

A Little Closer to Frontend Bliss with Elm 

# Who am I?

 terezka

 Copenhagen, Denmark

 Frontend developer @ Opbeat

 React.js going Elm

 Elm-plot

# Agenda



Today's developer experience



Elm to the rescue



Great, so what do I do?

Part I: Today's developer experience



# JavaScript!





# JavaScript



You can do whatever you want!



# JavaScript

- 😊 You can do whatever you want!
- 😢 You can do whatever you want.



# JavaScript

- 😊 You can do whatever you want!
- 😐 You can do whatever you want.
- 😦 Everyone can do whatever they want.



- map-async-to-generator
- place-supers
- s
- check-es2015-constants
- syntax-async-functions
- syntax-class-constructor-call
- syntax-class-properties
- syntax-decorators
- syntax-exponentiation-operator
- syntax-export-extensions
- syntax-object-rest-spread
- syntax-trailing-function-commas
- transform-async-to-generator
- transform-class-constructor-call
- transform-class-properties
- transform-decorators
- transform-es3-member-expression-literals
- transform-es3-property-literals
- transform-es2015-arrow-functions
- transform-es2015-block-scoped-functions
- transform-es2015-block-scoping
- transform-es2015-classes
- transform-es2015-computed-properties
- transform-es2015-destructuring
- transform-es2015-for-of
- transform-es2015-function-name
- transform-es2015-literals
- transform-es2015-modules-commonjs
- transform-es2015-object-super
- transform-es2015-parameters
- transform-es2015-shorthand-properties
- transform-es2015-spread
- transform-es2015-sticky-regex
- transform-es2015-template-literals
- > babel-helper-remap-async-to-generator
- > babel-helper-replace-supers
- > babel-helpers
- > babel-loader
- > babel-messages
- > babel-plugin-check-es2015-constants
- > babel-plugin-syntax-async-functions
- > babel-plugin-syntax-class-constructor-call
- > babel-plugin-syntax-class-properties
- > babel-plugin-syntax-decorators
- > babel-plugin-syntax-exponentiation-operator
- > babel-plugin-syntax-export-extensions
- > babel-plugin-syntax-object-rest-spread
- > babel-plugin-syntax-trailing-function-commas
- > babel-plugin-transform-async-to-generator
- > babel-plugin-transform-class-constructor-call
- > babel-plugin-transform-class-properties
- > babel-plugin-transform-decorators
- > babel-plugin-transform-es3-member-expression-literals
- > babel-plugin-transform-es3-property-literals
- > babel-plugin-transform-es2015-arrow-functions
- > babel-plugin-transform-es2015-block-scoped-functions
- > babel-plugin-transform-es2015-block-scoping
- > babel-plugin-transform-es2015-classes
- > babel-plugin-transform-es2015-computed-properties
- > babel-plugin-transform-es2015-destructuring
- > babel-plugin-transform-es2015-for-of
- > babel-plugin-transform-es2015-function-name
- > babel-plugin-transform-es2015-literals
- > babel-plugin-transform-es2015-modules-commonjs
- > babel-plugin-transform-es2015-object-super
- > babel-plugin-transform-es2015-parameters
- > babel-plugin-transform-es2015-shorthand-properties
- > babel-plugin-transform-es2015-spread
- > babel-plugin-transform-es2015-sticky-regex
- > babel-plugin-transform-es2015-template-literals
- > babel-helper-remap-async-to-generator
- > babel-helper-replace-supers
- > babel-helpers
- > babel-loader
- > babel-messages
- > babel-plugin-check-es2015-constants
- > babel-plugin-syntax-async-functions
- > babel-plugin-syntax-class-constructor-call
- > babel-plugin-syntax-class-properties
- > babel-plugin-syntax-decorators
- > babel-plugin-syntax-export-extensions
- > babel-plugin-syntax-object-rest-spread
- > babel-plugin-syntax-trailing-function-commas
- > babel-plugin-transform-async-to-generator
- > babel-plugin-transform-class-constructor-call
- > babel-plugin-transform-class-properties
- > babel-plugin-transform-decorators
- > babel-plugin-transform-es3-member-expression-literals
- > babel-plugin-transform-es3-property-literals
- > babel-plugin-transform-es2015-arrow-functions
- > babel-plugin-transform-es2015-block-scoped-functions
- > babel-plugin-transform-es2015-block-scoping
- > babel-plugin-transform-es2015-classes
- > babel-plugin-transform-es2015-computed-properties
- > babel-plugin-transform-es2015-destructuring
- > babel-plugin-transform-es2015-for-of
- > babel-plugin-transform-es2015-function-name
- > babel-plugin-transform-es2015-literals
- > babel-plugin-transform-es2015-modules-commonjs
- > babel-plugin-transform-es2015-object-super
- > babel-plugin-transform-es2015-parameters
- > babel-plugin-transform-es2015-shorthand-properties
- > babel-plugin-transform-es2015-spread
- > babel-plugin-transform-es2015-sticky-regex
- > babel-plugin-transform-es2015-template-literals



