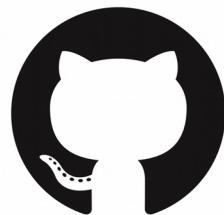
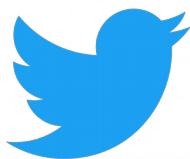


# Who is this guy talking?



@tapickell



**Todd Pickell**  
tapickell

Remote software developer working on Erlang, Elixir, Ruby, NodeJS, Haskell, React, Arduino, RaspberryPi, AWS and whatever else I can dig into.

- 
- TAP Software Development
  - North San Diego County
  - [tapickell@gmail.com](mailto:tapickell@gmail.com)
  - <http://tap-software.com/>
-

H. A. Q. T.

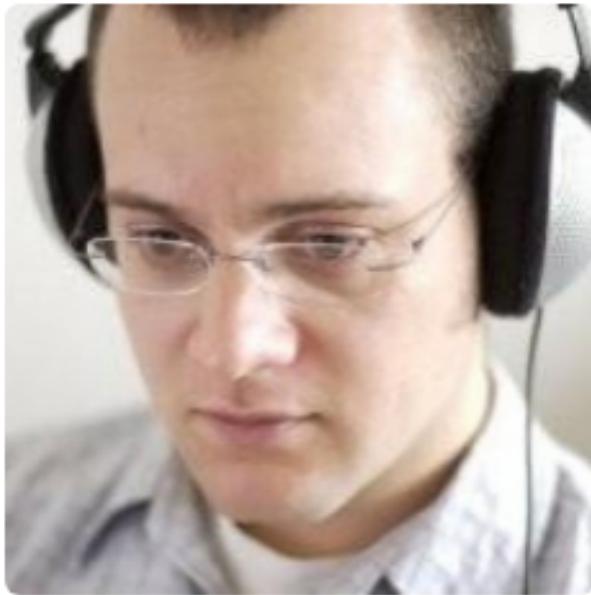
Handler |> Action |> Query |> Transform

<https://github.com/tapickell/haqt-elixir>

# It is an Idea



# Not another Framework



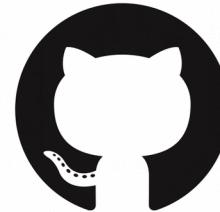
**Ethan Garofolo**  
juanpaco

[Unfollow](#)

[Block or report user](#)

- 
-  [Such Software, LLC](#)
  -  [Provo, UT](#)
  -  [juanpaco@gmail.com](mailto:juanpaco@gmail.com)
  -  <http://learnallthenodes.com>

# Who's Idea was this anyways?



@ethangarofolo

The image shows a man with a microphone, wearing a tie-dye shirt, speaking. To his right is a diagram titled "THE 7 LAYERS OF OSI". The diagram illustrates the OSI model's layers and their corresponding PDU (Protocol Data Unit) sizes. It shows two hosts connected by a "Physical Link, or Medium" (Layer 8). The layers from top to bottom are: Application layer (Layer 7), Presentation layer (Layer 6), Session layer (Layer 5), Transport layer (Layer 4), Network layer (Layer 3), Data link layer (Layer 2), and Physical layer (Layer 1). On the left, under "TRANSMIT", data flows from Layer 7 down to Layer 1. On the right, under "RECEIVE", data flows from Layer 1 up to Layer 7. A legend at the top defines the colors: red for Layer 7, pink for Layer 6, orange for Layer 5, yellow for Layer 4, green for Layer 3, blue for Layer 2, and dark green for Layer 1.

THE 7 LAYERS OF OSI

PDU (Protocol Data Unit)  
(units of data passed between layers)

Header Data

TRANSMIT

RECEIVE

Term for a unit of data at this layer

Term for a unit of data at this layer

DATA

USER (Sometimes called "Layer 8")

Application layer

Presentation layer

Session layer

Transport layer

Network layer

Data link layer

Physical layer

Data

Data

Data

Segment

Segment

Packet, Datagram

Packet, Datagram

Frame, Cell

Frame, Cell

Frame, Bit

Frame, Bit

Physical Link, or Medium - sometimes called "Layer 8" - data unit is a "bit"

The only architecture I know...  
...is layered. - Erik Meijer

Does Erik Meijer own a  
shirt that is not tie dye?





