Melange: OCaml for JavaScript developers

Overview of the Melange project, its history, and its integration within the OCaml Platform

Slides

bit.ly/melange-zaragoza-2023

Hello!

I am Javier Chávarri

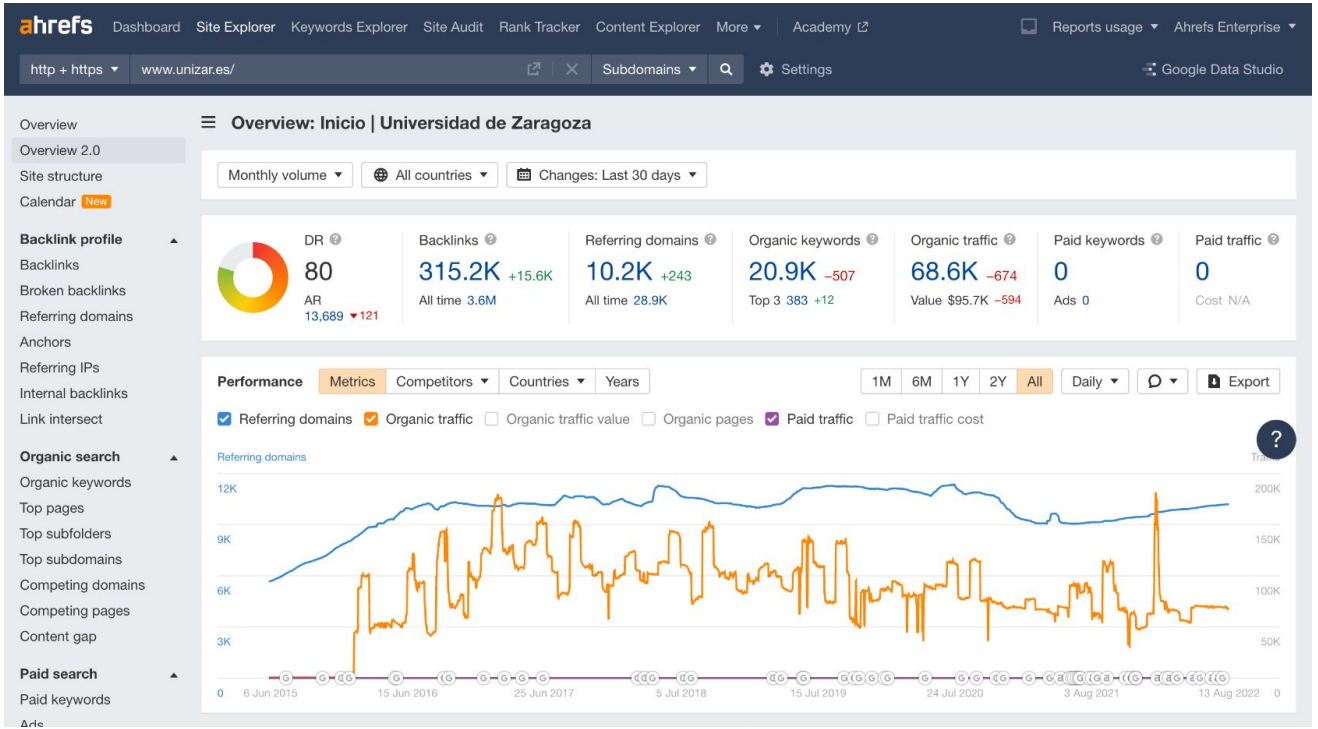
Software Engineer at Ahrefs.

