REASON (ML)

& OCAML

Michal **Miky** Jankovský

PROČ JSME TU

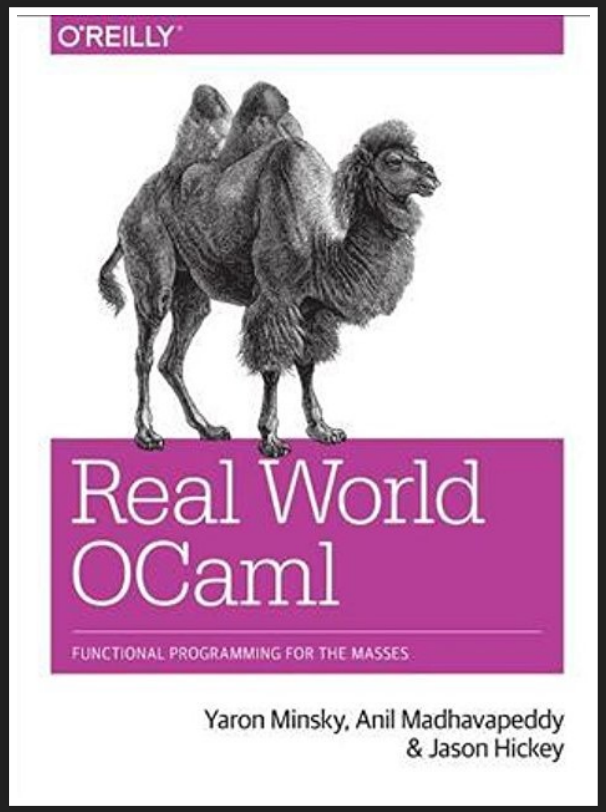
https://reactiveconf.com

OPRAVDU ZAJÍMAVÝCH TÉMAT BYLO MNOHO

- GraphQL subscriptions
- Logux
- GlimmerVM
- style-elements
- Převod aplikace z NG do Reactu
- Replicated Object Notation
- Vue (animace)
- bundle splitting
- D3

JAK JSEM POTKAL REASONML

ReactiveConf vypsala workshop...



matně si vybavuju z twitteru

FUNCTIONAL PROGRAMMING FOR THE MASSES

JAK JSEM POTKAL REASONML 1

26.10. Reason: JavaScript-flavored OCaml - Jared Forsyth

dojem **dost dobrý**

JAK JSEM POTKAL REASONML 2

26.10. network(dr)ing - "Maty"

dojem **úplně nejvíc nejlepší**

JAK JSEM POTKAL REASONML 3

- 27.10. workshop - Sean, Jared, Daniel

dojem **syntax... WTF**

JAK JSEM POTKAL REASONML 4

- muj pruzkum... ;)
  - dokumentace
    - https://reasonml.github.io
    - https://ocaml.org
  - blogy
  - youtube
- discord chat
  - oficiální kanál (support a diskuze)
  - https://discordapp.com/invite/reasonml
- příprava na tenhle talk

AGENDA

- **OCaml**
  - Historie
  - Hlavní výhody
- **ReasonML**
  - Cíl
  - Syntaxe
  - Toolchain
  - Aktuální stav
- **Ukázka kódu**

OCAML - HISTORIE

- 1981 | LCF proof assistant
- 1985 | CAM (Categorical Abstract Machine)
- 1987 | CAM + ML (Meta Language) = Caml
- 1990 | Caml Light
- 1995 | Caml Special Light
- **1996 | Objective Caml**
- **2011 | Renamed to OCaml**

https://ocaml.org/learn/history.html

OCAML - TOP BENEFITS

- funkcionální, ale rozšířený o objektové struktury
- s důrazem na **rychlost a bezpečnost** již od návrhu
  - bezpečná alokace paměti (žádné přetečení)
  - nejrychlejší garbage collector vůbec (pouze v jednom vlákně)
- **silně typovaný** jazyk
  - "fully **soundness**" https://en.wikipedia.org/wiki/Soundness
  - type inference (typy jsou odvozeny z kontextu)
- immutabilita (ale výjimky jsou povoleny)