MONOLITH



SOA



MICROSERVICES



We've gone from building a single ball of mud to orchestrating a lot of shit. —Hadi Hariri

Exit full screen

▶ ▶! 🔍 15:43 / 37:25

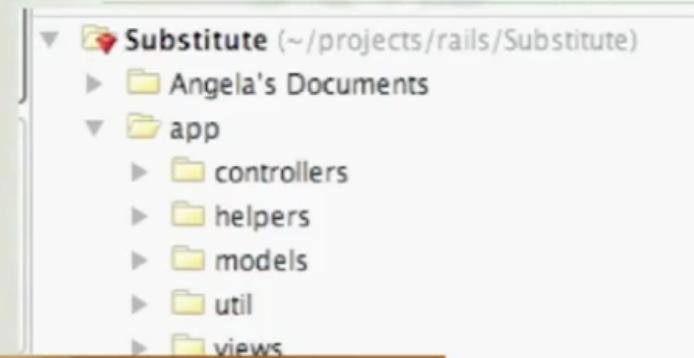
CC HD #

If we ever feel like we are just separating a monolith into smaller pieces...

# Back to the layers....



## A RAILS APP



```
1  <?xml version="1.0"
2  <!DOCTYPE html
3      PUBLIC "-//W3C//
4          "http://www.w3.
5
6      <html xmlns="http://www.w3.org/1999/xhtml">
7          <head>
8              <title>RSpec resu
9              <meta http-equiv=
10             <meta http-equiv=
11             <script type="text/javascript">
12                 function nopeProg
```



How do we know what this application does?  
We don't it's just a Rails App.



**Gary Bernhardt**

@garybernhardt

the thirst quencher

📍 Seattle, WA

🔗 [destroyallsoftware.com](http://destroyallsoftware.com)

