Building a Graphical IDE in Elm

for a Distributed PLC Language Compiling to BEAM

by @doppioslash

09/09/2016 - Erlang User Conference - Stockholm

DIPL. PHYS. PEER STRITZINGER GMBH

Hi, I'm

Claudia Doppioslash

Functional Programmer & **Game Developer**

@doppioslash
www.lambdacat.com

DIPL. PHYS. PEER STRITZINGER GMBH

Peer Stritzinger GmbH

Functional and Failure Tolerant
Programming for Embedded,
Industrial Control and Automotive

DIPL. PHYS. PEER STRITZINGER GMBH

www.stritzinger.com

DIPL. PHYS. PEER STRITZINGER GMBH

Why are you here?

“I need to get some frontend code done, and I hate Javascript”

Interested in Haskell-like languages

“I was promised a embedded Erlang demo”

What are you getting

This is a WIP-mortem:

- why we made the choices we made
- what went right/wrong
- enough Elm to understand what's going on
- a demo of embedded Erlang + Elm client

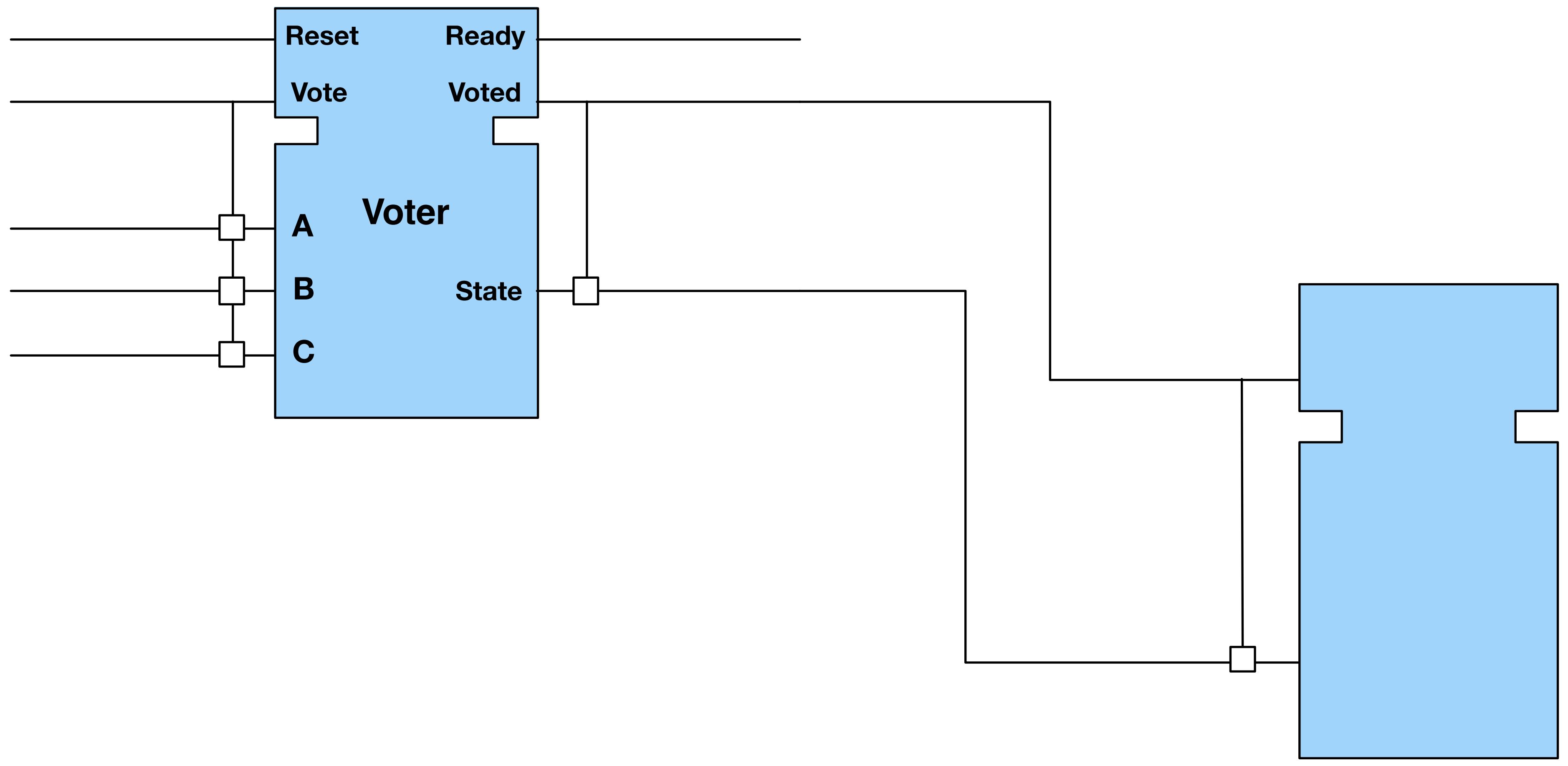
Not an Elm guide, also not latest Elm version.