# JavaScript: Theoretically, yes

Unfortunate circumstances 😞

# Unfortunate circumstances

- 😔 Pressure from deadline
- 🗣 Bad management
- 👨‍👩‍👧 Lack of team communication
- 🍼 New on the team

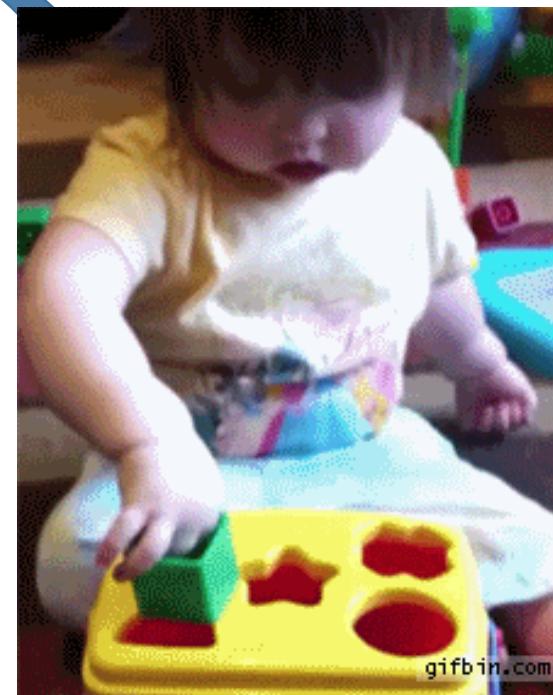


...

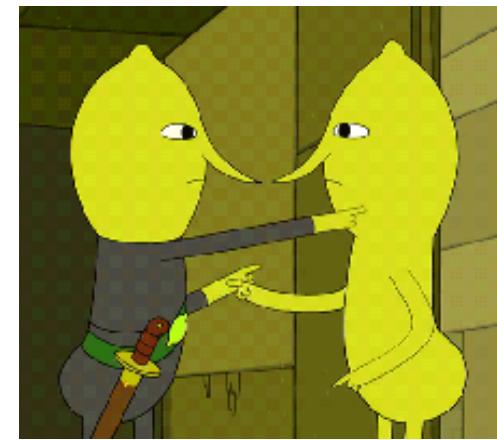
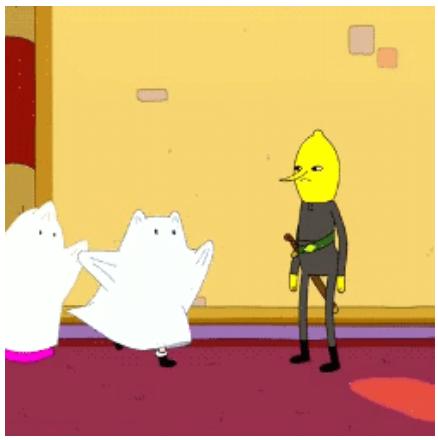
- 💔 Simply a bad day