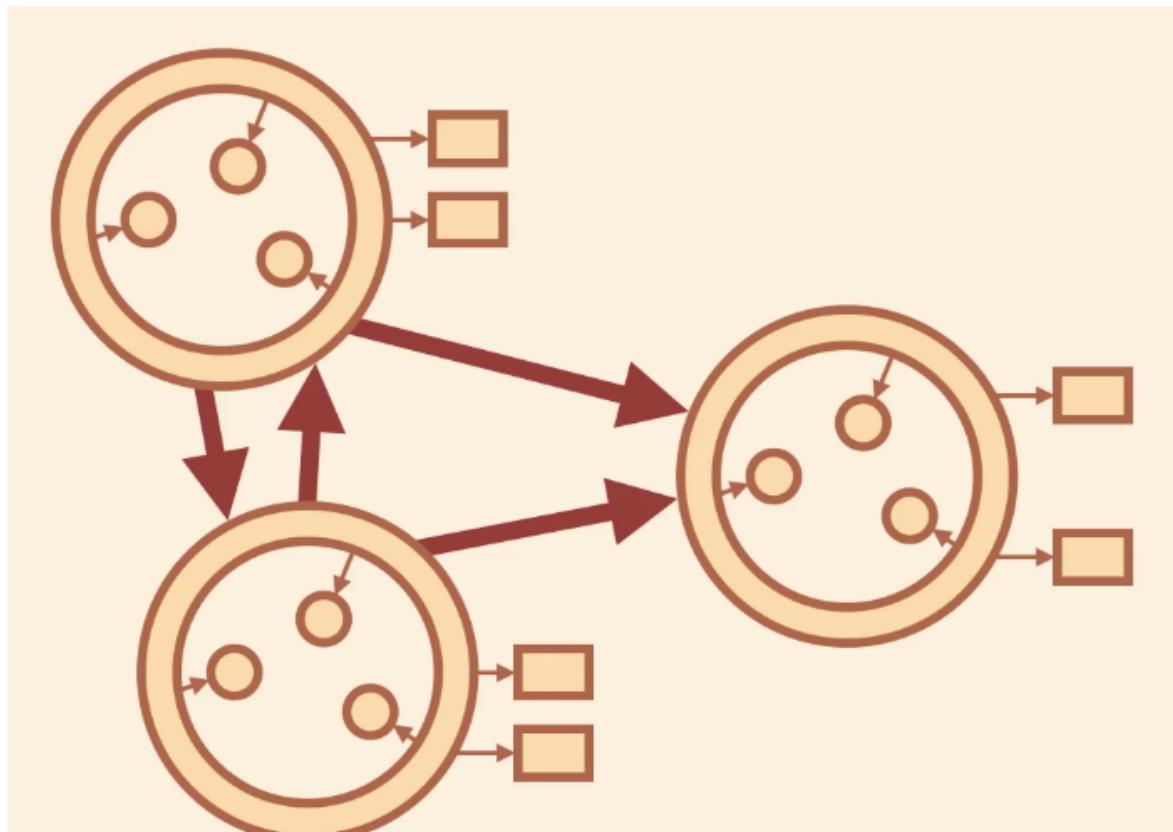
📅 Joined March 2007

## Natural Isolation

**Value  
in + out**

**No  
deps**

Boundaries Talk  
Describe a function for “+”



# Functional Core, Imperative Shell

Kinda like Jelly filled doughnuts.



Jessica Kerr  
@jessitron

Following



How to make a command-line application:

- 1) write a library that does the work
  - 2) write a CLI that uses it.
- taggart

RETWEETS

68

LIKES

43



11:11 AM - 10 Sep 2014

12

68

43

This is how we should build software right?  
Build the domain logic, then build interfaces to interact with it.

H. A. Q. T.

Handler |> Action |> Query |> Transform

# Handler

input

- |> grabs data that is needed
- |> appropriate action

# Action

data

|> validation

|> error handling

|> appropriate query/queries

# Query

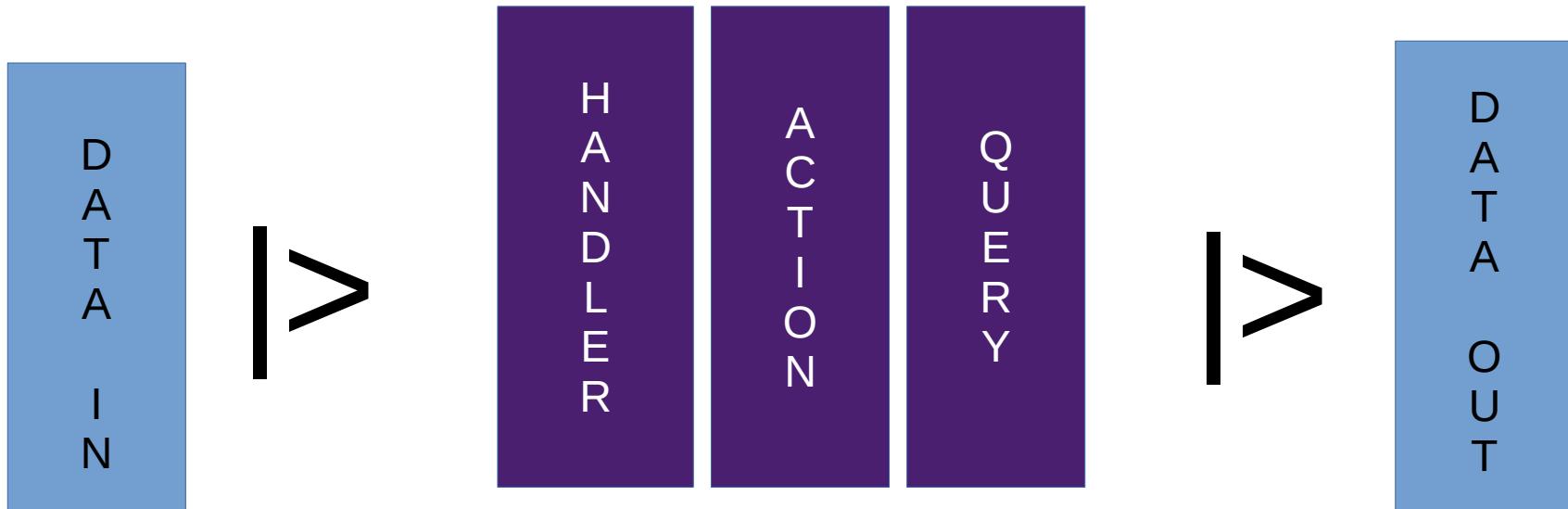
data  
|> appropriate transformation  
|> execute query