Our Project

Intelligente Cyber-physische IT-Systeme
Kompetenz in Produktion und IT für multiadaptive Fabriken: der Mensch im Mittelpunkt
smart F-IT
future talk
Rexroth
ZeMA
APAS assistant

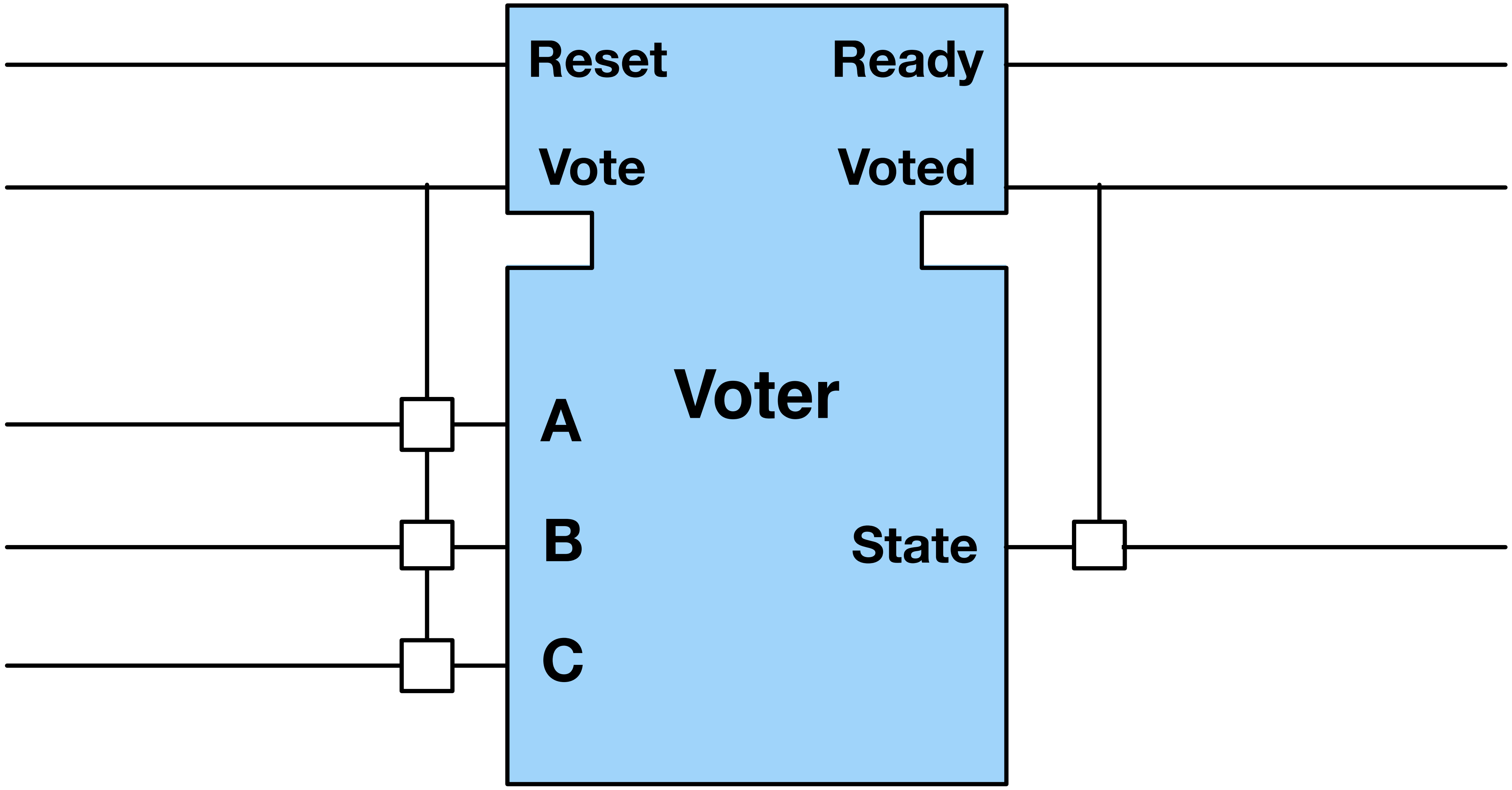
DIPL. PHYS. PEER STRITZINGER GMBH

Event flows with Data

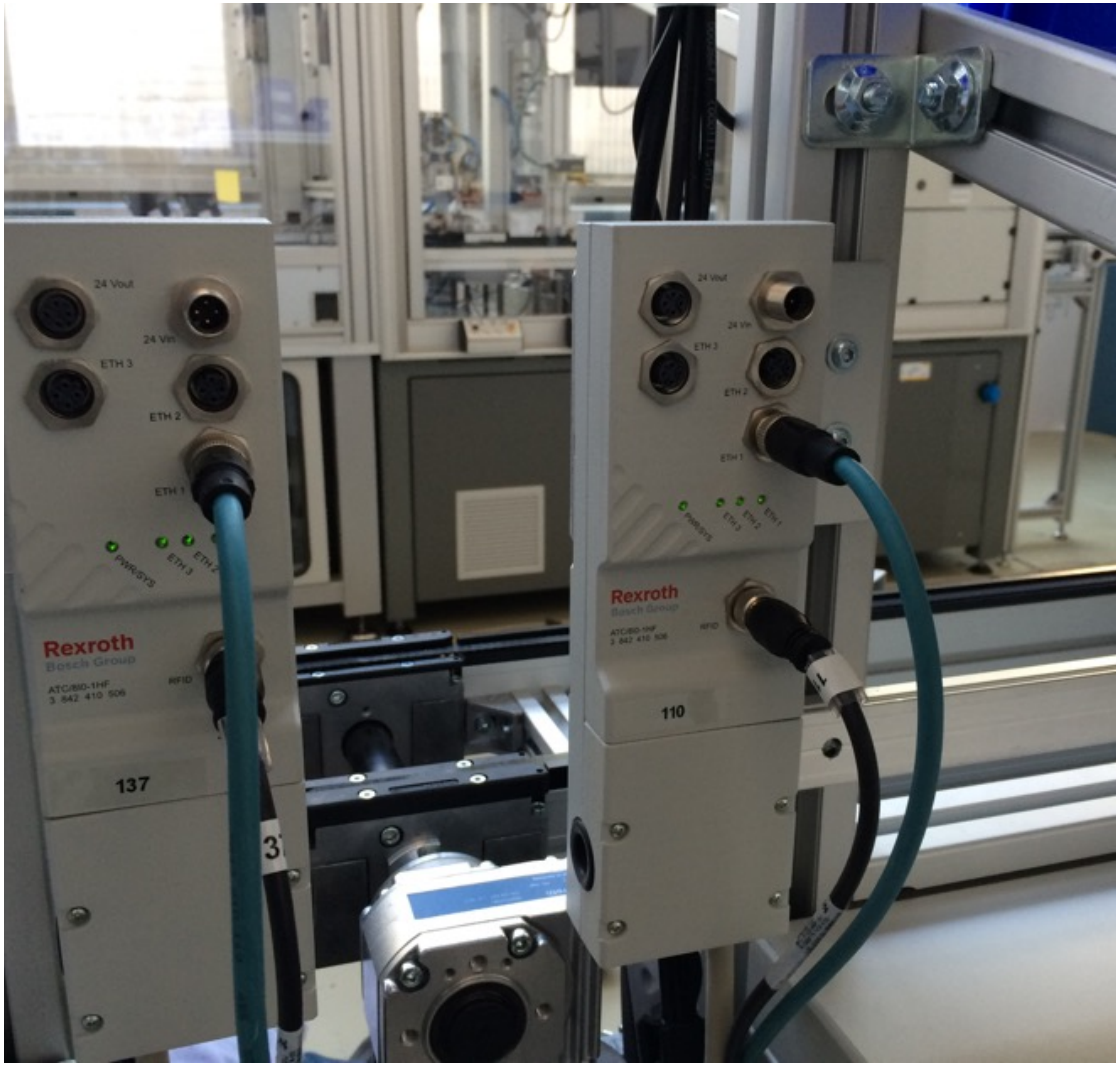


DIPL. PHYS. PEER STRITZINGER GMBH

Distributed PLC with IEC61499



DIPL. PHYS. PEER STRITZINGER GMBH


24 Vout
24 Vin
ETH 3
ETH 2
ETH 1
PWR/SYS
Rexroth
Bosch Group
ATC/BID-1HF
3 842 410 506
RFID
137
110

CeBIT
future talk

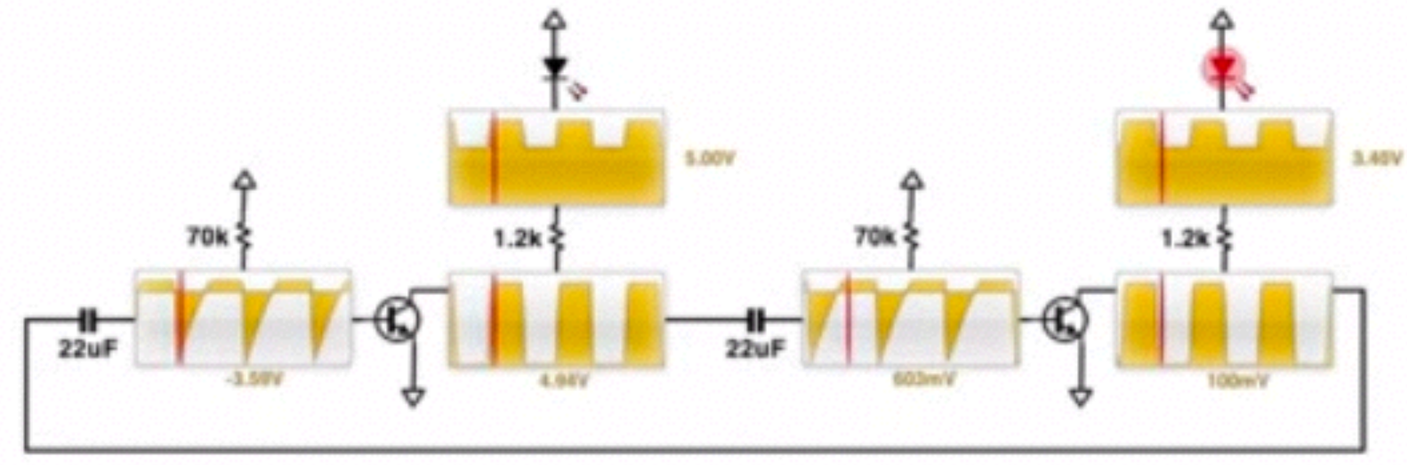
Rexroth

Our Project

Visual IDE for PLC language **IEC61499**

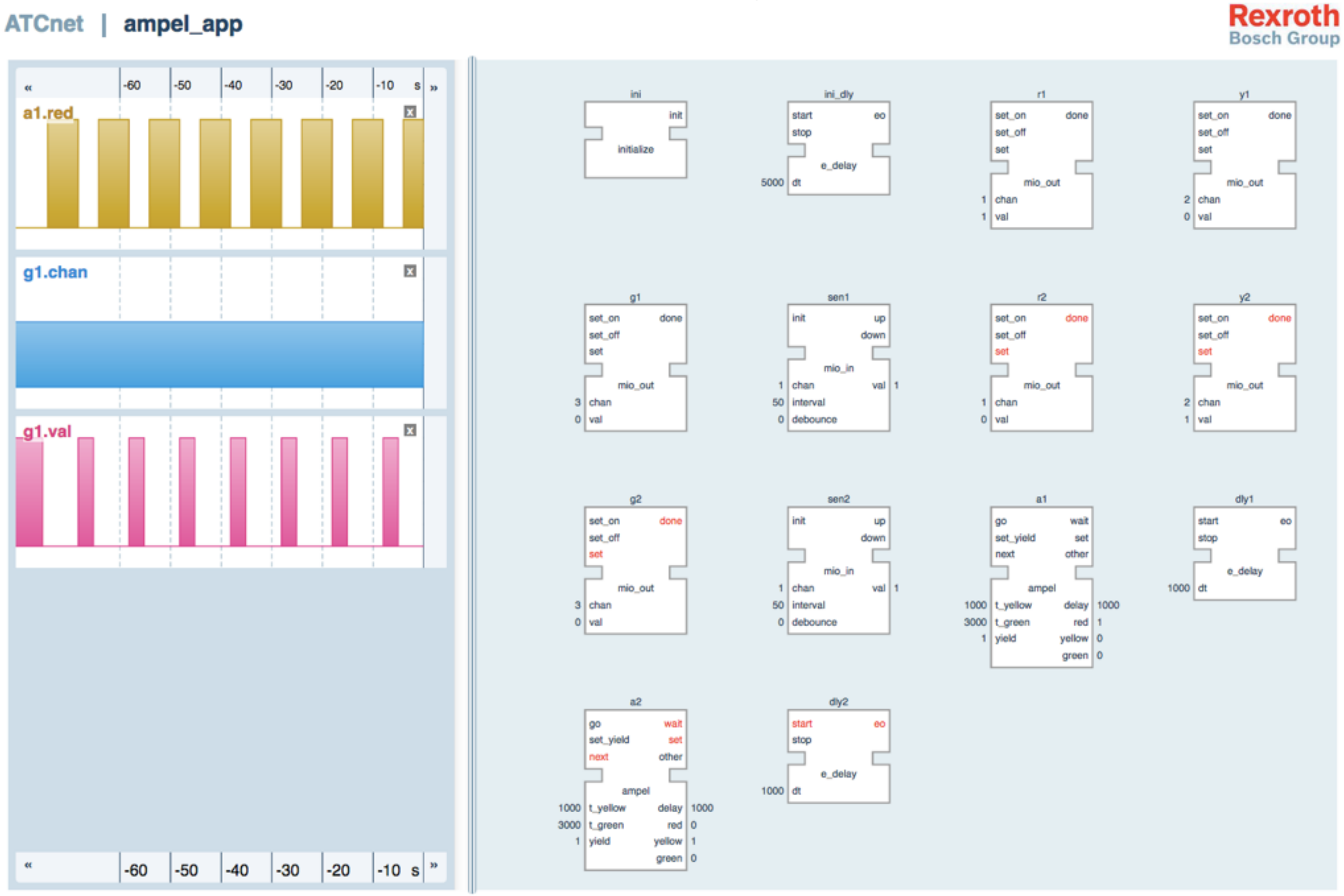
“A programmable logic controller, PLC, or programmable controller is a digital computer used for automation"

Inspired by Bret Victor’s “Inventing on Principle” talk:



DIPL. PHYS. PEER STRITZINGER GMBH

Our Project


ATCnet | ampel_app
Rexroth Bosch Group
a1.red
g1.chan
g1.val
-60 -50 -40 -30 -20 -10 s
ini: init, initialize