You can find me at @javierwchavarri

ahrefs

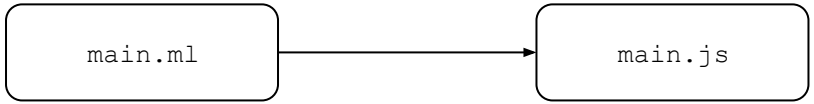
- SaaS for SEO agencies: ahrefs.com
- Transforming big data into relevant indicators
- 5 main tools to visualize and work with that data
- More recently, also a search engine: yep.com
- Founded in 2010, HQ in Singapore
- 100+ people, 65% remotely
- 17 OCaml developers, 25 Reason/Melange* developers

ahrefs


ahrefs
Dashboard
Site Explorer
Keywords Explorer
Site Audit
Rank Tracker
Content Explorer
More
Academy
Reports usage
Ahrefs Enterprise
http + https
www.unizar.es/
Subdomains
Settings
Google Data Studio
Overview
Overview 2.0
Site structure
Calendar
New
Backlink profile
Backlinks
Broken backlinks
Referring domains
Anchors
Referring IPs
Internal backlinks
Link intersect
Organic search
Organic keywords
Top pages
Top subfolders
Top subdomains
Competing domains
Competing pages
Content gap
Paid search
Paid keywords
Ads
Overview: Inicio | Universidad de Zaragoza
Monthly volume
All countries
Changes: Last 30 days
DR
80
AR
13,689 ▼121
Backlinks
315.2K +15.6K
All time 3.6M
Referring domains
10.2K +243
All time 28.9K
Organic keywords
20.9K −507
Top 3 383 +12
Organic traffic
68.6K −674
Value $95.7K −594
Paid keywords
0
Ads 0
Paid traffic
0
Cost N/A
Performance
Metrics
Competitors
Countries
Years
1M
6M
1Y
2Y
All
Daily
Export
Referring domains
Organic traffic
Organic traffic value
Organic pages
Paid traffic
Paid traffic cost
Referring domains
12K
9K
6K
3K
0
200K
150K
100K
50K
6 Jun 2015
15 Jun 2016
25 Jun 2017
5 Jul 2018
15 Jul 2019
24 Jul 2020
3 Aug 2021
13 Aug 2022

What is Melange

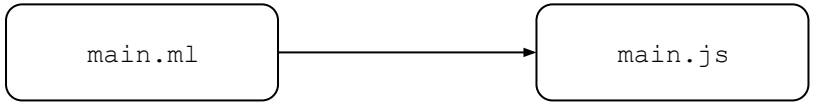
A backend for the OCaml compiler that emits JavaScript…



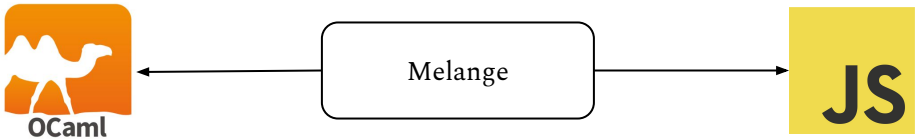
Another OCaml compiler to JS?

What is Melange

A backend for the OCaml compiler that emits JavaScript…



…and strives to provide the best integration with both the OCaml and JavaScript ecosystems.



What is Melange

```
let rec factorial n =
  match n <= 0 with
  | true -> 1
  | false -> n * factorial (n - 1)

let () = Js.log (factorial 6)
```



```
function factorial(n) {
  if (n <= 0) {
    return 1;
  } else {
    return Math.imul(n, factorial(n - 1 | 0));
  }
}

console.log(factorial(6));
```

What is Melange

```
(melange.emit
 (target app)
 (alias my-app)
 (libraries lib)
 (module_systems es6))
```

