



ecwid

Elm в production

Василий Васильков, Ecwid

Есwid

- SaaS-платформа для создания интернет-магазинов
- больше миллиона клиентов
- ~250.000.000 посетителей в месяц
- 120+ человек в команде
- Ульяновск, Самара, San Diego

Наш опыт

- Текущая версия “морды” написана на GWT
- Есть нюансы, но GWT правда хорошая технология ;)
- ~300.000 LOC на Java, компилируемого в JS
- Это транслируется в ~7mb (!) “красоты”

Вот такой красоты —

```
{1bD[a]=Y6D(WJD({l:l,m:0,n:0},a))
function x7b(a,b,c,d,e){var f,g,h
{A7b(a,d,c);return}for(h=0;h<d;h+
{g=eJb(eJb(sJb(fJb(lJb(f),{l:y0e,
{1:y0e,m:1023,h:0})),fJb(lJb(c[h+
{1:y0e,m:1023,h:0})));c[h+j]=CJb(c
function AWc(a,b){this.a=new CWb;
iXb;this.i=new y8b;this.d=new ZJ;
FWc(this);this.c=a;this.k=b;sd(th
content');ax(dd(this.b),Olf);sd(t
navigation');this.f.Jh(this.e);ge
e(this.a,this.g);DTb(this,this.a)
function GWd(b,c,d,e,f,g,h){var j
JRb(b,Ltf,'register');try{m=IRb(l
```

Наши условия

- больше 20 разработчиков в команде
- срок жизни — не меньше 5 лет
- больше 100.000 LOC
- Очень активная разработка и рефакторинг

Новая “морда”

- GWT
- Plain JS
- TypeScript + Redux + React
- <Lang>.js

Elm

- Первое упоминание - 2012
- Независимый язык, НЕ надстройка над JS
- Компиляция Elm → JavaScript (очень быстрая!)
- Функциональный
- Строгая статическая типизация

Elm

- Глобальный вывод типов
- Неизменяемые структуры данных
- Чистые функции
- Алгебраические типы данных

Почему Elm?

Это все прекрасно, однако:

- **Функциональных языков много**
- **Компилируемых в JS тоже достаточно**

Никаких “undefined is not a function” :)

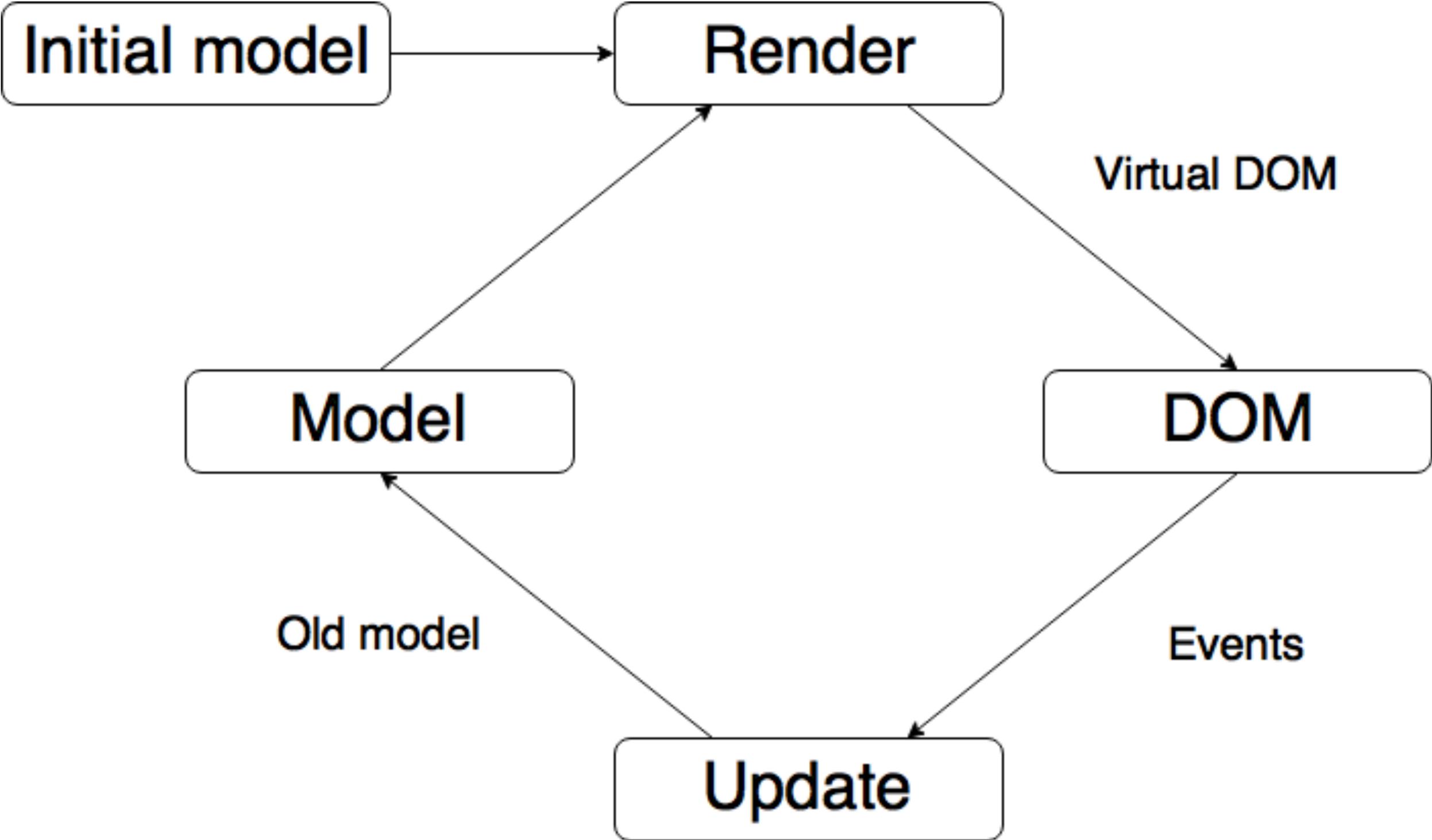
- **Нет ошибок в runtime**
- **Очень простой синтаксис**