ini_dly: start, stop, eo, e_delay, 5000 dt
r1: set_on, set_off, set, done, mio_out, 1 chan, 1 val
y1: set_on, set_off, set, done, mio_out, 2 chan, 0 val
g1: set_on, set_off, set, done, mio_out, 3 chan, 0 val
sen1: init, up, down, mio_in, 1 chan, 50 interval, 0 debounce, val 1
r2: set_on, set_off, set, done, mio_out, 1 chan, 0 val
y2: set_on, set_off, set, done, mio_out, 2 chan, 1 val
g2: set_on, set_off, set, done, mio_out, 3 chan, 0 val
sen2: init, up, down, mio_in, 1 chan, 50 interval, 0 debounce, val 1
a1: go, set_yield, next, wait, set, other, ampel, 1000 t_yellow, 3000 t_green, 1 yield, delay 1000, red 1, yellow 0, green 0
dly1: start, stop, eo, e_delay, 1000 dt
a2: go, set_yield, next, wait, set, other, ampel, 1000 t_yellow, 3000 t_green, 1 yield, delay 1000, red 0, yellow 1, green 0
dly2: start, stop, eo, e_delay, 1000 dt

Requirements

Many platforms to support

All PC OSs & iPad Pro

Decent performance

Needs to be interactive
~30fps should be fine

Frontend Tech Choice

Web Technologies because cross-platform

Hence: **Javascript, CSS, Svg**

Wait a minute, Javascript?

…let’s not.

Possible Choices, Then

Ready at the time:

- Clojurescript
- Elm
- CoffeScript
- Typescript

Possible Choices, Now

Ready now:

- Purescript
- Fable
- Reason
- Clojurescript
- Bucklescript
- Elm
- …

Why Elm?

Functional Reactive Programming

(it's gone now though)

Good error messages

(so good everyone is imitating them)

No runtime exceptions

Some concept somewhat similar to Erlang

(e.g. Mailboxes)

What is Elm?

Pure Functional

Strongly Typed

Eagerly evaluated

Compiles to **Javascript**

Functional Reactive Programming (< 0.17)

Haskell-like syntax

Very **small**

Optimised for **learning curve** (>0.16)

Similar to Haskell but no advanced types

Elm package manager enforces **semantic versioning**

DIPL. PHYS. PEER STRITZINGER GMBH

Elm Pros compared to JS

If it compiles, it works (90% of the time)

Confident **refactoring**

Clean

Much fewer LOC

The famous great error messages

DIPL. PHYS. PEER STRITZINGER GMBH

The famous Elm errors

They are good, because:

