elm

New language redesign client-side

MTS@HDE 2014/11/21 CPRD開発部 小本健司

こんばんは！
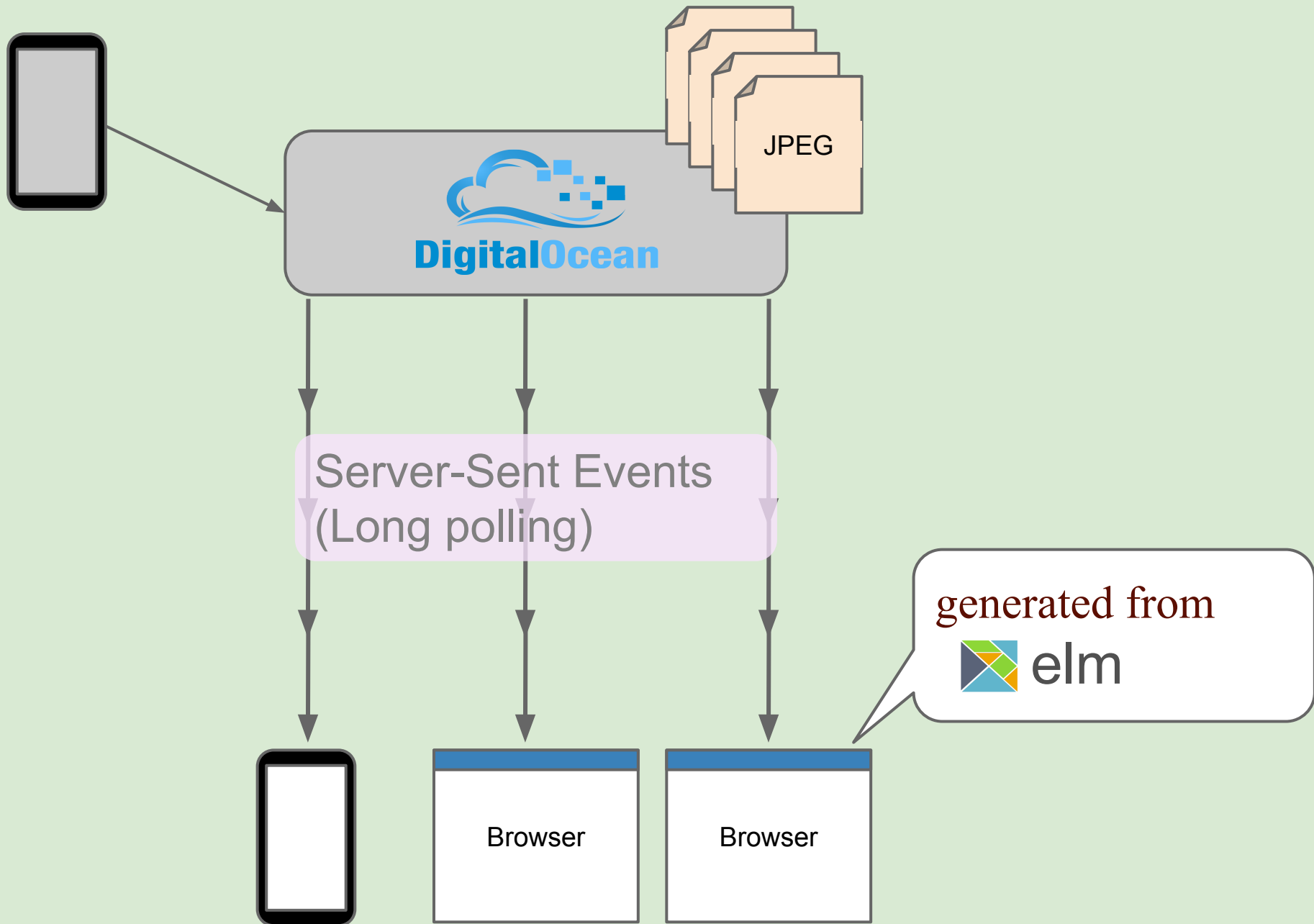
Good evening !!

どうです？動いていますか？

How it is? Is it working ?

動いていることにして話します

I assume it's working

Why **elm** ?

AngularJS is Complex for me

Many different languages

HTML, CSS, JavaScript, Less, CoffeeScript, Slim, etc...

Over-Optimizad for their usage

Less support for drawing...

Much attributes

ngBind, ngResource, ngDirective etc…

What about elm ?

Monolithic language

Layout and style are subject of logic

Simple Haskell based functional language

Everything is function

FRP

What is this ?

```
main : Element
main = scene 100 200

scene:Int -> Int -> Element
scene x y = flow right
  [
    plainText (show x)
  , plainText (show y)
  ]
```

- scene create static text from static Int

```
main : Signal Element
main = lift2 scene Mouse.x Mouse.y

scene:Int -> Int -> Element
scene x y = flow right
  [
    plainText (show x)
  , plainText (show y)
  ]
```

- `scene` create static text from static Int

- `lift2` convert `scene` to accept Signal and to create Signal'd Element

- Then, `scene` is automatically updated

Signal is versatile

```
Mouse.x
    Signal Int

Window.dimension
    Signal (Int, Int)

Http.send
    Signal(Request a)->Signal(Response String)

Time.fps
    number -> Signal Time
```

Most functional languages are too theoristic

Type, Macro, Monad etc... didn't exicite working programers
e.g. F#, Haxe, D etc…

But, elm also has **realistic** advantages

FRP (better than AngularJS)

Layout and style libraries

Blazing Fast elm-html

Better for Game?
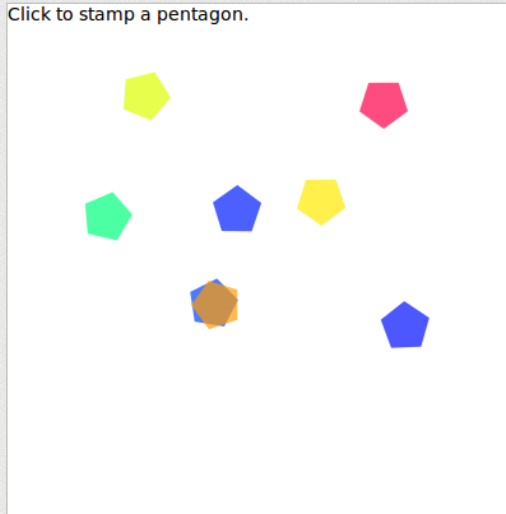
Embeding elm in HTML/CSS/JS

Stamps

This demo shows how to:

1. embed an Elm program in HTML
2. send events back and forth between Elm and JavaScript

The white square is an Elm program, but the rest is HTML, CSS, and JS. See the source code for more info.

Click to stamp a pentagon.

There are currently 8 stamps. Reset

You have created 8 stamps in total.

have a small sprout and raise it to grow big

fancy playground

```
-- Show an image of Yogi that resizes while maintain:

import Mouse

edgeLength : Signal Int
edgeLength =
    lift (\(x,y) -> max x y) Mouse.position

resizeableYogi : Int -> Element
resizeableYogi n =
    image n n "/yogi.jpg"

main : Signal Element
main =
    lift resizeableYogi edgeLength
```

Hints: ☑ Options: ☐   Auto-update: ☐ | Hot Swap | | Compile |

elm is good for beginer of FP

Live long and prosper

Appendix

- Official
  - http://elm-lang.org/

- Articles about Elm-lang's advantage
  - http://postd.cc/tag/elm/

- Server-side of this presentation is Go-lang
  - Server-sent event
  - net/http.Hijacker