Elm architecture

Классика

- начальная модель
 - отрисовка модели
 - изменение модели
 - опять отрисовка
 - и опять изменение
- Update: Model → Model
 - Render: Model → Html



Elm

-

0

+

```
type alias Model = {  
    value : Int  
}
```

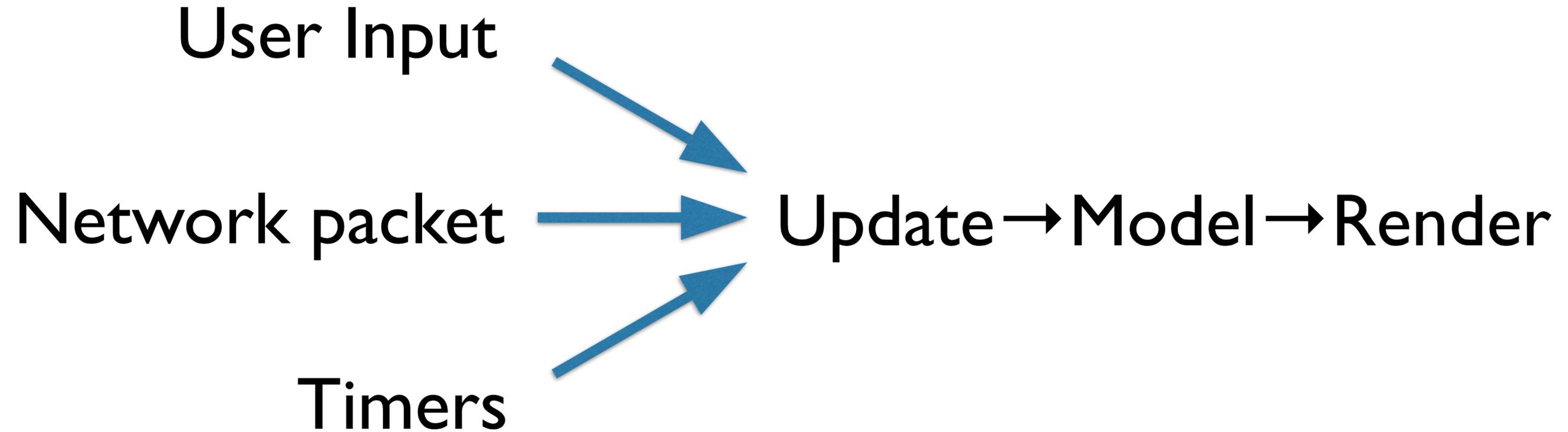
Elm

```
render model =  
  div []  
    [ button [ onClick Decrement ] [ text "-" ]  
    , div [] [ text (toString model.value) ]  
    , button [ onClick Increment ] [ text "+" ]  
    ]
```

Elm

```
update msg model =  
  case msg of  
    Increment ->  
      Model {value = model.value + 1}  
  
    Decrement ->  
      Model {value = model.value - 1}
```

Elm



Все проходит через update-функцию

Elm debug

- Отладка не нужна ;)
- Если всё-таки нужна — есть Time Travel Debugger
- Ещё можно JavaScript почитать, но без фанатизма

Time Travel Debugger

Все в debug-console. Как-то так:

- Initial Model
- onClick + Network (User logged In)
- onClick + Network (User add first item)
- onClick + Network (User add second item)

Time Travel Debugger

-- *Начальная модель*

```
store = {  
    customer = Anonymous,  
    cart = [],  
    wishList = []  
}
```

Time Travel Debugger

-- *Клиент залогинился*

```
store = {  
  customer = {  
    name = "Петрович",  
    email = "user@example.com"  
  },  
  cart = [],  
  wishList = []  
}
```

Time Travel Debugger

-- Клиент выбрал какие-то товары и положил в корзину

```
store = {  
  customer = {  
    name = "Петрович",  
    email = "user@example.com"  
  },  
  cart = [  
    {name = "Тихий Дон", price = 24.99, quantity = 2},  
    {name = "Война и Мир", price = 32.99, quantity = 1}  
  ],  
  wishList = []  
}
```



about:blank

Init	0
Login	1
AddToCart	2
AddToCart	3

```
▼ {  
  ▼ cart = List(2)  
    ▶ 0 = { name = "Тихий Дон", price = 24.99, quantity = 2 }  
    ▶ 1 = { name = "Война и Мир", price = 32.99, quantity = 1 }  
  }  
  ▼ customer = Customer {email,name}  
    email = "user@example.com"  
    name = "Петрович"  
  ▼ wishList = List(0)  
}
```

Resume

Import / Export

Elm packages

- У Elm есть стандартная библиотека
- Ну и пакетный менеджер (с пакетами)
- И поддержка semver (просто космос!)

Как попробовать

- Установить готовый пакет с elm-lang.org
- Через прт-магию
 - elm-make
 - elm-reactor
 - elm-format
 - elm-package
 - elm-repl



Хотите писать на Elm?
<https://github.com/Ecwid/new-job>

Василий Васильков, @2vgv