OCAML - TOP BENEFITS 2

- FFI - Foreign Function Interface
  - možnost napojení na jiné jazyky
- debugger umí krokovat oběma směry
- tree-shaking ale nejen funkce i větve v ifech a switchich
- pattern matching
- uni-kernels (binarka + mini linux = iso)

REASONML

https://github.com/facebook/reason

Simple, fast & type safe code that leverages the JavaScript & OCaml ecosystems

REASONML - CÍL

OCaml je skvělý, ale developerů je málo, co s tím?

- **tak OCaml zůstane jen na pozadí**
- **zkrátit čas nastavení Reason prostředí**
  - z 2-3 dní na `"npm install"`
- **intuitivnější syntaxe**
  - **konverze** mezi verzemi syntaxe **je zadarmo** převedení na **AST (Abstract Syntax Tree)** a pak jakýkoli **fully Soundness** jazyk (Reason v2, Reason v3, FlowType)
  - konkétně cíli na JS programátory (ReactReason)

REASONML - TOOLCHAIN

Reason => Ocaml => BuckleScript => Javascript

(Flow + Babel + ESLint + Prettier) reason do it even better

merlin - poskytuje IDE všechny tooltipy, warningy a intelisense

REASONML - AKTUÁLNÍ STAV

- **Reason jeste neni "production ready"**
  - hlavní soustředění je teď na JavaScriptovou část
  - realistický odhad - **zbývají 2 roky**
    - **Jared** říkal před půl rokem "půl roku" https://jaredforsyth.com/2017/06/23/when-will-reasonml-be-ready/
    - **Sean** říká 1 - 2 roky
    - **Maty** 2 - 3 roky
- **Zatím malá komunita**
  - na discordu max 190 lidí online celosvětově

UKÁZKY KÓDU

- FFI čistý javascript
- Ocaml VS Reason
- React Reason

FFI ČISTÝ JAVASCRIPT

reference na js funkci

```
let module Document = {
  type element;
  let window: element = [%bs.raw "window"];
  external createElement : string => element = "document.createElement" [@@bs.val];
  external appendChild : element => element = "document.body.appendChild" [@@bs.val];
```

použití

```
let canvas = Document.createElement "canvas";
Document.appendChild canvas;
let setCanvasSize () => {
  let width = (Document.getWidth Document.window);
  let height = (Document.getHeight Document.window);
```

http://szymonkaliski.com/blog/2017-05-31-exploring-reasonml

CAMEL VS REASON

OCaml Record	Reason Record
`type r = {x: int; y: int}`	`type r = {x: int, y: int};`
`let myRec: r = {x = 0; y = 10}`	`let myRec: r = {x: 0, y: 10};`

OCaml Function	Reason Function
`type func = int -> int`	`type func = int => int;`
`let x: func = fun a -> a + 1`	`let x: func = (a) => a + 1;`

https://reasonml.github.io/guide/ocaml

CAMEL VS REASON 2

pattern matching

OCaml

```
let res = match x with
  | A (x, y) -> match y with
    | None -> 0
    | Some i -> 10
  | B (x, y) -> 0
```

Reason

```
let res = switch x {
  | A((x, y)) => switch y {
      | None => 0
      | Some(i) => 10
    }
  | B((x, y)) => 0
};
```

asymptotická složitost O(1)

REACTREASON

```
let component = ReasonReact.statelessComponent("Greeting");
let make = (~name, _children) => {
  ...component,
  render: (_self) =>
    <button>
      {ReasonReact.stringToElement("Hello!")}
    </button>
};
```

