



A PEEK IN TO ELM ARCHITECTURE

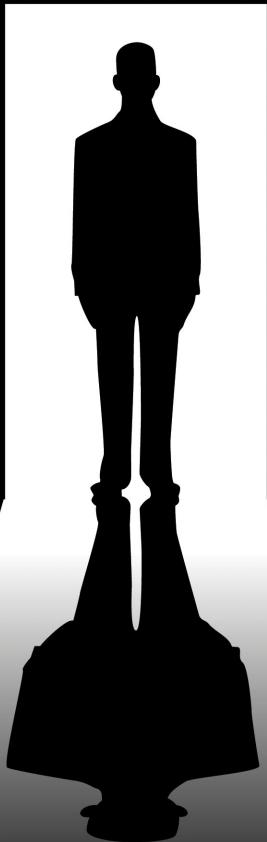
Functional Programming For the Web



@jayaramsankara

FunctionalConf 2017

18-Nov-2017



jayaramsankara



ABOUT ME

Architect @ cisco

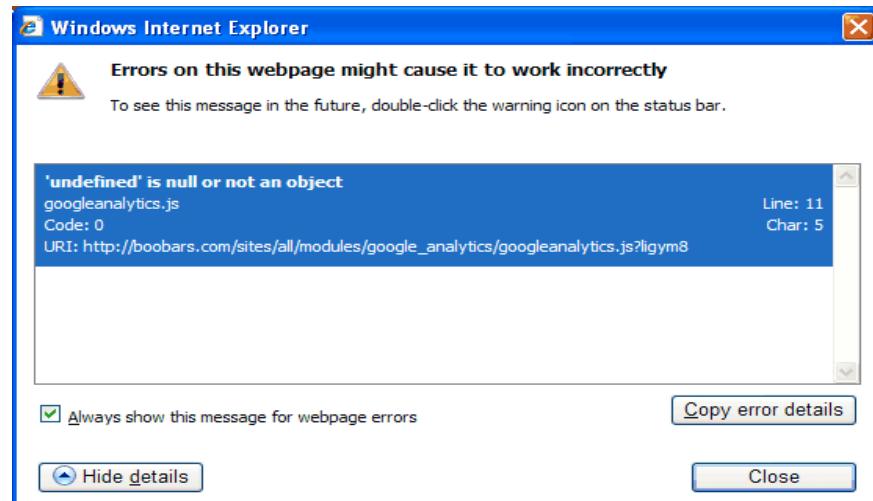
Have been doing imperative, procedural and OO for first decade and half of my career

Introduced to Functional Programming few years back and have been building solutions using technologies and programming languages that enables use of functional principles

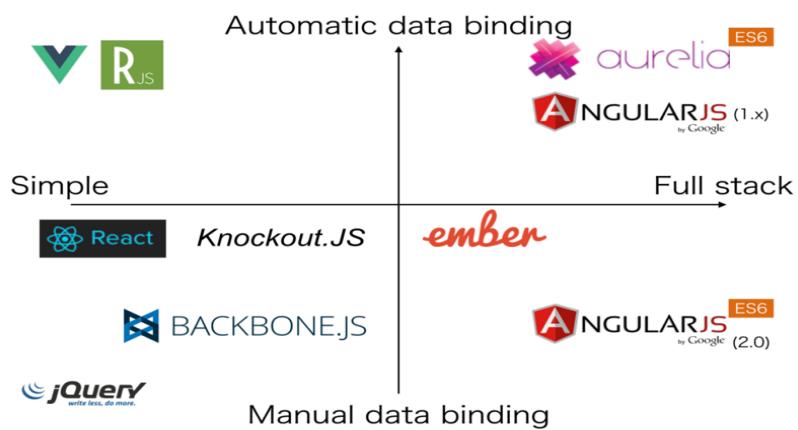
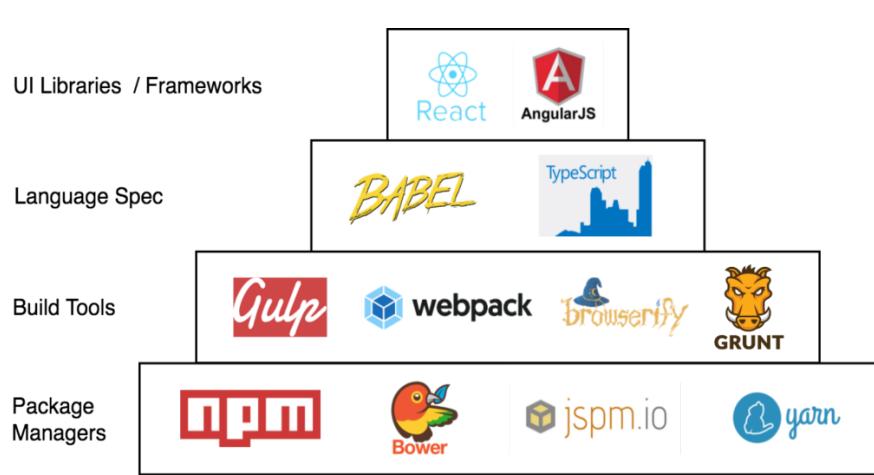
Now passionate about helping teams adopt the practices and change the way software is designed and developed for solving real world problems



UI DEVELOPMENT



DECIDED ON A JAVASCRIPT FRAMEWORK





source: https://jeremybutterfield.files.wordpress.com/2016/01/peak_take-a-peek.jpeg



elm



WHAT TO EXPECT?