- contextual
- correct common errors
- carefully tracked on a git repo

But

DIPL. PHYS. PEER STRITZINGER GMBH

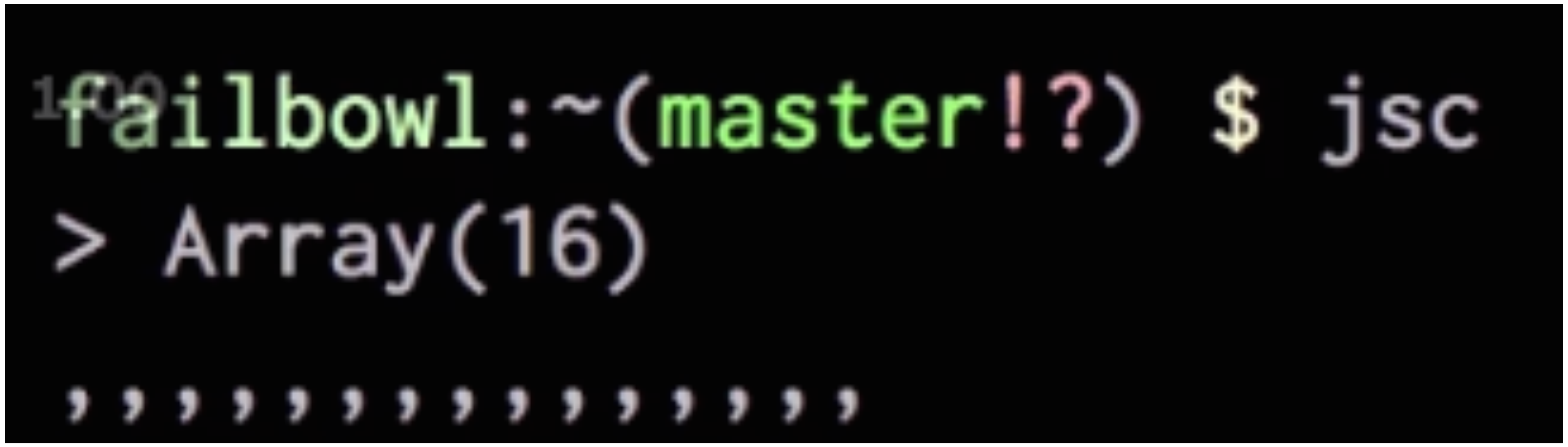
The famous Elm errors

you can **call** something wrong
or **define** something wrong

and it defaults on wrong definition
while it would be more useful to find incorrect use

Elm Pros compared to JS

Elm actually makes sense (seen the '**Wat**' talk?)



```
failbowl:~(master!?) $ jsc
> Array(16)

,,,,,,,,,,,,,,,
```

DIPL. PHYS. PEER STRITZINGER GMBH

Elm Cons compared to JS

Javascript **interop inflexible**

(less in 0.17)

new language, still 0.x

…so, not that much.

DIPL. PHYS. PEER STRITZINGER GMBH

0.16? 0.17?

The jump from 0.16 and 0.17 in Elm

0.16

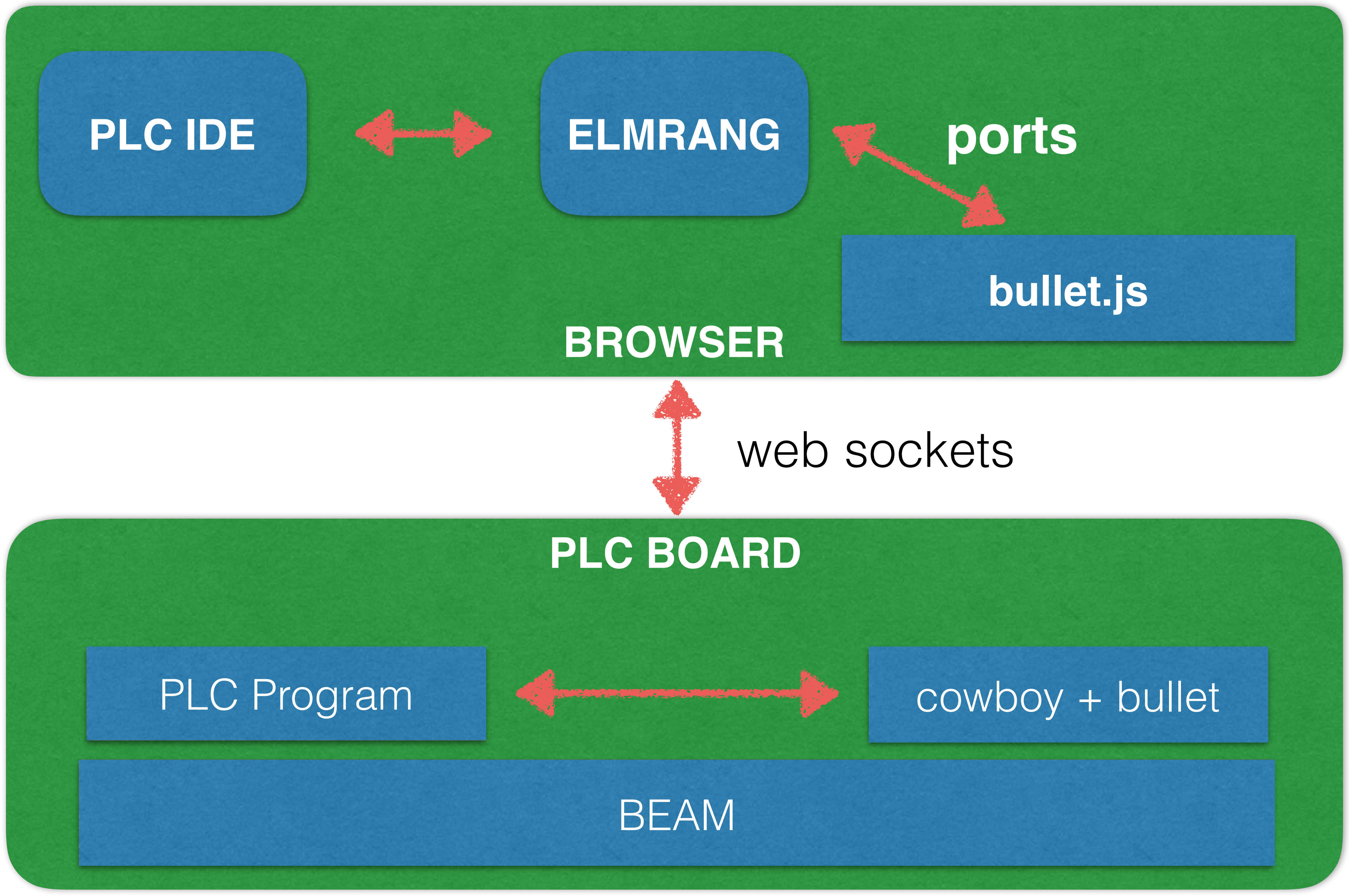
FRP
mailboxes
addresses
signals
foldp

0.17

Confusing name overlap with Erlang

mailboxes are sent **signals**
through **addresses**
signals are streams of values
foldp accumulates the state
ports are “doors” into JS, of a certain type-shape