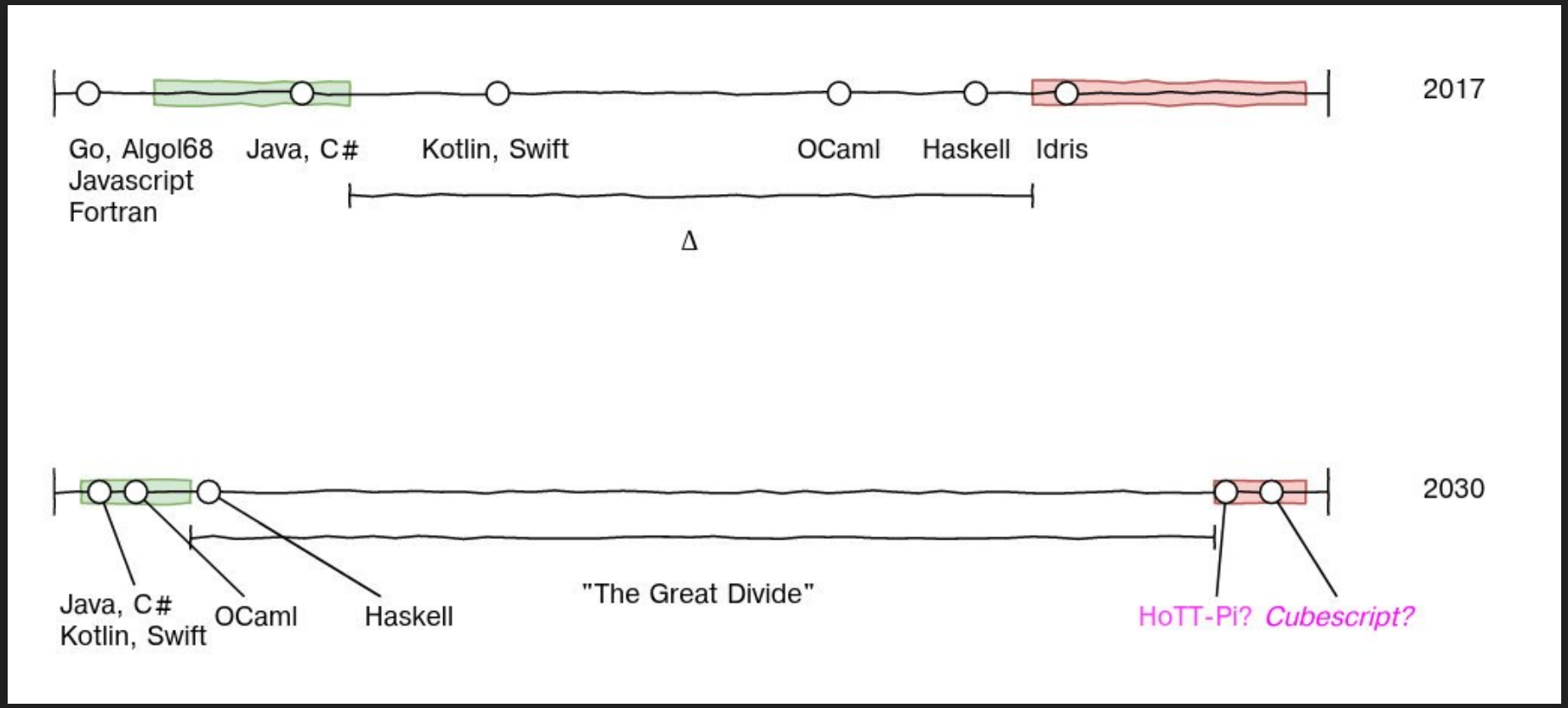
ZÁVĚR

- **OCaml funguje přes 20 let** (35 let se vyvíjí)
- **Funkcionální Programování dává smysl, tak proč ne v Reasonu**
  - syntaxí je mnohem blíž "běžnému" programátorovi než čistý OCaml / Haskell

ZÁVĚR 2

- **Facebook má už teď v Reasonu velkou část svého produkčního kódu** a přepis pokračuje
  - (celý FlowType, přes půlku messenger.com, část instagramu)
  - **bude ho udržovat**
  - **bude ho vylepšovat**

ZÁVĚR 3



http://dev.stephendiehl.com/nearfuture.pdf

Q&A