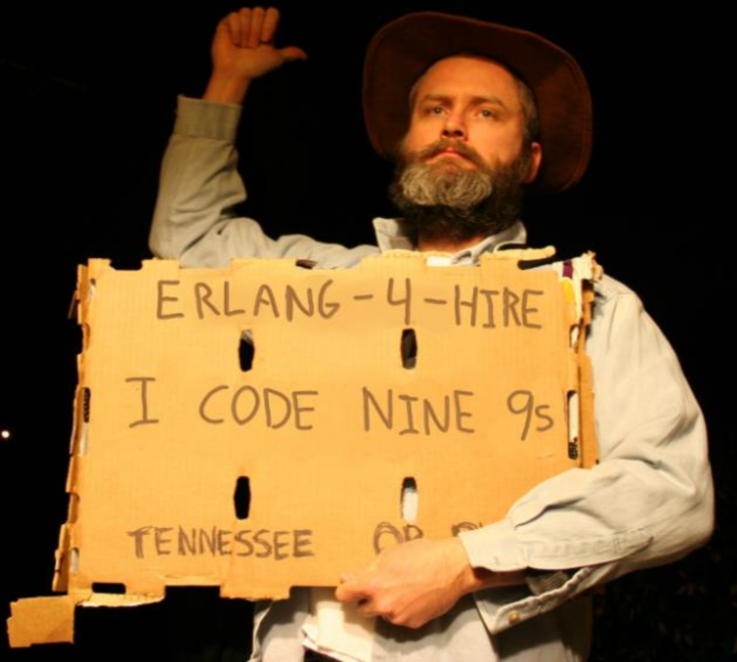
Links and monitors

No masters

Hot code loading

simple distribution, safe concurrency

Why -> reliability



When you reach for CQRS you are solving problems that led these companies

amazon.com

f

YAHOO!

EA
ELECTRONIC ARTS™

riak

wooga
world of gaming

ERICSSON

github
SOCIAL CODING

MOTOROLA

T · · Mobile ·

CouchDB
relax

RabbitMQ
Open Source Enterprise Messaging

yammer
The Enterprise Social Network

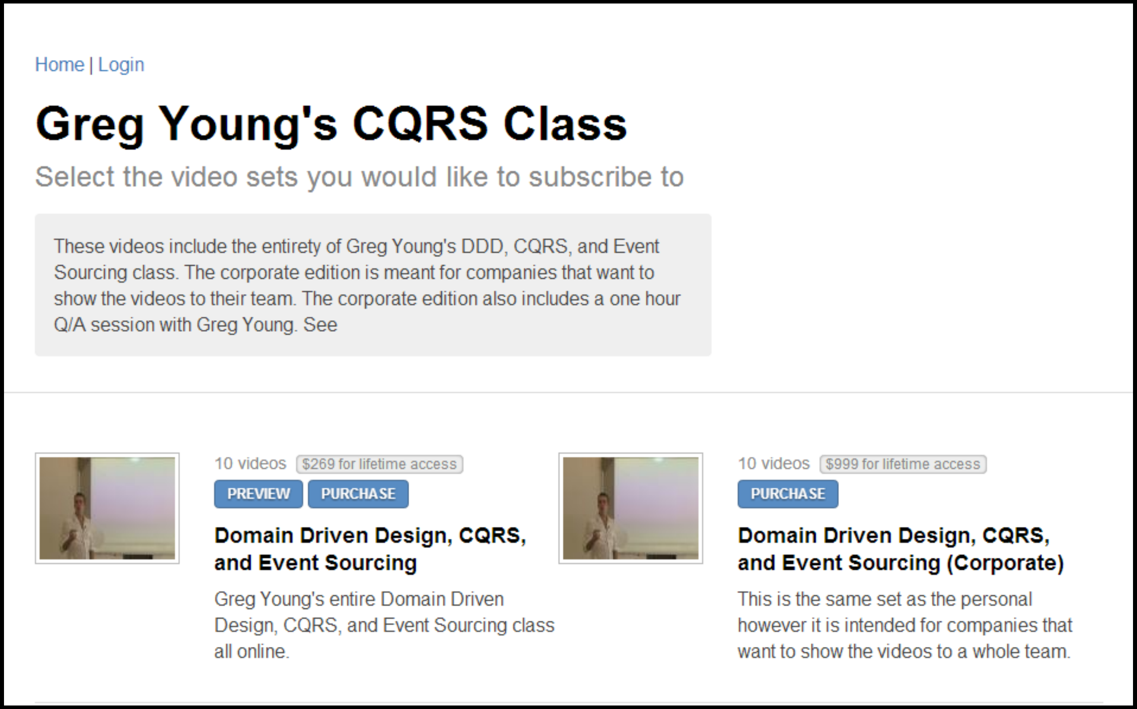
...to

ERLANG

[insert awe-inspiring demos here]

Right tool for the job!

http://subscriptions.viddler.com/GregYoung



Home | Login

Greg Young's CQRS Class

Select the video sets you would like to subscribe to

These videos include the entirety of Greg Young's DDD, CQRS, and Event Sourcing class. The corporate edition is meant for companies that want to show the videos to their team. The corporate edition also includes a one hour Q/A session with Greg Young. See

10 videos $269 for lifetime access

Domain Driven Design, CQRS, and Event Sourcing

Greg Young's entire Domain Driven Design, CQRS, and Event Sourcing class all online.