Recap

Basic Functional
Programming
Principles

Functions

Purity

Immutability

Partial Application

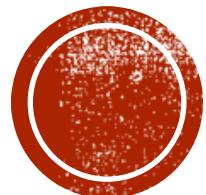
Quick Tour :
UI to send /
receive
notifications

Basic Syntax

Elm Architecture

Doing Stuff i.e. Performing Effects





FP QUICK RECAP

Functional Programming Basics

RECAP : BASIC FUNCTIONAL PROGRAMMING

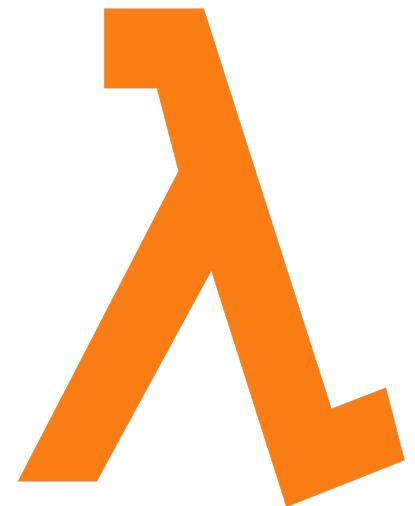
Pure Functions

- No Side Effects
- Referential Transparency

Immutable Data Structures

Expressive Power

- Ability to Reason - Equational Reasoning
- Function Composition
 - $f: b \rightarrow c, g: a \rightarrow b$
 - $f \circ g : a \rightarrow c$
 - $f \circ g = f(g(x))$



RECAP :PROGRAMMING BY WHOLE VALUES

List.map (a ->b) -> List a -> List b

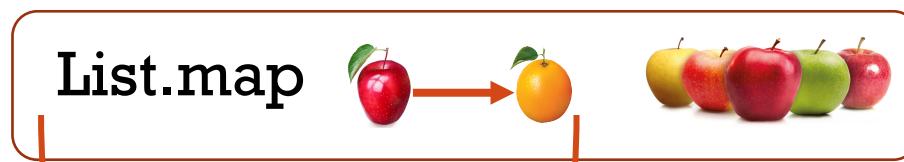
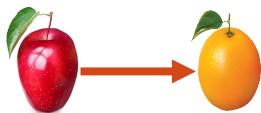
List.filter (a ->Bool) -> List a -> List a

List.foldr (a -> b -> b) -> b -> List
a -> b

Maybe.map (a -> b) Maybe a -> Maybe
b



RECAP :PARTIAL APPLICATION



apples2oranges = List.map





elm



SYNTAX : LITERALS

```
numRegistrations : Int  
numRegistration = 550
```

```
separator : Char  
separator = ;'
```

```
title : String  
title = "Elm Architecture"
```

```
cool : Bool  
cool = True
```

name : type
name = value



SYNTAX : FUNCTIONS

Similar to defining values
name : type

Functions take one parameter at time

```
add : Int -> Int -> Int  
add 1 3 = 1 + 3
```

add : Int -> Int -> Int
add a: Int -> Int
add a b: Int

Function Application:
Parameters are separated by white space



SYNTAX : RECORDS & UNION TYPES

Define Custom
Records

```
type alias Song = {  
    singer : String,  
    album: String  
}
```

Define New Types

```
type Music = Vocals |  
Acoustic | Other String
```

OTHER COOL STUFF

Strong Static Type Check

- Extremely Useful compiler error messages
- In most cases, if it compiles, it will work

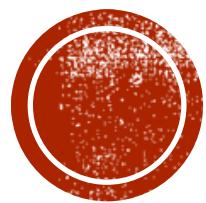
Get rid of module version dependencies

- Elm package manager enforces version semantics
- Breaking backward compatibility forces library developers to bump up major version.

Elm Debugger

- Replay Events and see how the app performs.