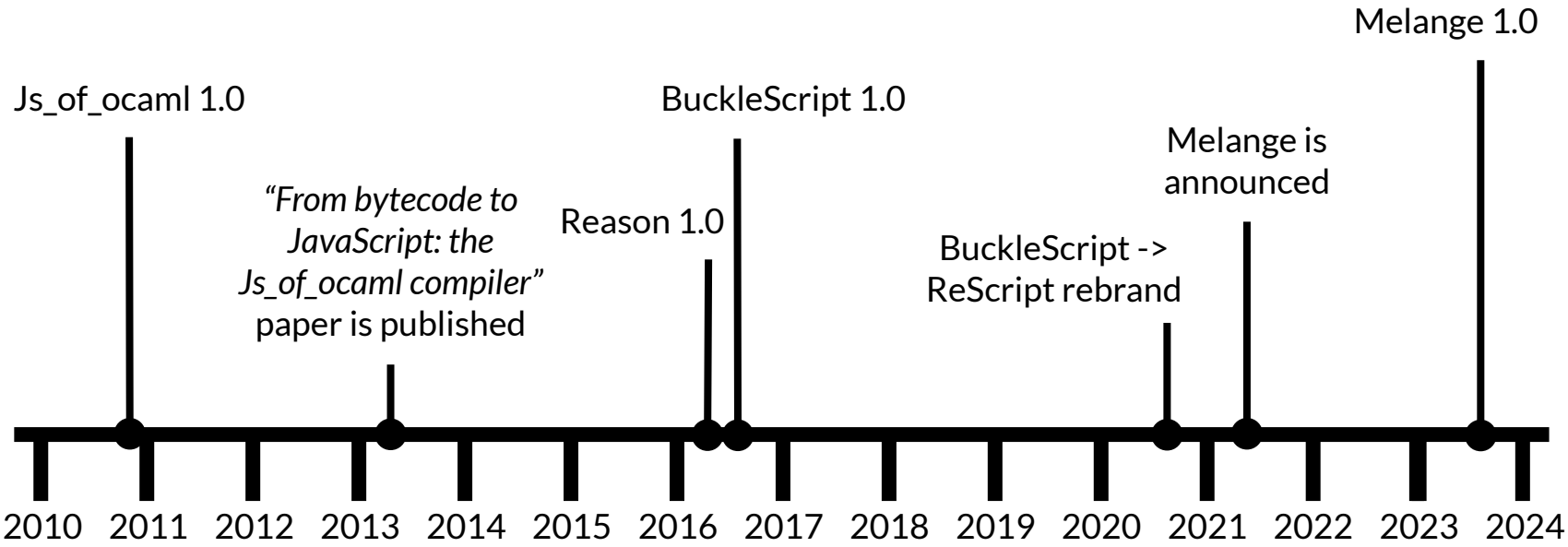
```
(library
 (name reshowcase)
 (public_name reshowcase)
 (preprocess
  (pps melange.ppx reason-react-ppx))
 (libraries reason-react)
 (modes melange))
```

Index

- History
- Compilation model
  - Melange compared to Js_of_ocaml
  - Reason
- Runtime representation
- Integration with Dune and opam
- Available libraries
- What’s next?

