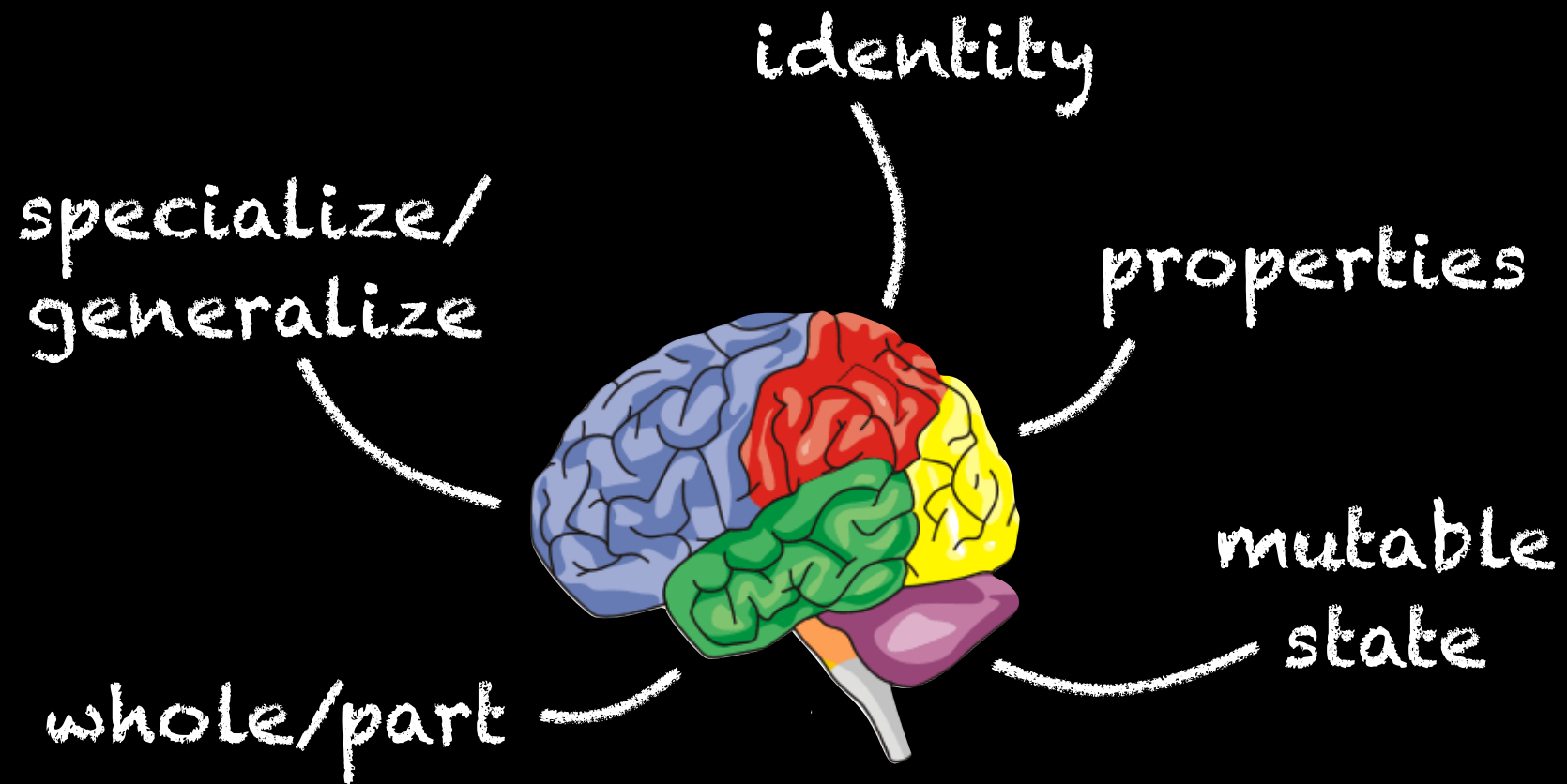


# From Java to Erlang Thinking

Kresten Krab Thorup  
CTO, Trifork

@drkrab





Java™





software pilots  
**TRIFORK.**

Java gets boring  
after a while...

software pilots  
**TRIFORK.**

# Anthropology Class

The discouraging  
truth about  
Group Dynamics

Software  
Innovation  
"as a Service"

The 10% rule

To really learn  
something:

Teach it!

Why did objects  
succeed?



Ole-Johan Dahl & Kristen Nygaard

"...each Smalltalk object is a recursion on the entire possibilities of the computer.