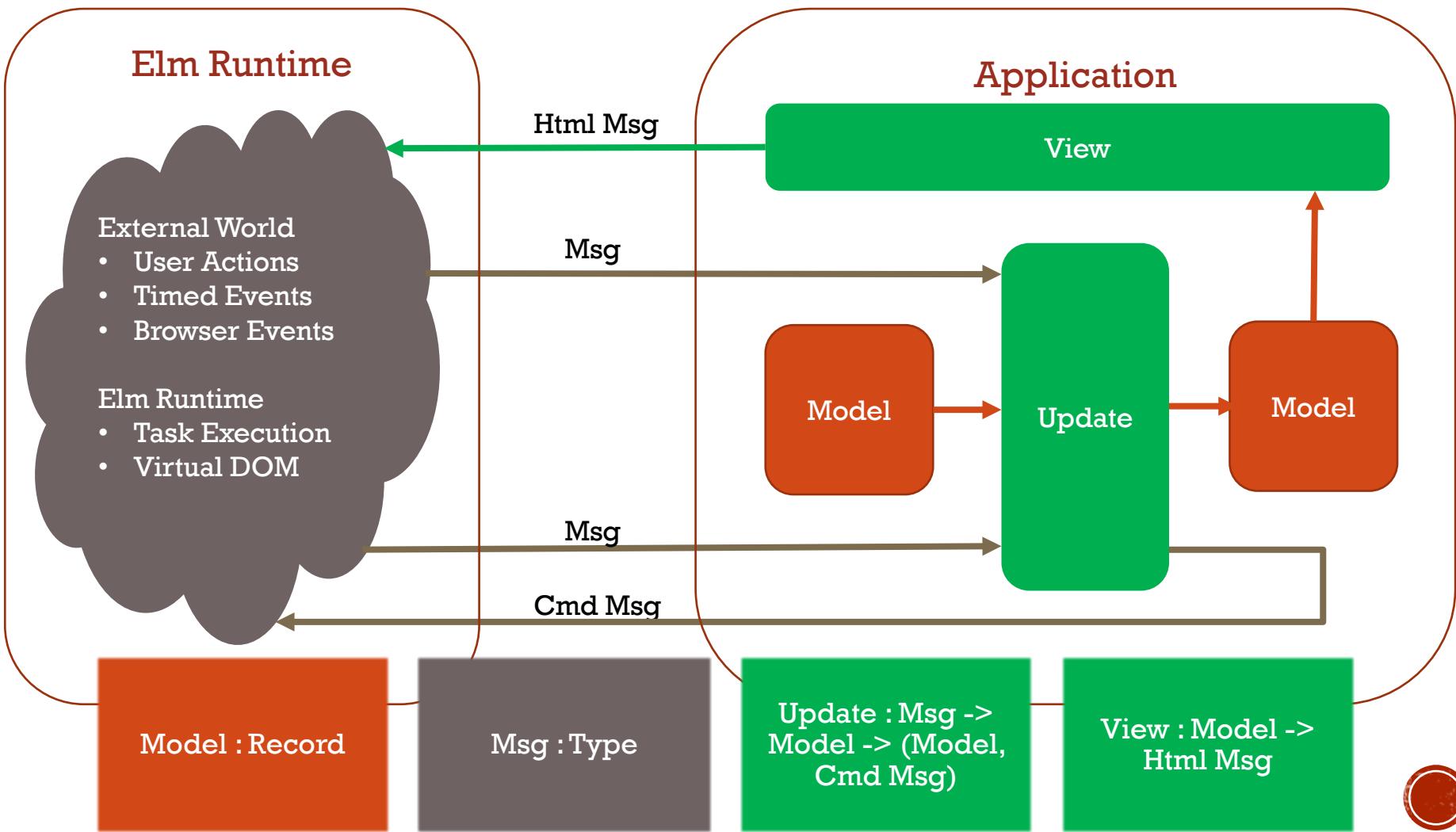


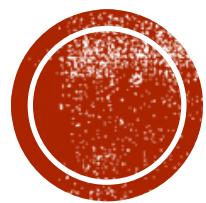


LETS GET IN TO THE ARCHITECTURE

The ELM Architecture