History

History

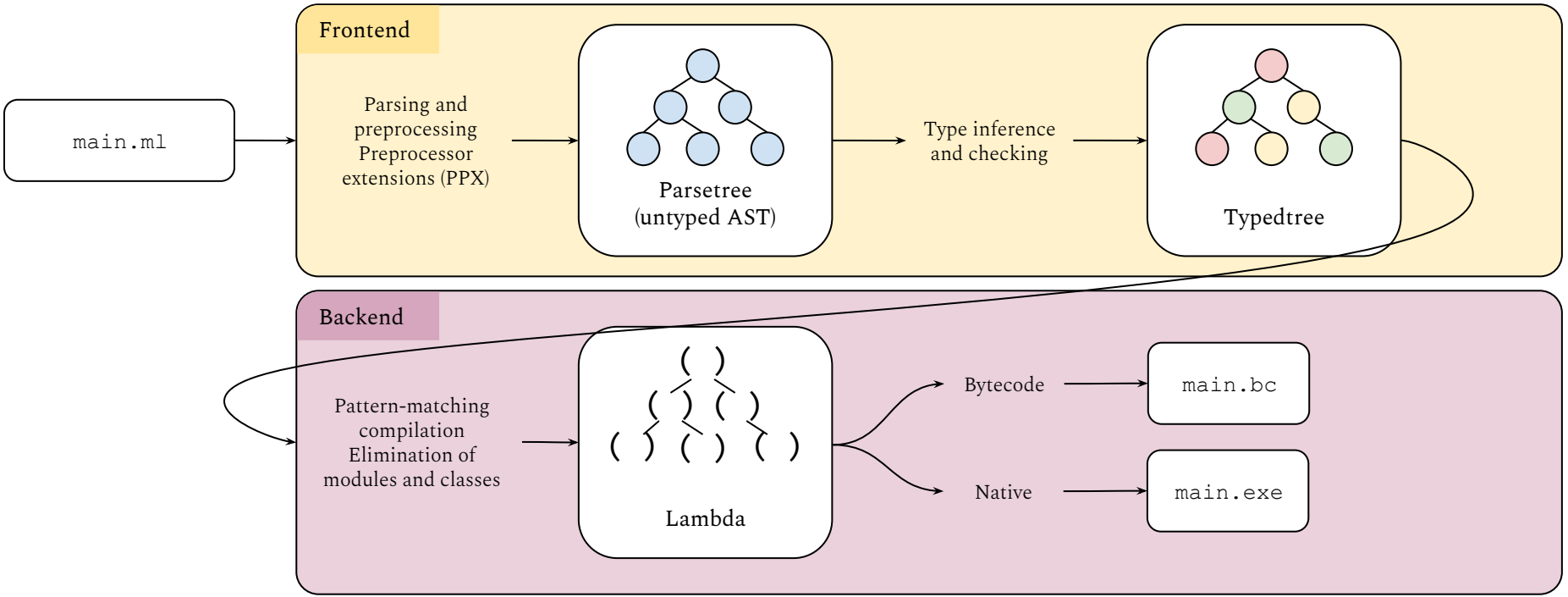

Melange 1.0
Js_of_ocaml 1.0
BuckleScript 1.0
Melange is announced
"From bytecode to JavaScript: the Js_of_ocaml compiler" paper is published
Reason 1.0
BuckleScript -> ReScript rebrand
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024

History

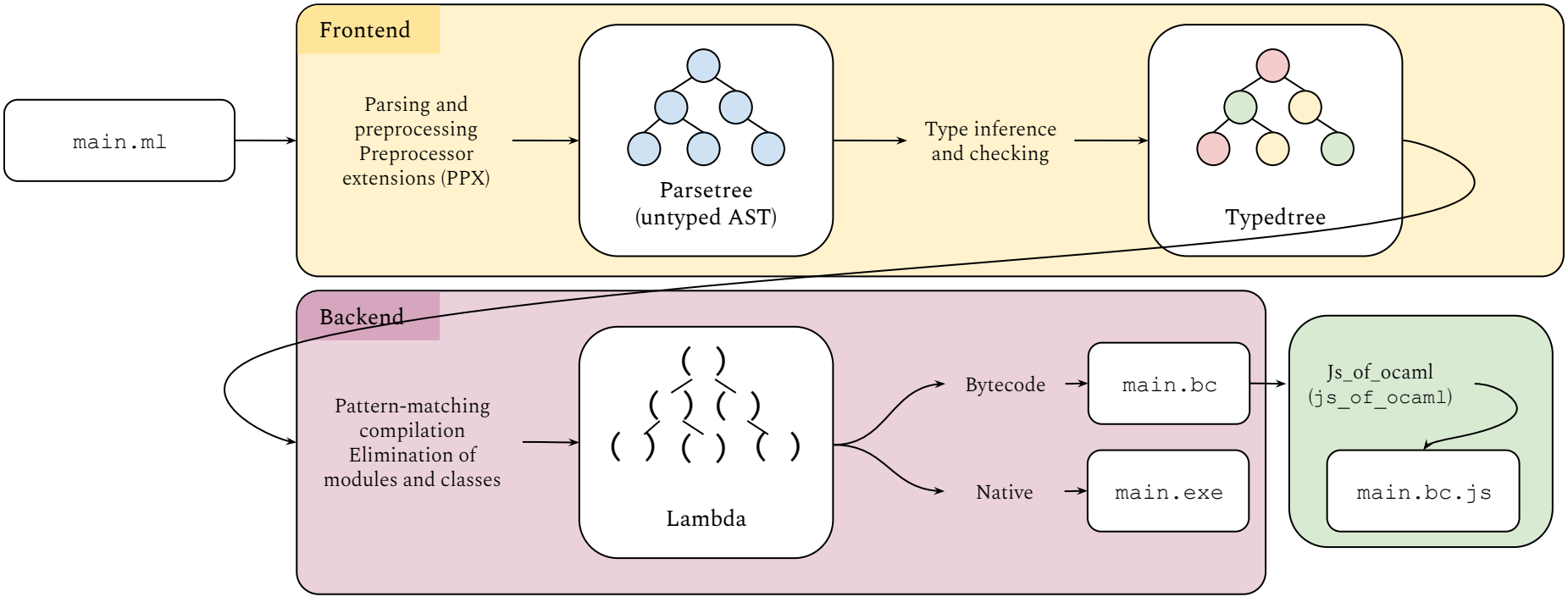
https://melange.re/v1.0.0/rationale/#a-bit-of-history