Thus its semantics are a bit like having thousands and thousands of computers all hooked together by a very fast network."

Alan Kay,  
HOPL 1993

Objects are  
interpreters

("book!", ...) → object → (result, ...)

"book!" → monster.say

4 → sqrt → result

[BETA Language]

# Conceptual modeling



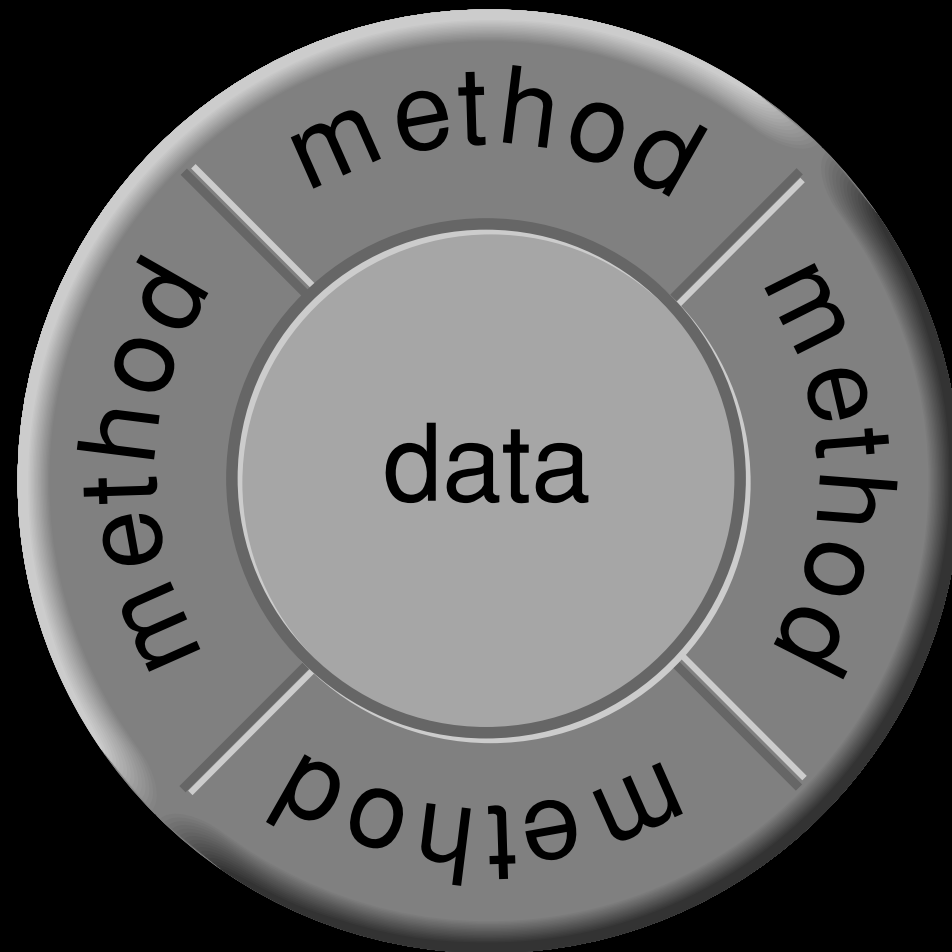
# Thinking Tools

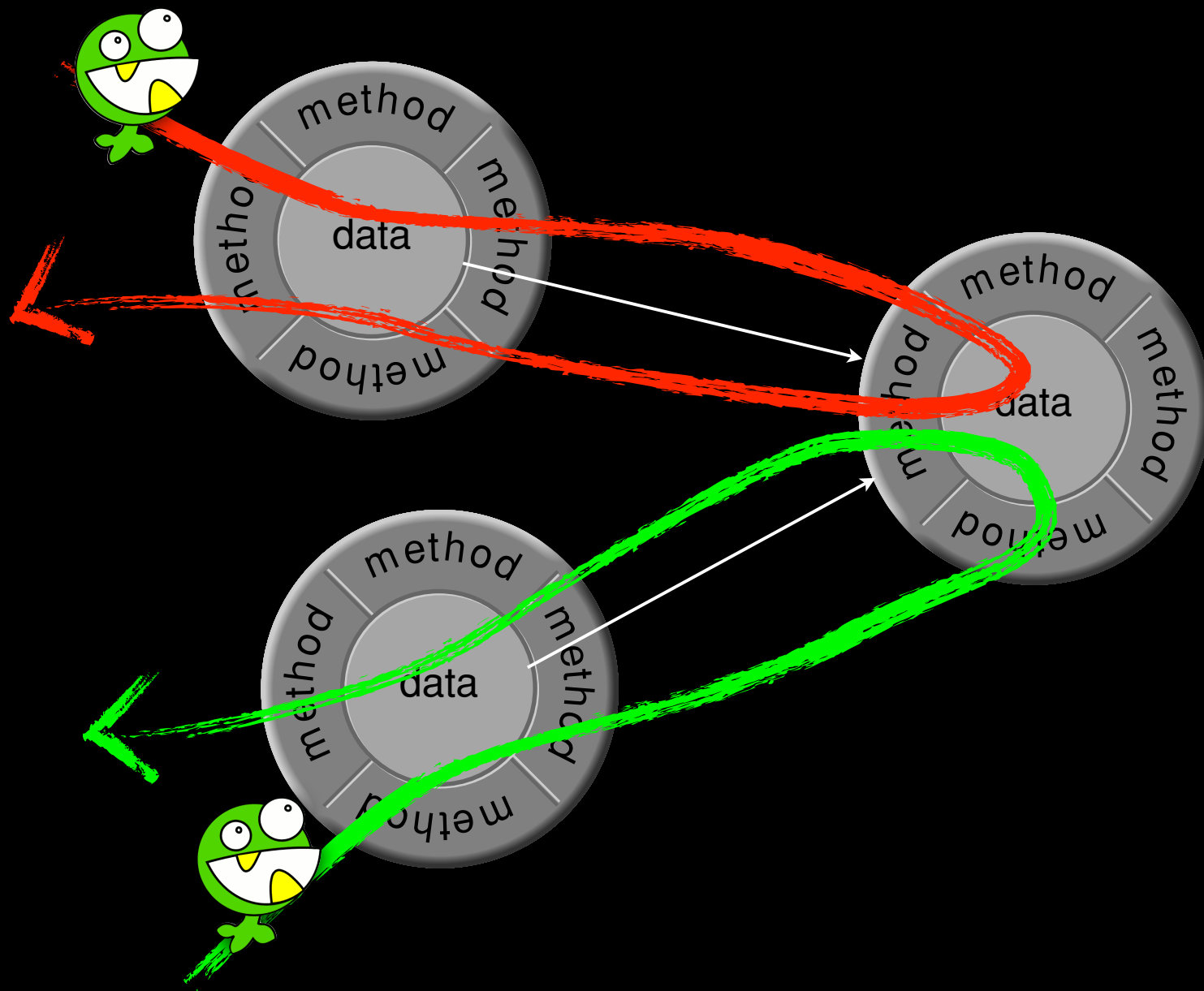