MODEL - VIEW - UPDATE

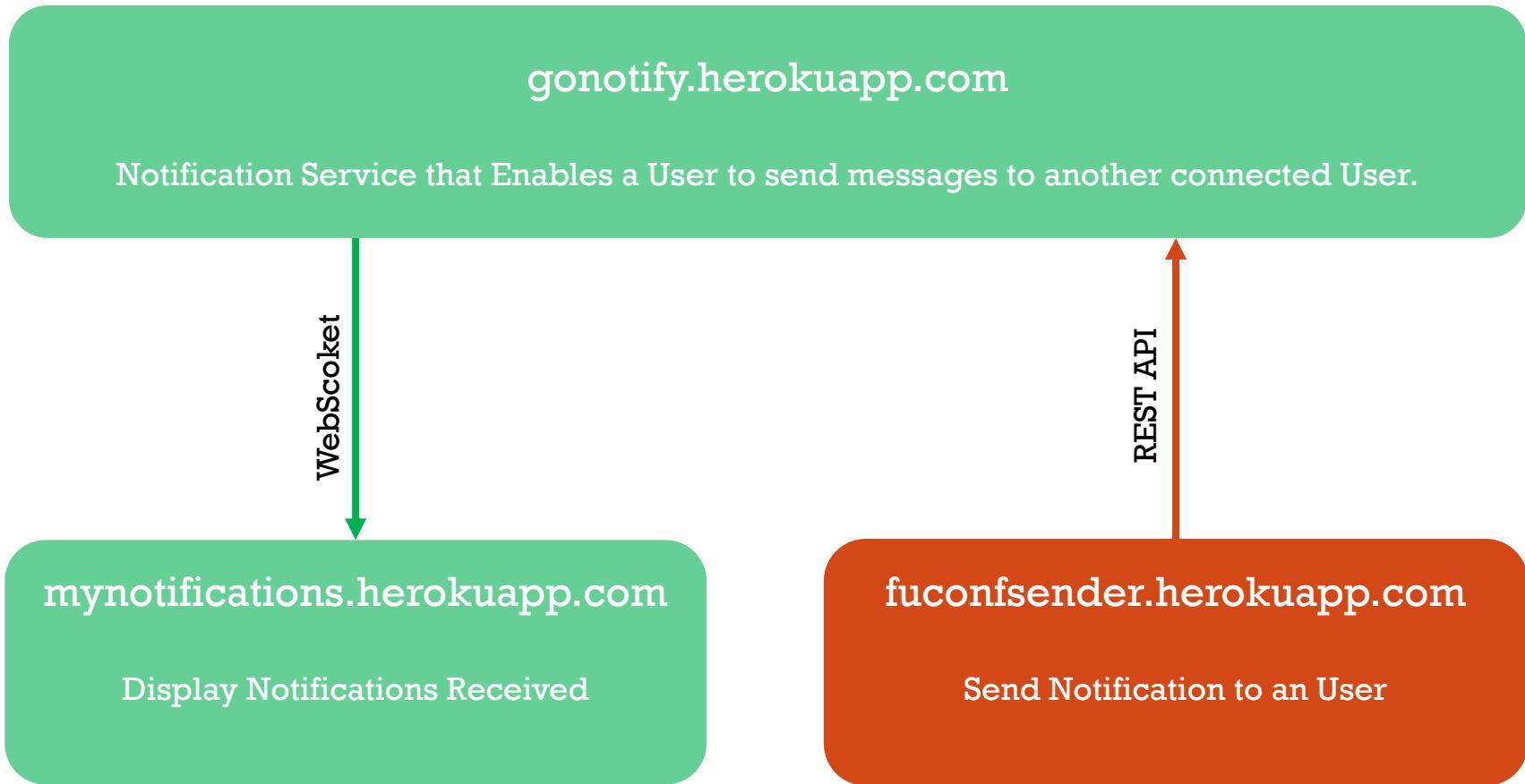




LETS GET OUR HANDS DIRTY

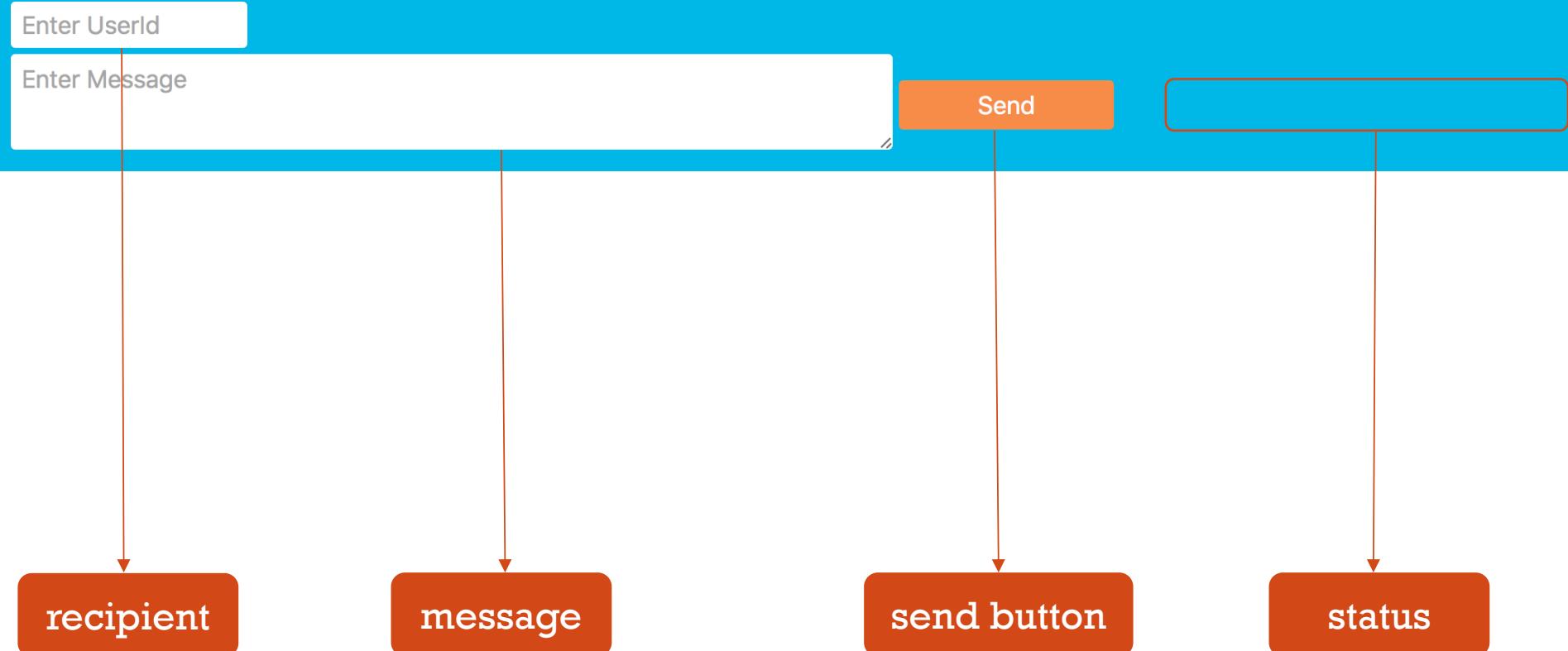