Compilation model

OCaml compilation pipeline



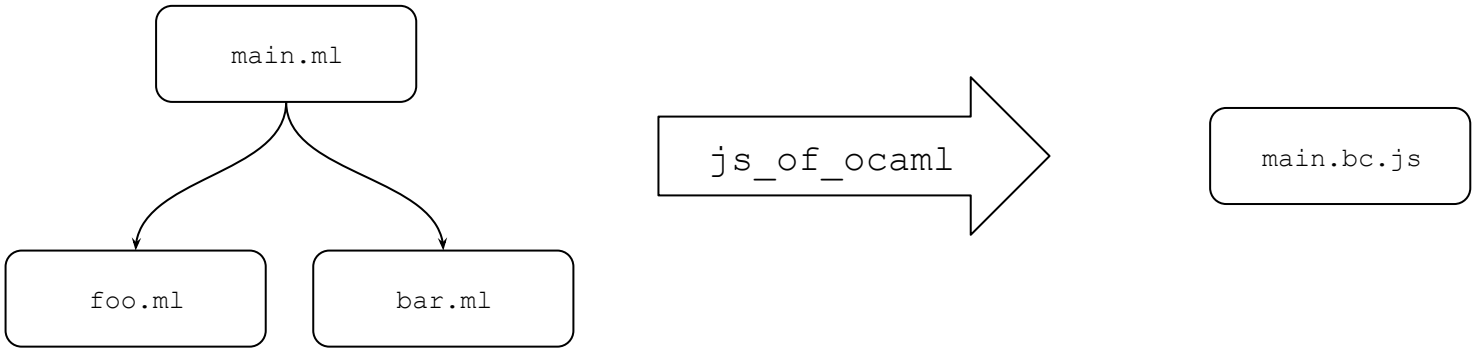
Js_of_ocaml: compilation model



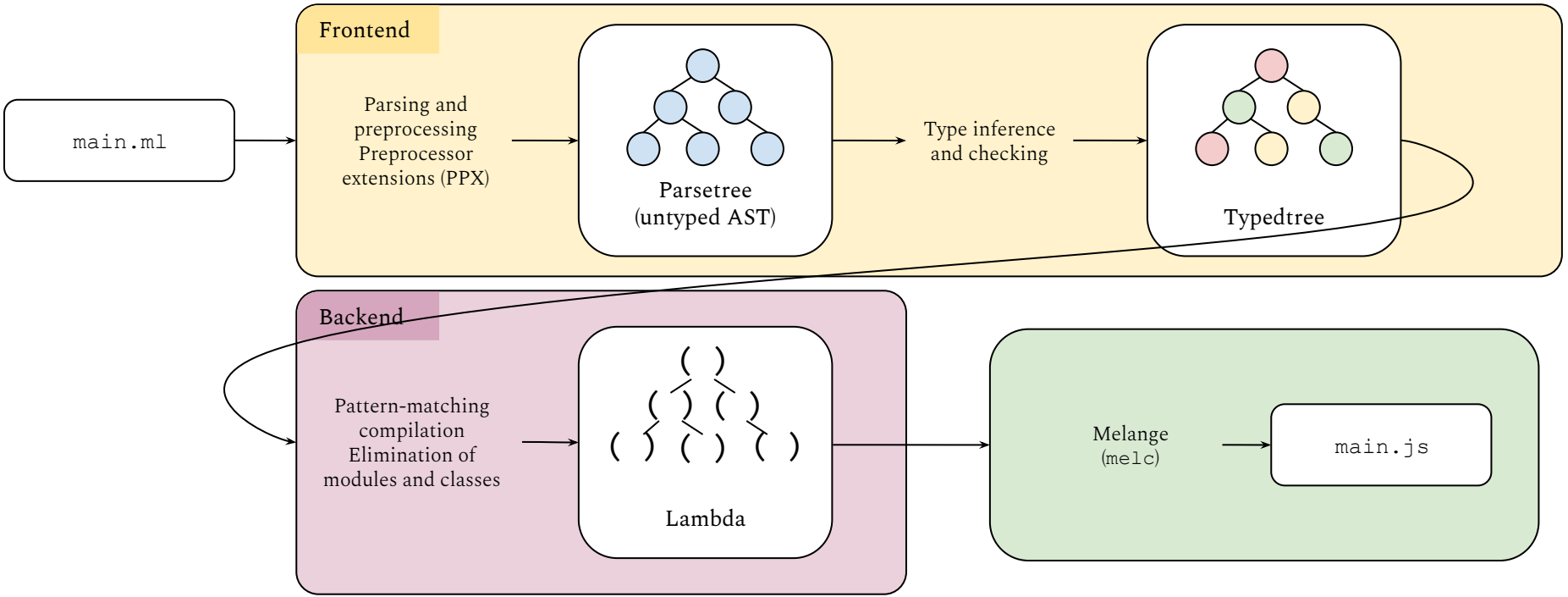
Js_of_ocaml: design goals

- Allow to reduce “language impedance” between server and client code
- Leverage JavaScript pervasiveness
- Low maintenance, thanks to stable API