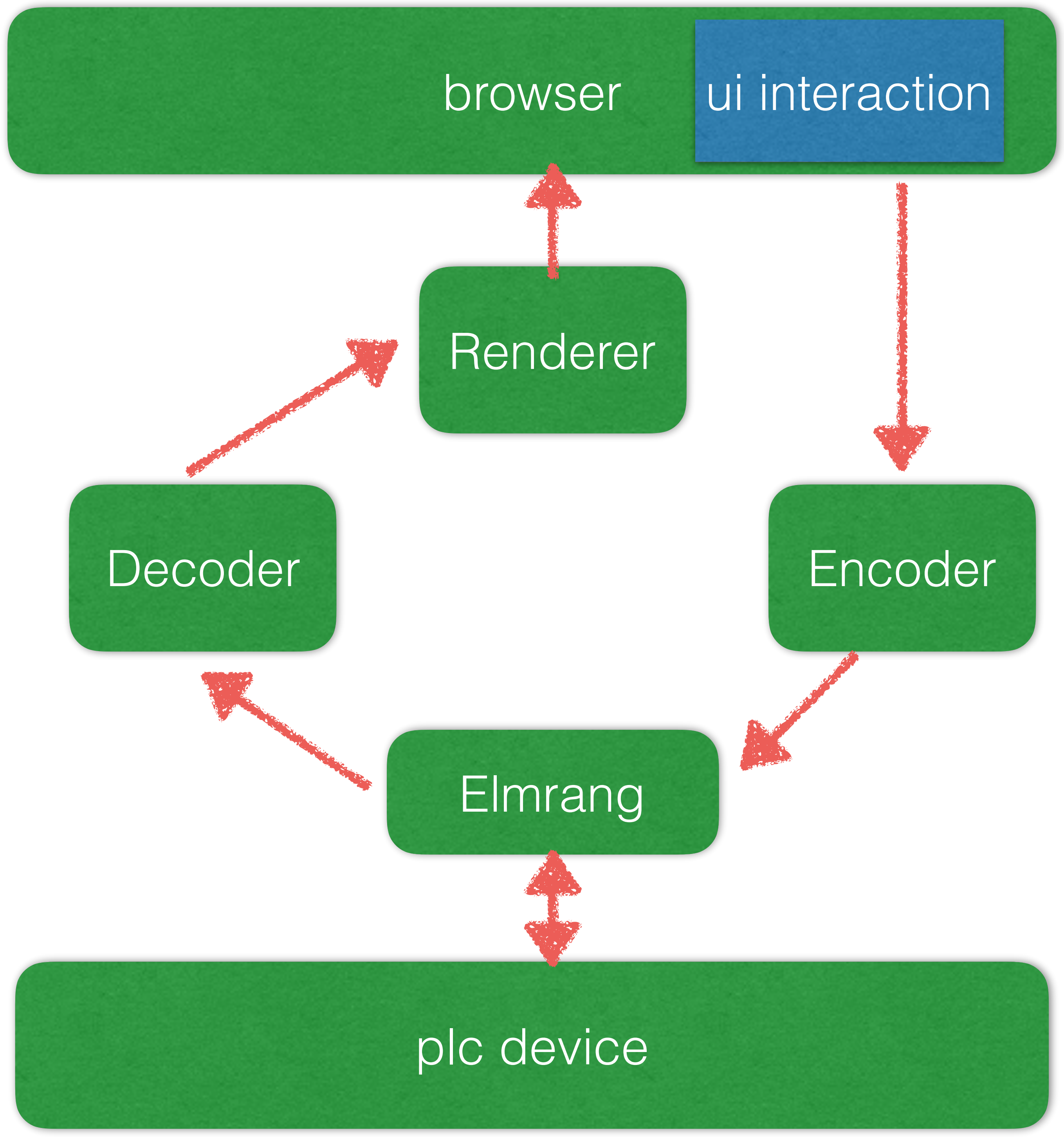
Our Project


PLC IDE
ELMRANG
ports
bullet.js
BROWSER
web sockets
PLC BOARD
PLC Program
cowboy + bullet
BEAM

ERLANG
Embedded With
RTEMS
www.rtems.com
GRiSP

Demo

PLC IDE Structure



What is StartApp?

Implementation of **The Elm Architecture** for **0.16**

In 0.17 it **is** the language

Action **Model** **Update** **View**

Beware: this is different in 0.17

What is StartApp?

Action

```
type Action
= Increment
| Decrement
```

Just a Union Type (aka ADT, etc)

DIPL. PHYS. PEER STRITZINGER GMBH

What is StartApp?

Model

```
type alias Model = Int
```

A type alias

DIPL. PHYS. PEER STRITZINGER GMBH

What is StartApp?

Update

```
update : Action -> Model -> Model
update action model =
  case action of
    Increment -> model + 1
    Decrement -> model - 1
```

Returns the new model state

What is StartApp?