Build an UI to send notifications

DEMO APP: UI TO SEND NOTIFICATION



VIEW

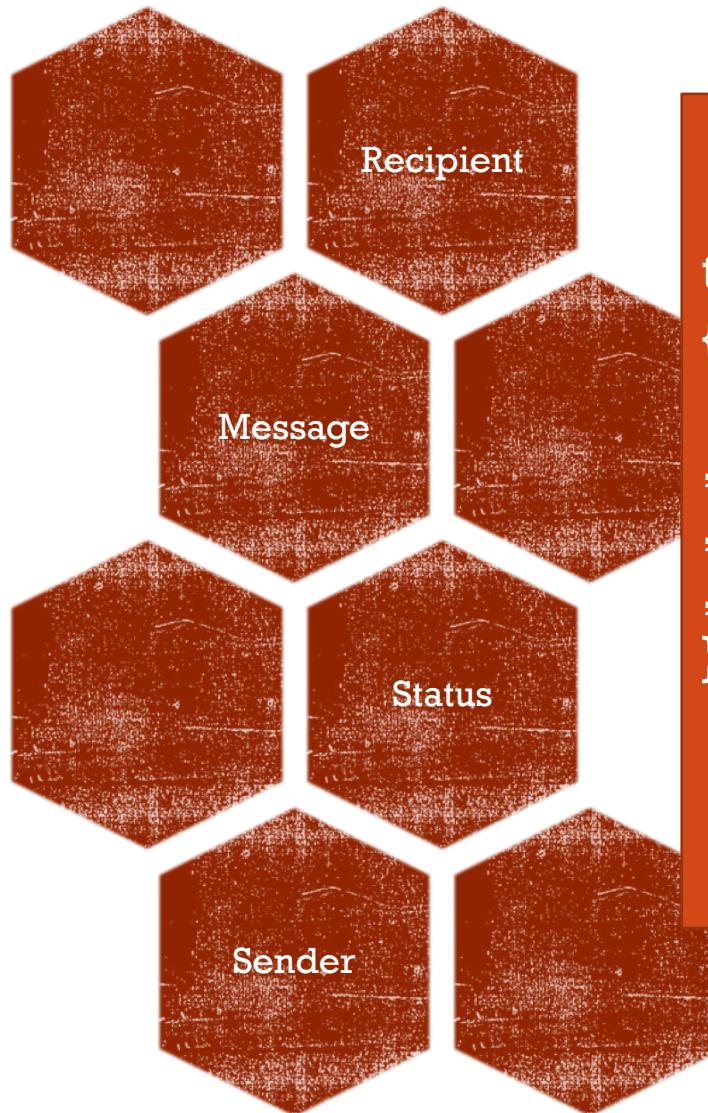
Send Message:



LIVE CODING & DEMO



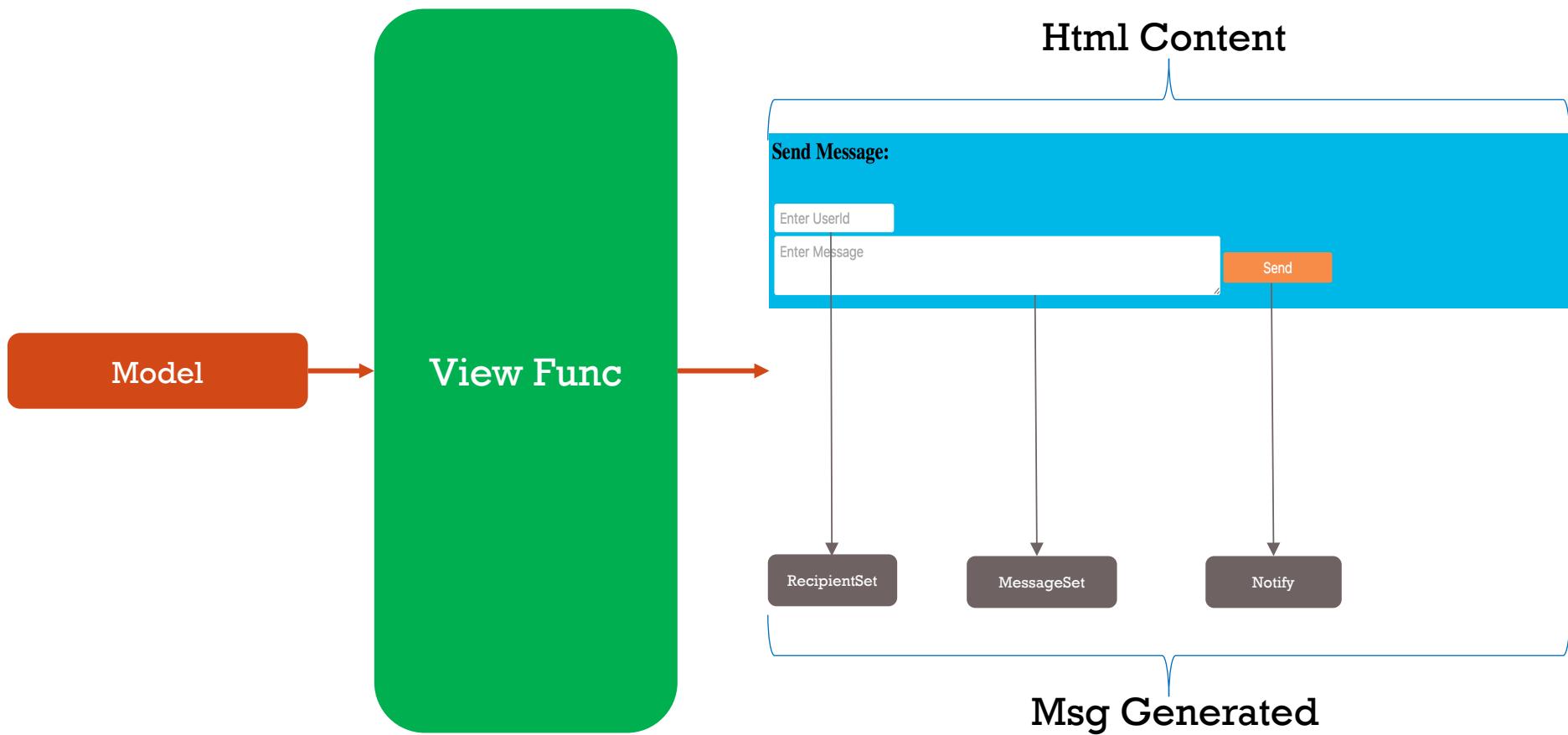
MODEL



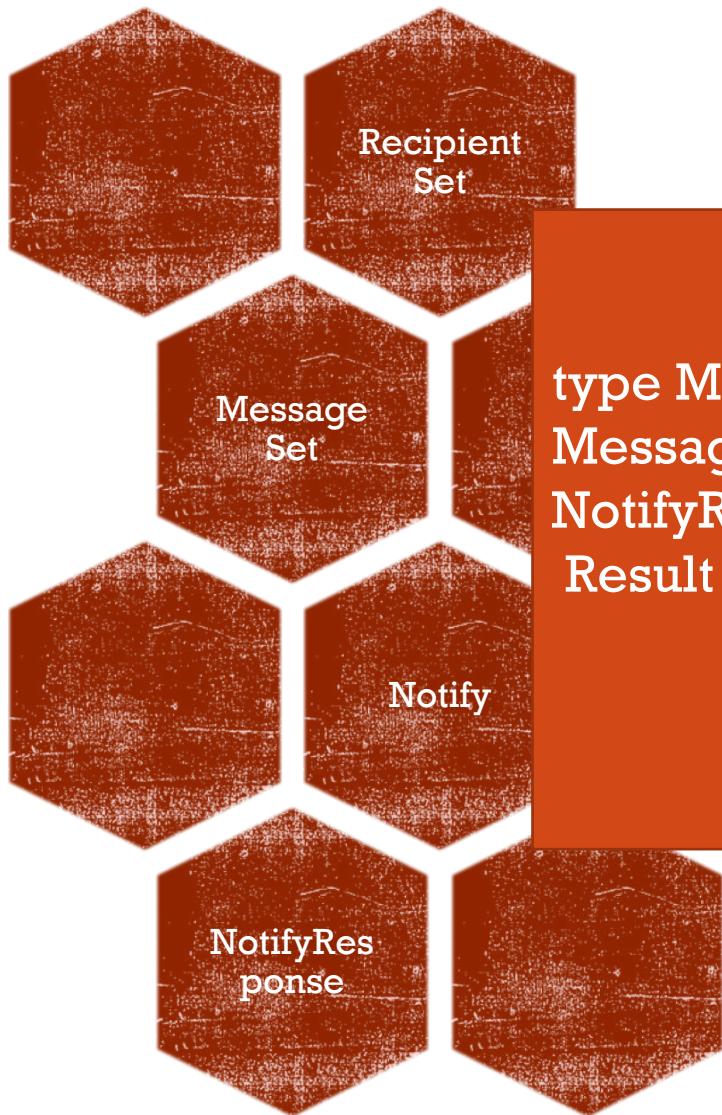
```
type alias Model =  
{  
    recipient : String  
, message : String  
, status : Maybe String  
, sender : String  
}
```