Connect the  
solution to the  
problem

**DETOUR**

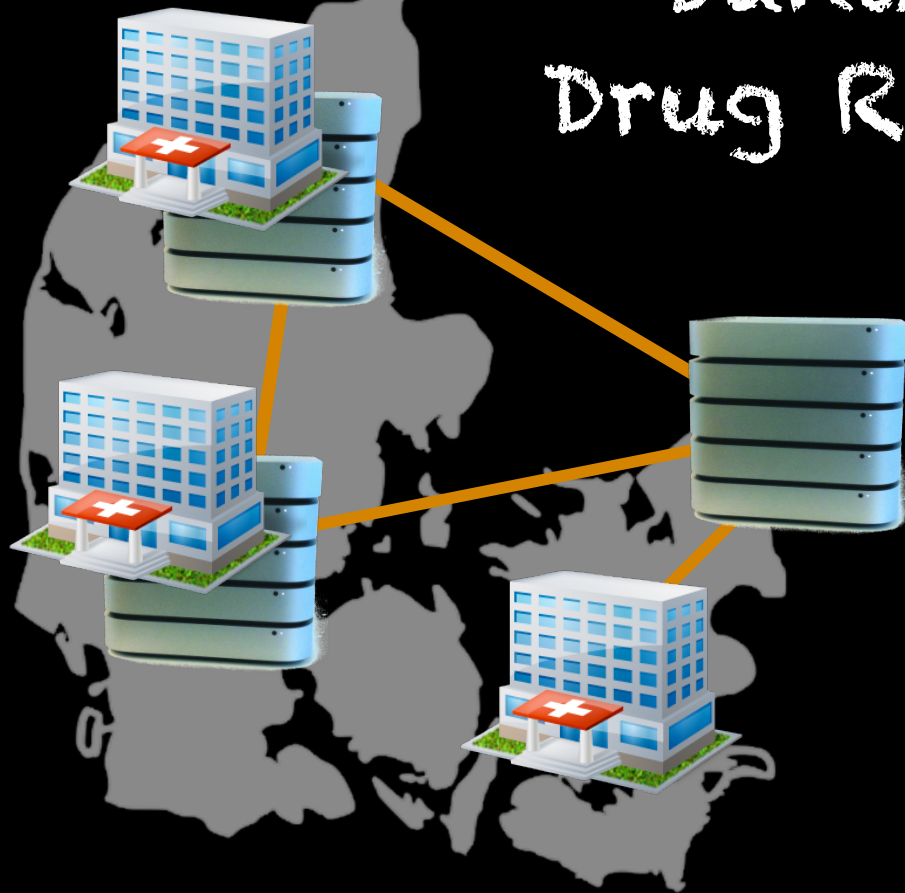








# Danish National Drug Record System



Software as  
Infrastructure

Enter: Erlang

Erlang is...

Not a functional  
concurrent  
programming  
language

Erlang is...

System for  
Reliable  
Computing

OK, I didn't  
know that

My "holy grail"