10 videos $999 for lifetime access

Domain Driven Design, CQRS, and Event Sourcing (Corporate)

This is the same set as the personal however it is intended for companies that want to show the videos to a whole team.

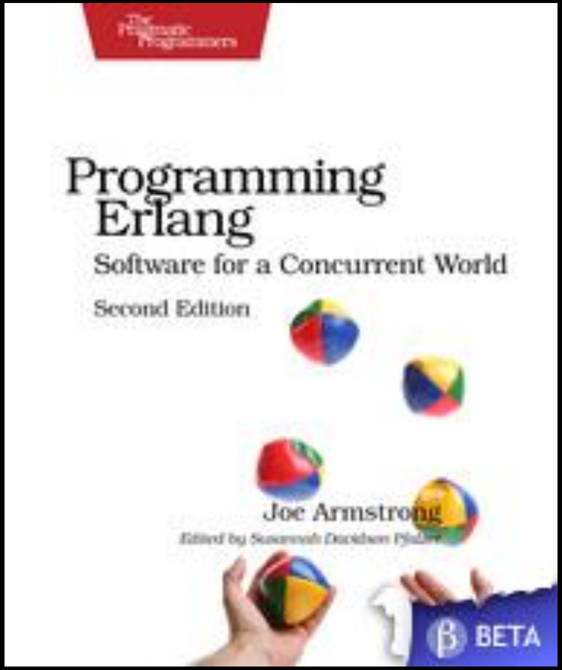
How -> watch some videos

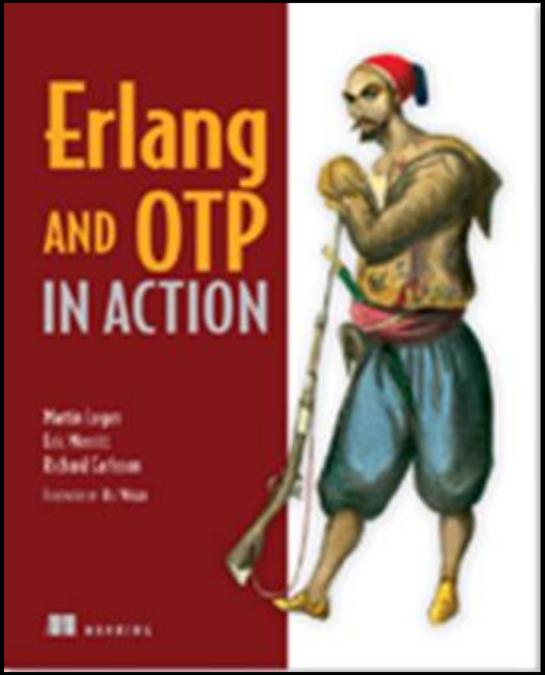
“Learn You Some Erlang for Great Good!”

http://learnyousomeerlang.com/

How -> read this excellent and free online book

Start here


The Pragmatic Programmers
Programming Erlang
Software for a Concurrent World
Second Edition
Joe Armstrong
Edited by Susannah Davidson Pfalzer
BETA



How -> read some books

http://erlangcamp.com

How -> Two day intense hands-on training

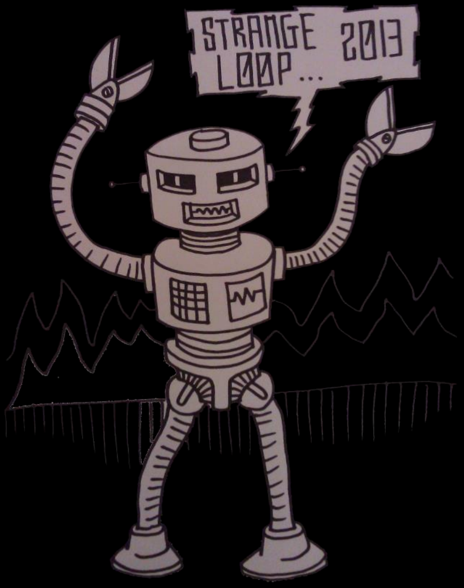
NashFP

NASHVILLE FUNCTIONAL PROGRAMMERS

http://nashfp.org

How -> Better yet; pack up and move to Nashville!

Thanks!



Bryan Hunter
Twitter: @bryan_hunter
Email: bryan.hunter@fireflylogic.com

Firefly Logic, Inc.
1000 Main Street #201
Nashville, TN 37206
http://fireflylogic.com

> **keep in touch**