VIEW



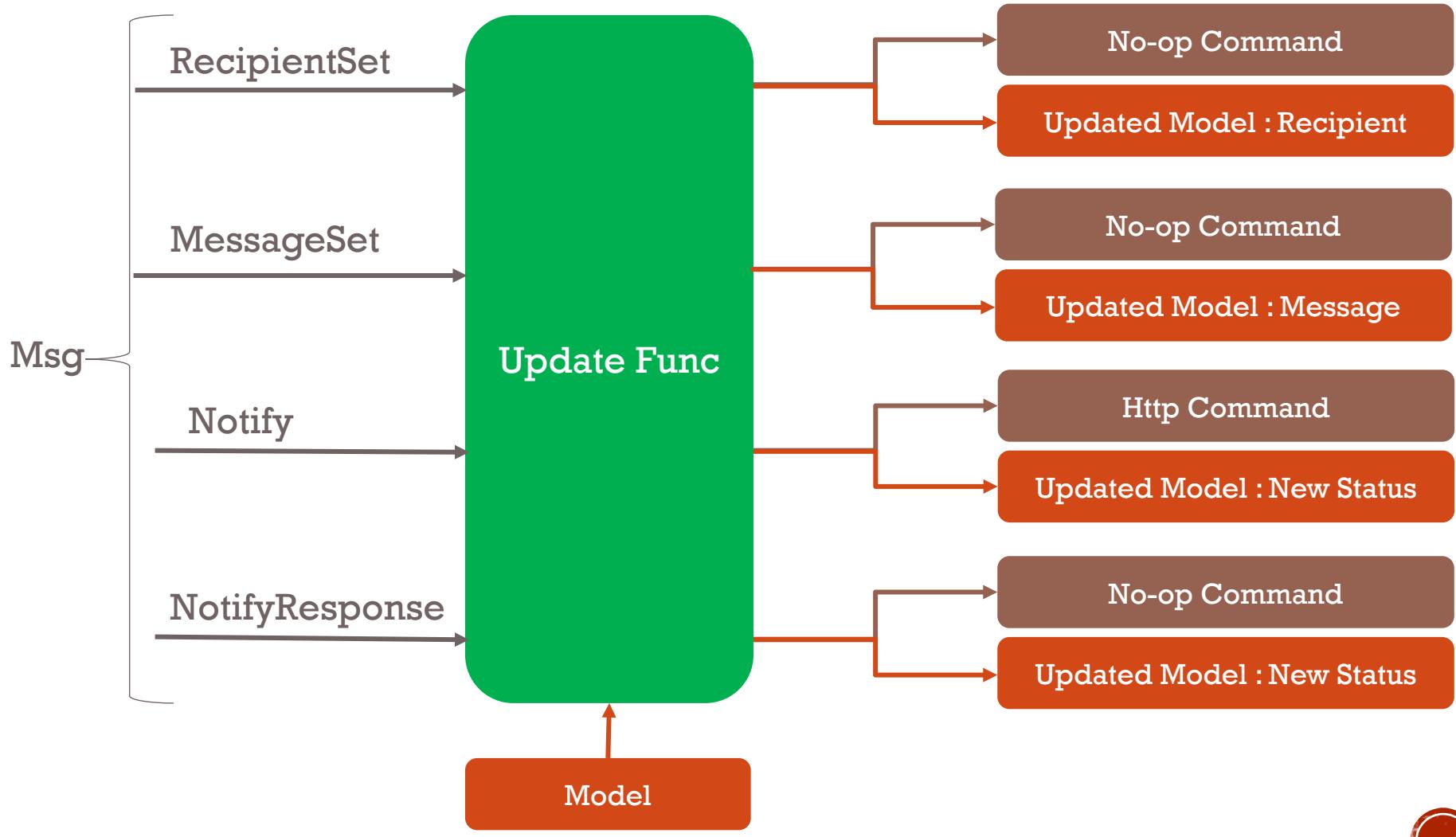
MSG



```
type Msg = RecipientSet String |  
MessageSet String | Notify |  
NotifyResponse  
Result Err NotifyResponseBody
```