# I'll fix it later

```
✖ ▾ Uncaught TypeError: undefined is not a function
  Ext.cmd.derive.applyTitle
  j
  b.implement.setConfig
  Ext.cmd.derive.show
  Ext.cmd.derive.alert
  Ext.cmd.derive.loginClick
```



# Emotional consequence



*JavaScript Fatigue* 💀

## Part II: Elm to the rescue



Constrains



Only pure functions 💎

# Pure and impure functions

// Pure function 💎

```
function addOne(x) {  
    return x + 1;  
}
```

// Impure function 💥

```
function addOne(x) {  
    x++;  
    fireMissiles();  
}
```

# Pure functions and immutability

// JavaScript ✨

// (anything goes)

var x = { foo: "bar" }

var y = f(x)

-- Elm 🎈

-- (only pure functions)

x = { foo = "bar" }

y = f x

What is x? ↴

x is *always* unchanged

Static type system



# Static types and type inference

```
-- With function signature 🐾  
horizontalGrid : List GridAttr -> PlotElement msg  
horizontalGrid attrs =  
    Grid (List.foldr (<|) defaultGridConfig attrs)
```

```
-- Without function signature 🚶  
horizontalGrid attrs =  
    Grid (List.foldr (<|) defaultGridConfig attrs)
```

No run time errors! 

# No run time errors

The program 

```
dostoevsky =  
{ firstName = "Fyodor"  
, lastName = "Dostoevsky"  
}  
  
greet person =  
"Hello " ++ person.firstname  
  
main =  
text (greet dostoevsky)
```

The compiler 

```
-- TYPE MISMATCH -----  
  
The argument to function `greet` is causing a mismatch.  
15|     greet dostoevsky)  
          ^^^^^^^^^^  
  
Function `greet` is expecting the argument to be:  
  
  { a | firstname : ... }  
  
But it is:  
  
  { firstName : ..., lastName : ... }  
  
Hint: I compared the record fields and found some potential typos.  
      firstname <-> firstName
```

# No run time errors? For real?



Enforced semantic versioning



Improved performance 🚀

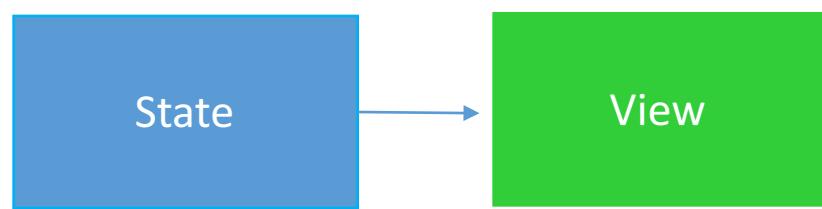
Architecture? 