Conceptual  
framework for  
Concurrency

Your Erlang Program

OTP Framework



BEAM Emulator

BIFs

Your favorite OS

Your Erlang Program

OTP Framework

BEAM

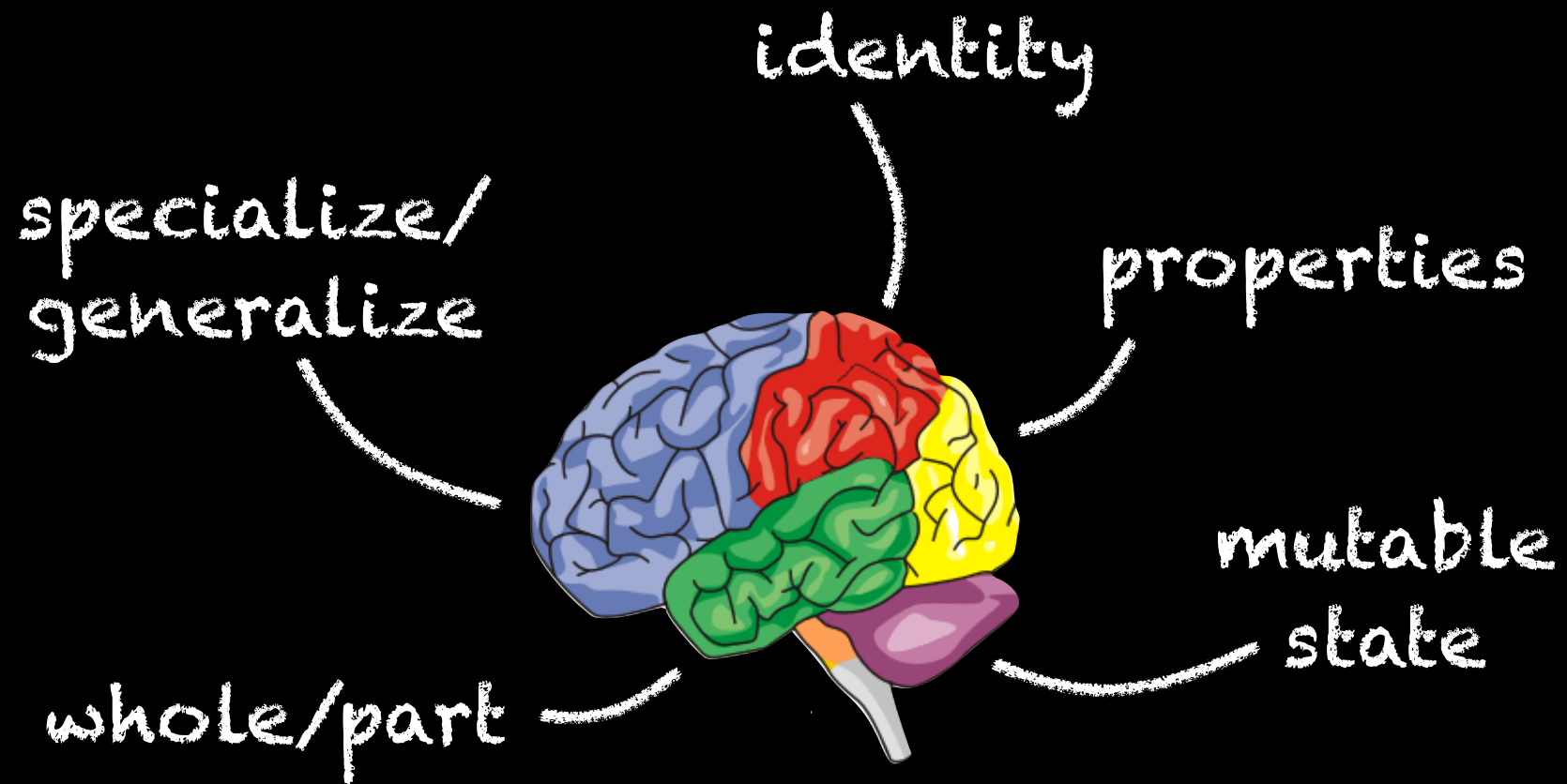
ERJANG

BIFs

JVM

Java Virtual Machine

Your favorite OS



• Polymorphism

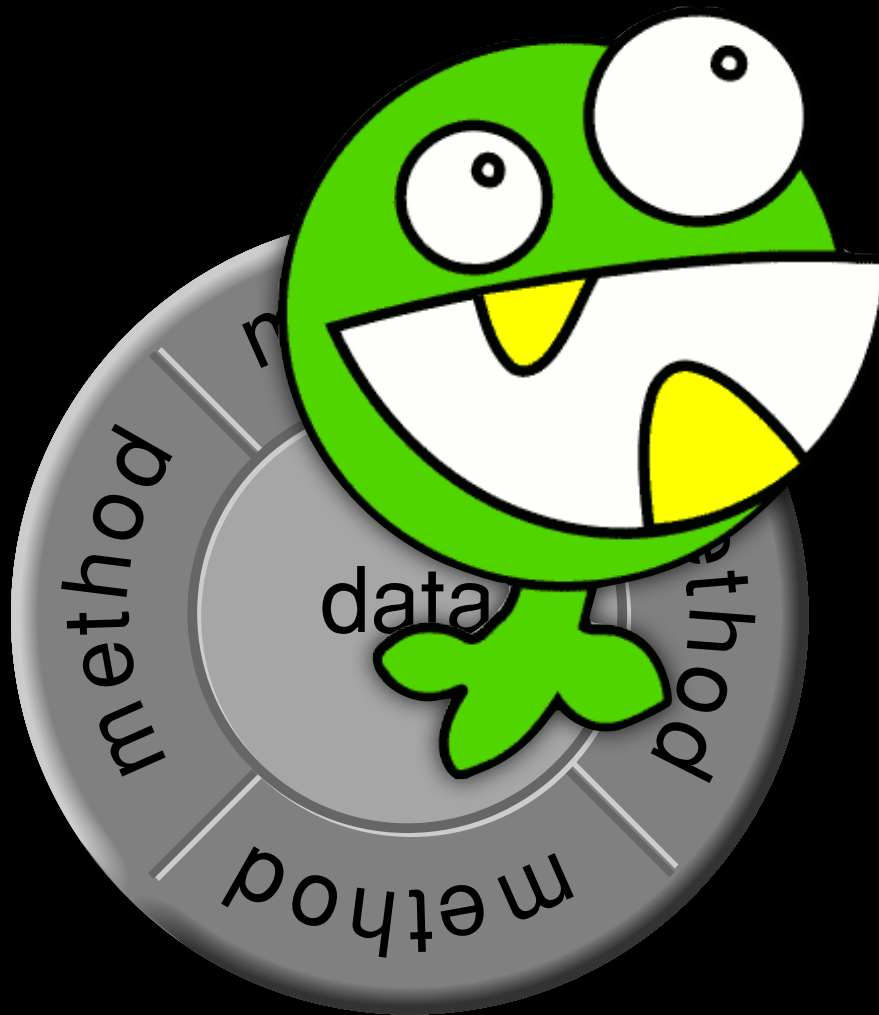
• State Management

• Systems Design

⦿ Encapsulation

⦿ Polymorphism

⦿ Systems Design



@private

@public

@protected

"...each Erlang process is a recursion on the entire possibilities of the computer.

Thus its semantics are a bit like having thousands and thousands of computers all hooked together by a very fast network."

• Encapsulation

• Polymorphism

• Systems Design

● Pattern matching is  
polymorphism for data

● Erlang does lack extensible  
data types (the equivalent of  
clojure's protocols) .. Elixir

set.size()

vs.

size(set)

`set.size()`

vs.