View

```
view : Address -> Model -> Html
view address model =
  p [] [text model]
```

Returns html

DIPL. PHYS. PEER STRITZINGER GMBH

PLC IDE Structure

Four **StartApp** connected by **Mailboxes**

Wired into a parent StartApp, so nested StartApps

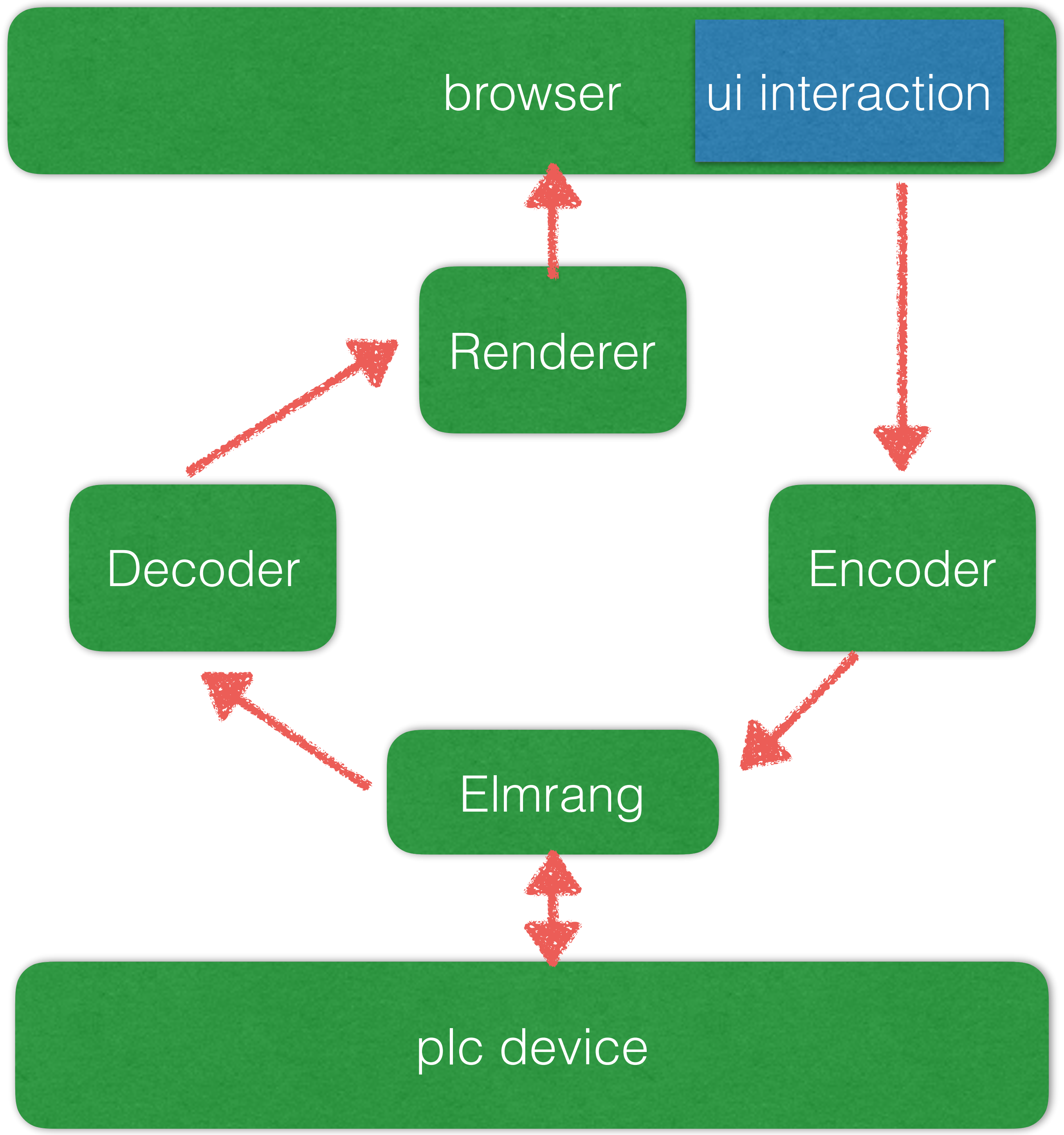
As in the structure invented by **foxdonut**

Easy to expand, add components

But no one ported it to 0.17 (may be impossible)

Elmrang can be a component using this structure

PLC IDE Structure



DIPL. PHYS. PEER STRITZINGER GMBH

Why are we still on 0.16?

We use **FRP** heavily

Porting code might not be **cost effective**

Frustrated with **lack of communication**
(e.g. no deprecation warnings)

Waiting for Elm evolution to **stabilise**

Elmrang

(casualty of the FRP wars)

is a **websocket** library mostly in Elm

it wraps the **bullet** library (for cowboy) using Elm **ports**

includes **javascript code**, so elm-package won't accept it

we were meant to open source it

BUT

it relies on our app's **structure**

0.17 has got socket **anyways**

so, ¯\_(ツ)_/¯

DIPL. PHYS. PEER STRITZINGER GMBH

Why Elmrang?

Once upon a time…

no working **websockets** in Elm

wanted to **use only ports**,

not javascript wrapping

Production Problems

How to organise subcomponents in a big Elm app?

How to store deps not on elm-package?

How to include an Elm project into an Erlang app?

The file structure

Every component has:

```
component/Action.elm
component/Model.elm
component/View.elm
component/Update.elm
component/Feature.elm
```

Wired in in App.elm and fed to Main.elm

DIPL. PHYS. PEER STRITZINGER GMBH

Non elm-package deps

- fetch it from repo
- store it in a subdir of the erlang project
- move only the elm files to a subdir of the elm project
- not under elm-stuff
- include the subdir in elm-package.json

DIPL. PHYS. PEER STRITZINGER GMBH

Mixed Elm/Erlang Project

- /elm subdir in Erlang project
- compiler Elm files to /priv
- add the .js to your html file

Rendering

Choices we had:

- WebGL (2d rendering engine)
- SVG (w or w/o CSS layout and animations)
- Html (not ideal)

Rendering

We use **Svg with CSS**

We try to do as much as we can with CSS

Animation in Elm can get complicated

CSS styles are in separate CSS files

We have an Svg & CSS expert on call

Rendering

elm-html and **elm-svg** have great syntax:

```
div [class “somecssclass”]
    [ p [] [text “a very well written paragraph”]
    , p [] [text “and another one”]
    ]
```

Based on virtualdom = fast

DIPL. PHYS. PEER STRITZINGER GMBH

Several words to the wise

Be aware of what Elm is good for.