# Architecture

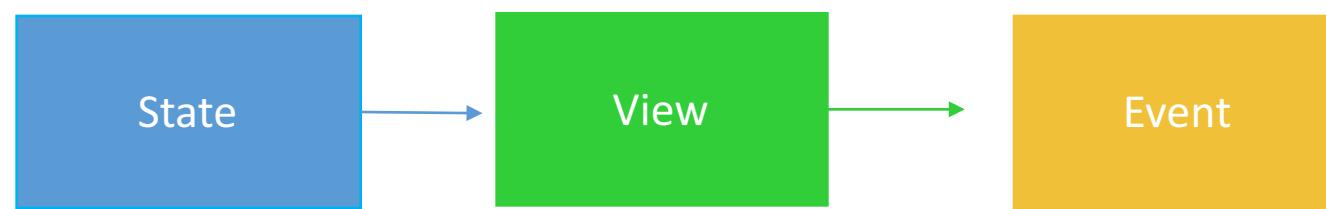


State

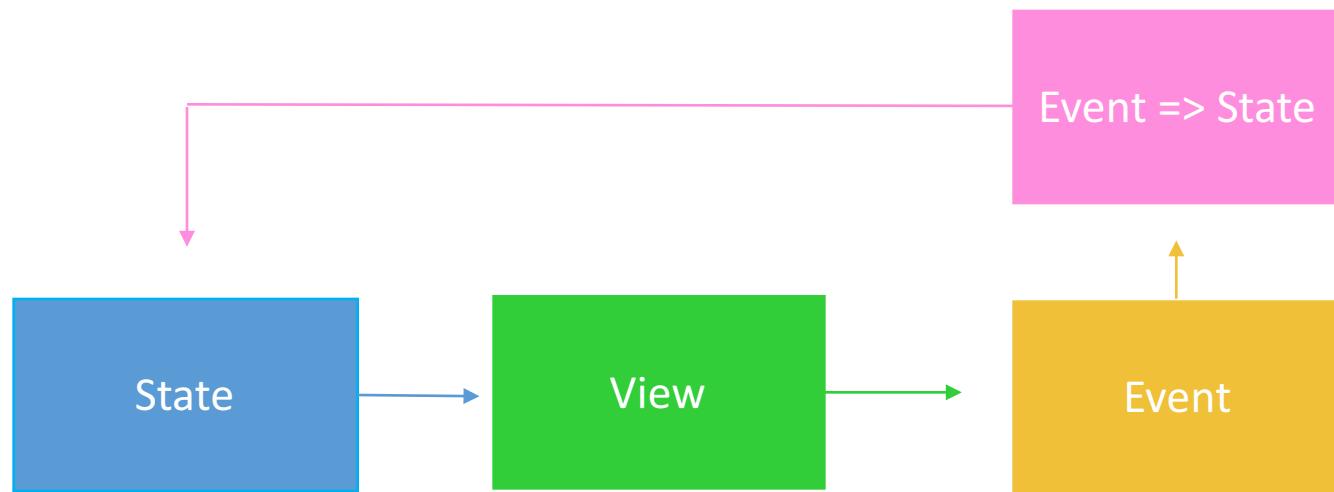
# Architecture



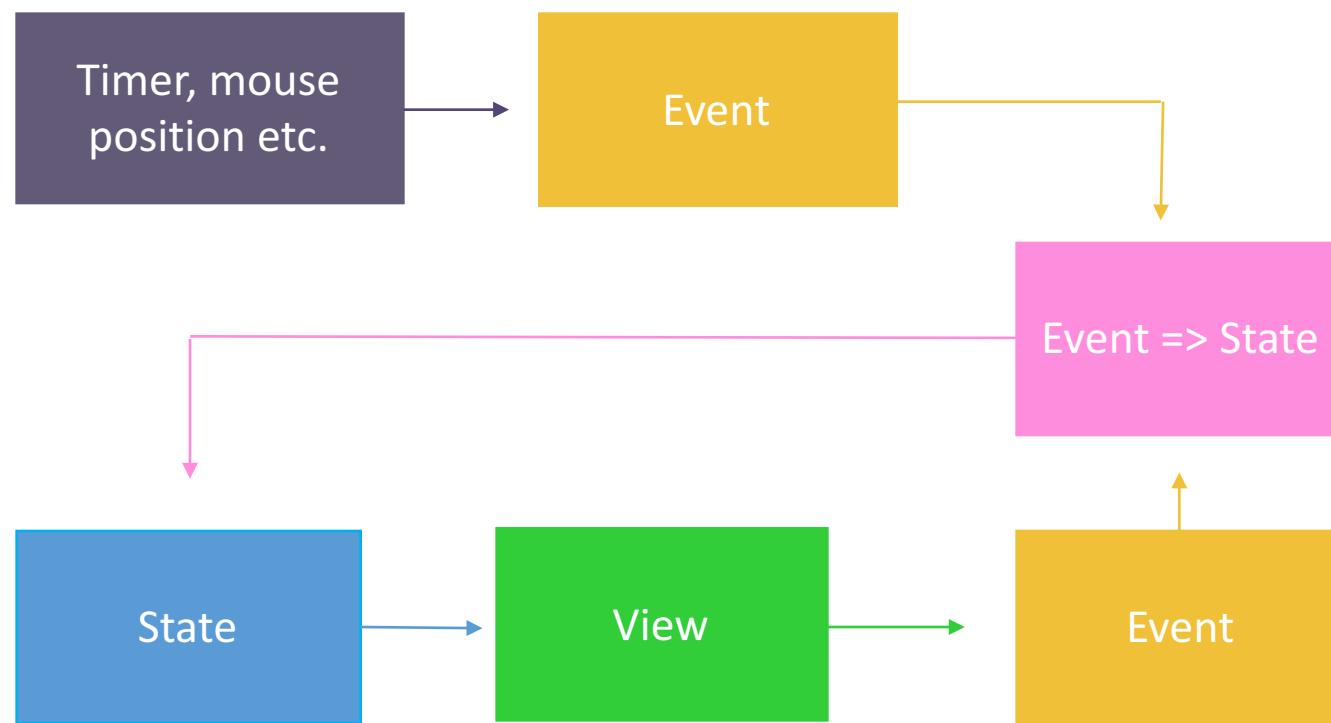
# Architecture



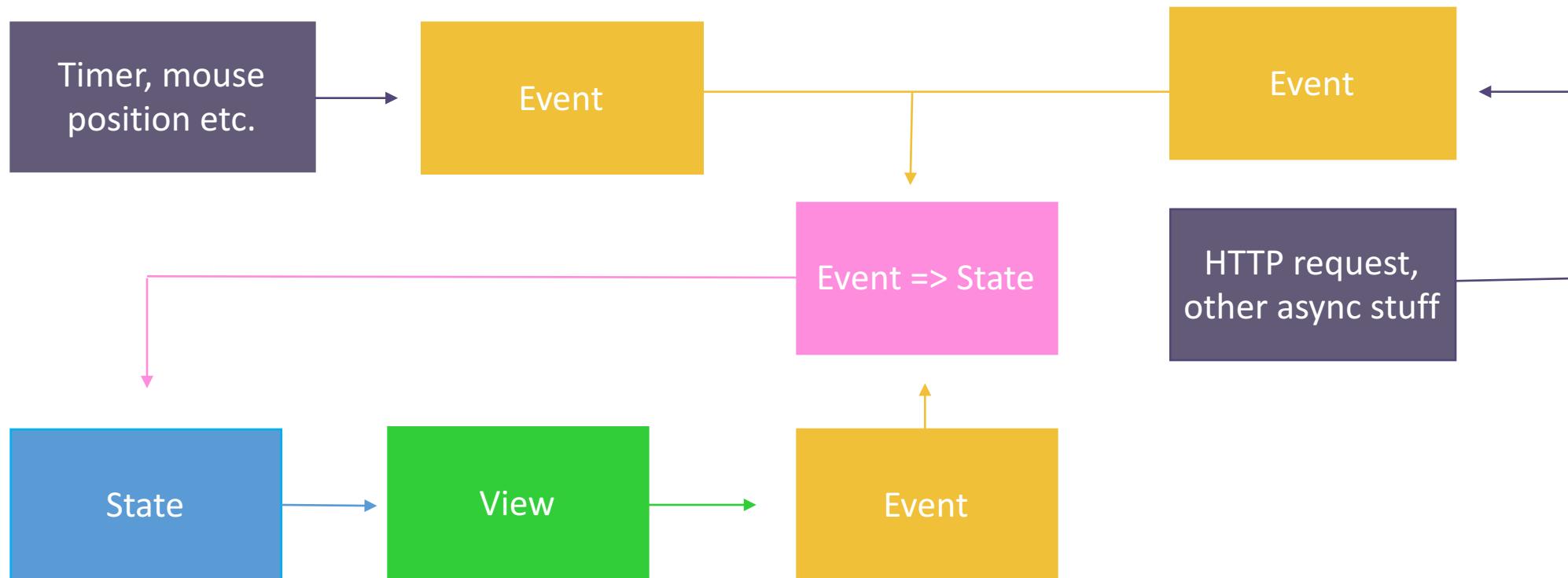
# Architecture



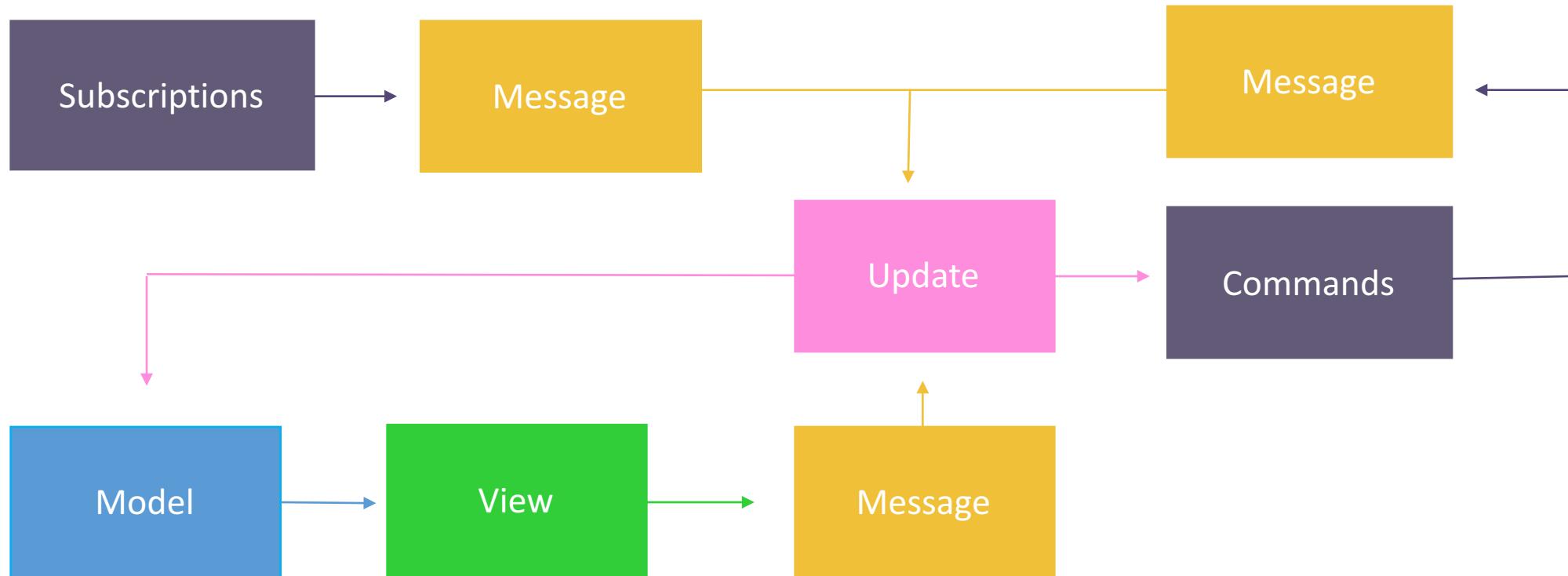
# Architecture



# Architecture



# The Elm architecture



Time traveling debugger! 

# Constrains

## Rules

-  Only pure functions
-  Static type system
-  Opinionated on architecture

## Benefits

-  No run time errors!
-  Enforced semantic versioning
-  Improved performance
-  Time traveling debugger!

Do what you do best



# JavaScript interop



Show time 

Part III: Great, so what do I do? ☀

Gradual learning 

# Gradual implementation



Helpful community (#beginners!) 

Try it out ❤

Q & A



# Contact

-  Tereza Sokol
-  Opbeat
-  terezasokol@gmail.com
-  terexka
-  terezka