UPDATE

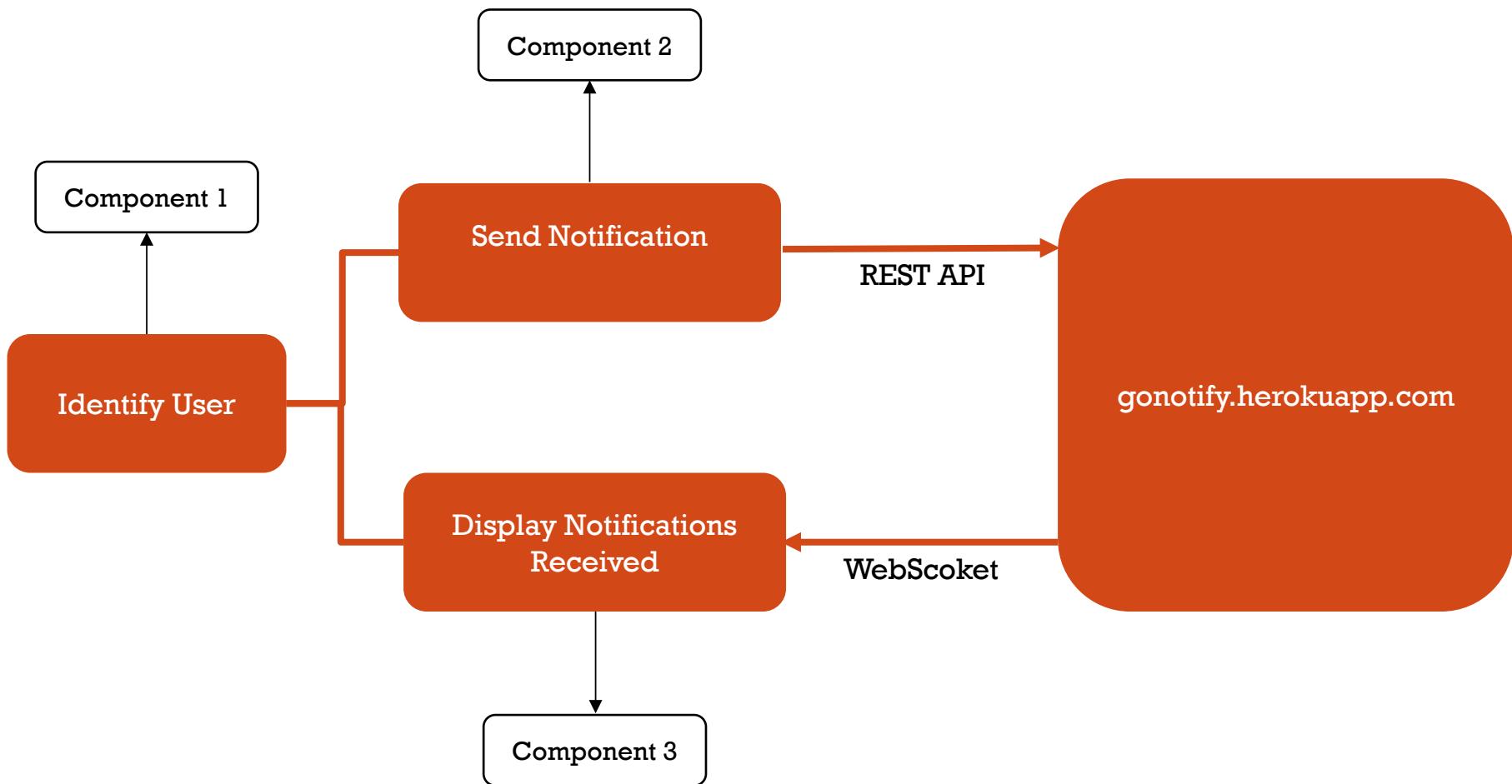




A LOOK AT COMPONENTS

Build and Use Elm Components

A COMPLETE NOTIFICATION UI



Refer <https://github.com/jayaramsankara/client-gonotify> for more details



CODE WALK THRU & DEMO



RECAP : COMBINING ELM COMPONENTS

Combine Msg

- Msg to have a Container for Msg from components

Combine Model

- Model to have fields for Model from components

Delegate View & Update to components

- Map component's Msg to Msg
 - In View function
 - `Html Component.Msg -> Html Msg`
 - In Update function
 - `Cmd Component.Msg -> Cmd Msg`



CODE/API REFERENCES

<https://github.com/jayaramsankara/peek-elm>

<https://github.com/jayaramsankara/client-gonotify>

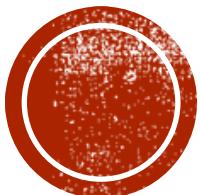
<https://github.com/jayaramsankara/gonotify>



QUESTIONS?

@jayaramsankara





THANK YOU!