An Elm program has to fit the Elm Architecture (which is good if it does fits, less if it doesn't)

Native modules

There is no path to get a library that wraps a javascript library on elm-package (e.g. elm-d3)

Several words to the wise

Elm is still experimental

Elm is still subject to big changes, expect to have to rewrite some of your code with a new version.

Elm lacks a roadmap

There are short beta previews, and you can keep up by looking at the changes in the compiler.
Recently Evan started doing semi-regular updates of what he’s up to in the mailing list

What next?

We're going to skip 0.17

Maybe come back when Elm is nearer to 1.0

Meanwhile taking Purescript for a spin
and Clojurescript is on the list, too

What is Purescript?

Pure Functional

Strongly Typed

Eagerly evaluated

Compiles to Javascript

Haskell-like syntax (with all the squiggles)

Generates readable Javascript, has no runtime

Advanced Types

Open community, a bit of a roadmap

DIPL. PHYS. PEER STRITZINGER GMBH

Why Purescript next?

The advantages of types in Elm were great

Elm stops at typeclasses, but the ceiling is much higher

Pragmatic reasons, it works, it's possible to implement Elm in it, but not the other way around

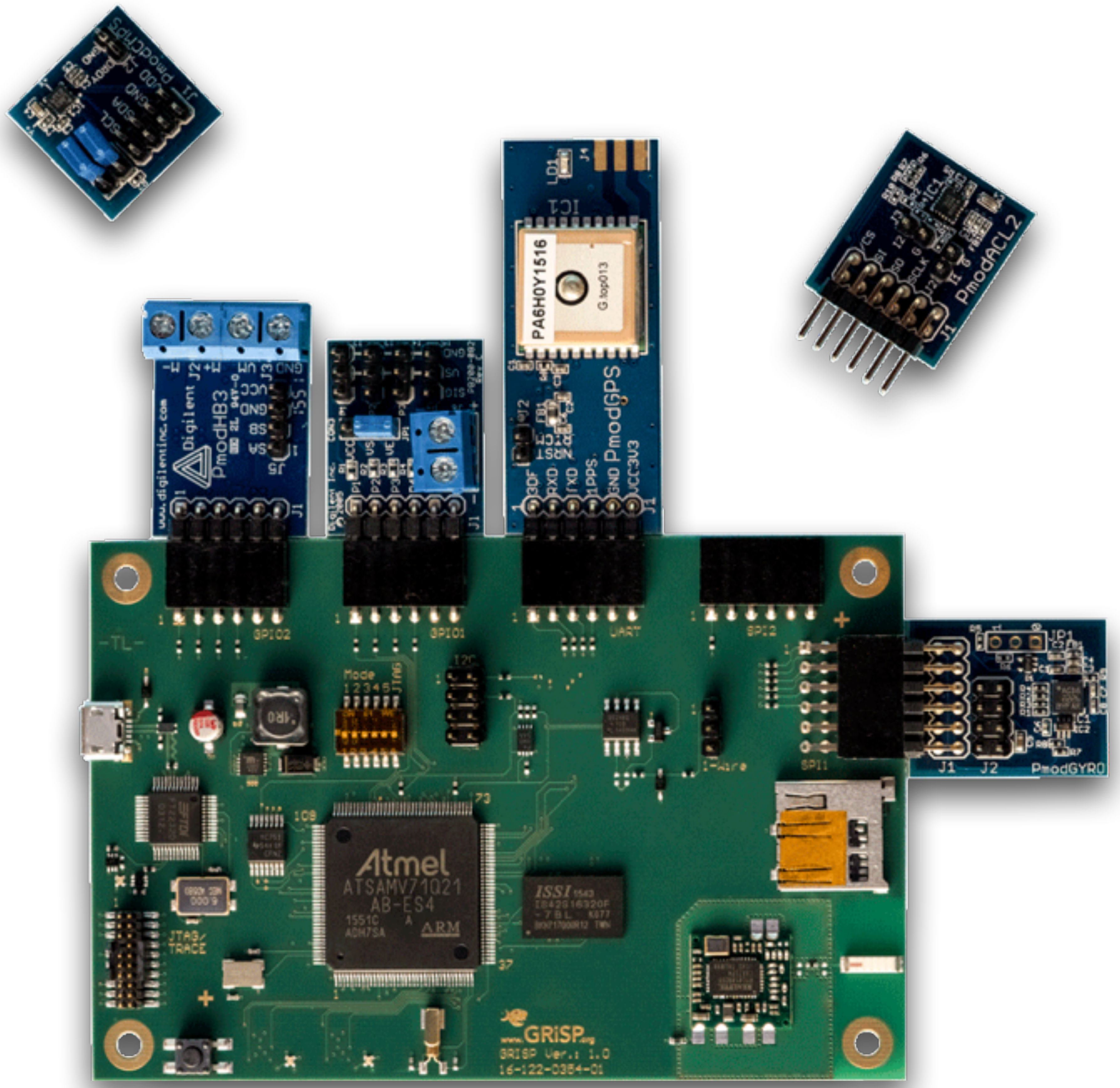
Small, open community, communication still works

Fun!

DIPL. PHYS. PEER STRITZINGER GMBH

tl;dr

Elm works fine with Erlang
If Elm compiles, it works (mostly)
boilerplate can get annoying
never expect fancy types
Haskell syntax (with less squiggles)
unexpected removal of FRP was :/



DIPL. PHYS. PEER STRITZINGER GMBH

Win One of 10 Boards by subscribing to the Newsletter until September 15th

GRiSP

www.grisp.org

DIPL. PHYS.
PEER STRITZINGER GMBH

Questions?

DIPL. PHYS. PEER STRITZINGER GMBH