# Transform

data

|> transformation

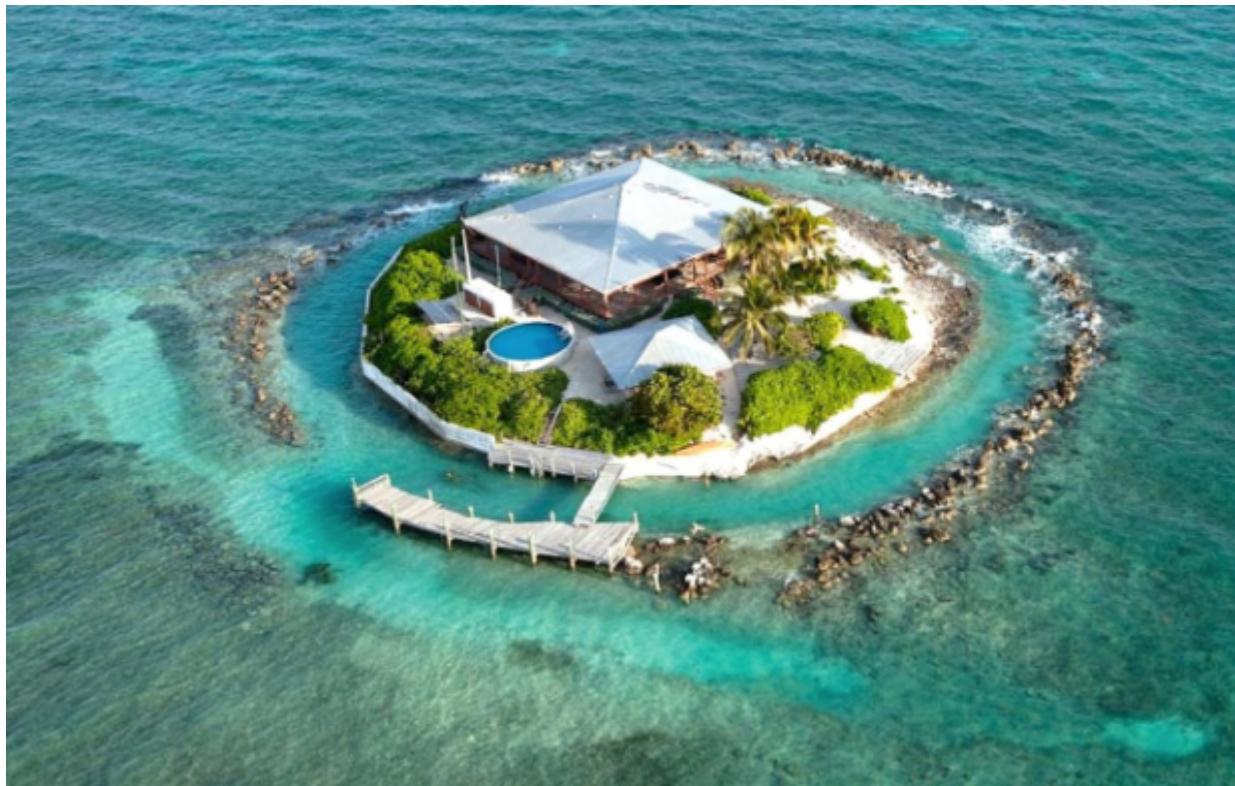
# Data flows through this pattern



HTTP Request  
Event from stream  
CLI Tool  
Message from Queue  
Message from other process  
???

Database  
ETS  
GenServer  
FlatFile  
API call  
???

# Natural Isolation



**SEPARATION  
LEADS  
TO  
THOUGHT**

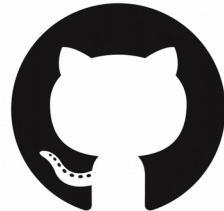
# DEMO

# It is an Idea



# Not another Framework

# Who is this guy talking?



@tapickell



**Todd Pickell**  
tapickell

Remote software developer working on Erlang, Elixir, Ruby, NodeJS, Haskell, React, Arduino, RaspberryPi, AWS and whatever else I can dig into.

- 
- TAP Software Development
  - North San Diego County
  - [tapickell@gmail.com](mailto:tapickell@gmail.com)
  - <http://tap-software.com/>
-