`set_module::size(set)`

```
set.intersection(set2)
```

vs.

```
intersection(set, set2)
```

- Message interpretation (receive) is polymorphism for process interaction
- Necessary for distributed systems in order to loosen the coupling between peers

• Encapsulation

• Polymorphism

• Systems Design

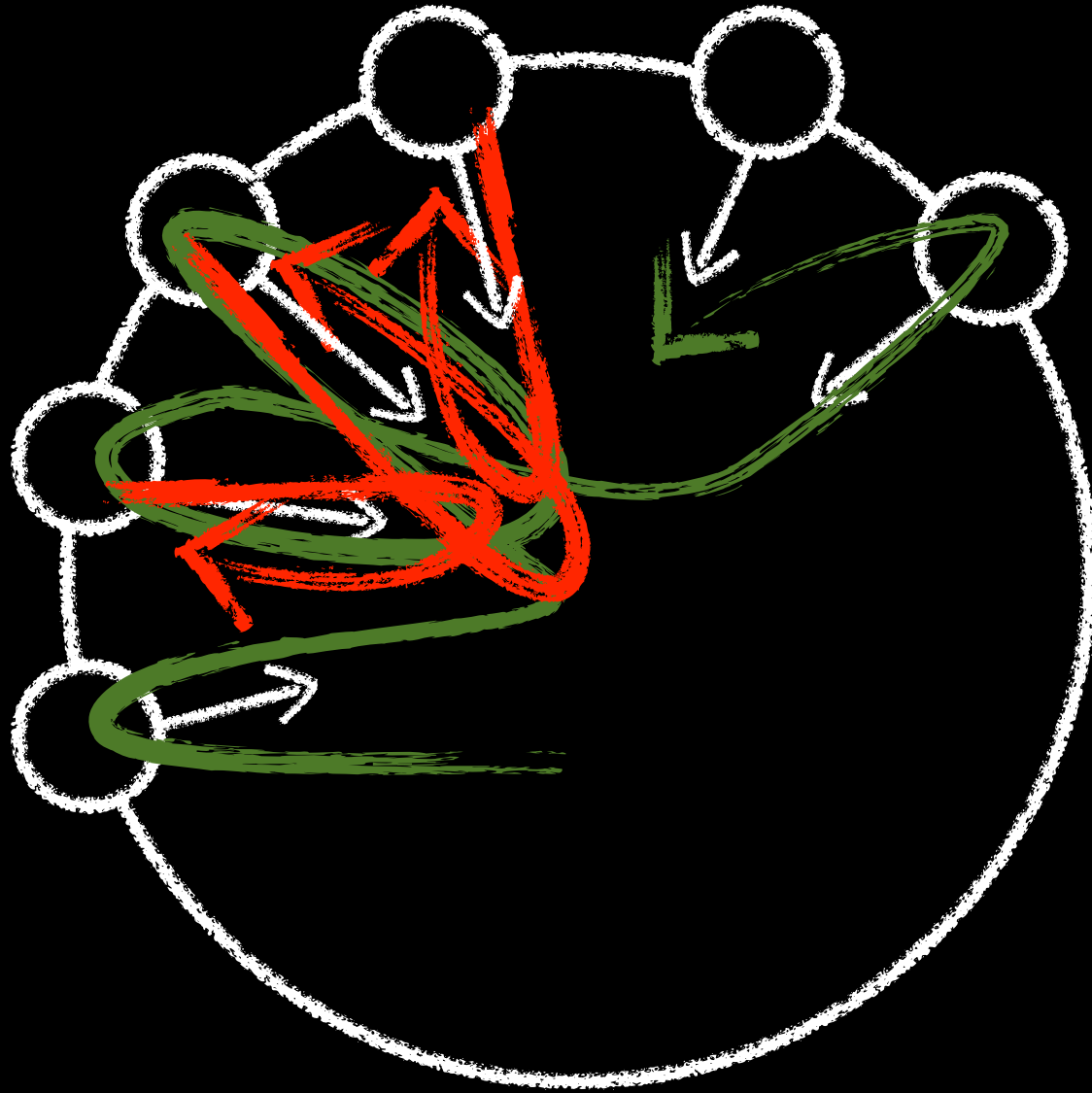
Erlang is...

FP Language

---

OS for code

# Pull vs. Push



# Concurrency

Interactive

Coordination

Messaging  
+ processes

# Parallelism

Functional

Computation

Utilize  
hardware

# Concurrency

Want to  
understand

# Parallelism

Want to NOT  
understand

Thanks

@drkrab