- Maximize compatibility with the OCaml ecosystem
  - Integration with native toolchain
  - Almost all existing libraries can be compiled with Js_of_ocaml
  - Editor integration
- https://www.irif.fr/~vouillon/publi/js_of_ocaml.pdf

Js_of_ocaml: compilation model



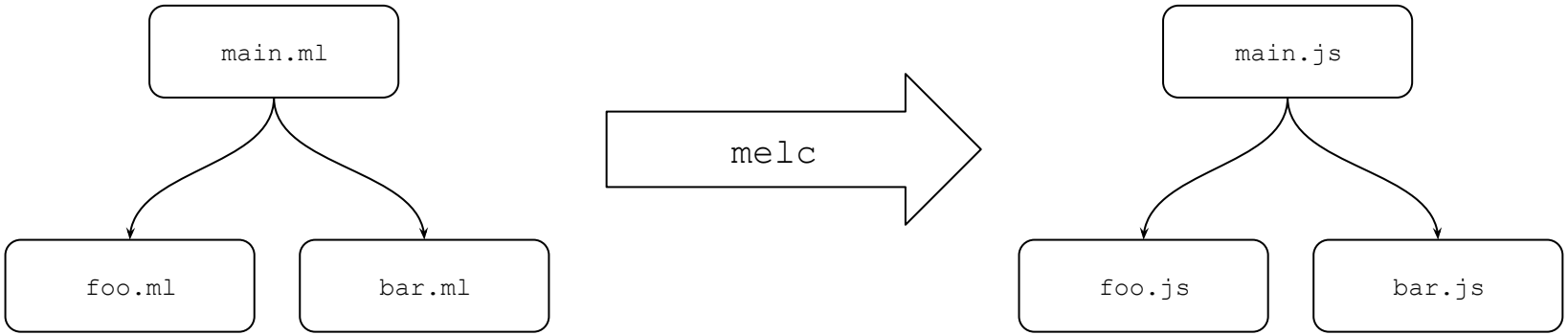
Melange: compilation model



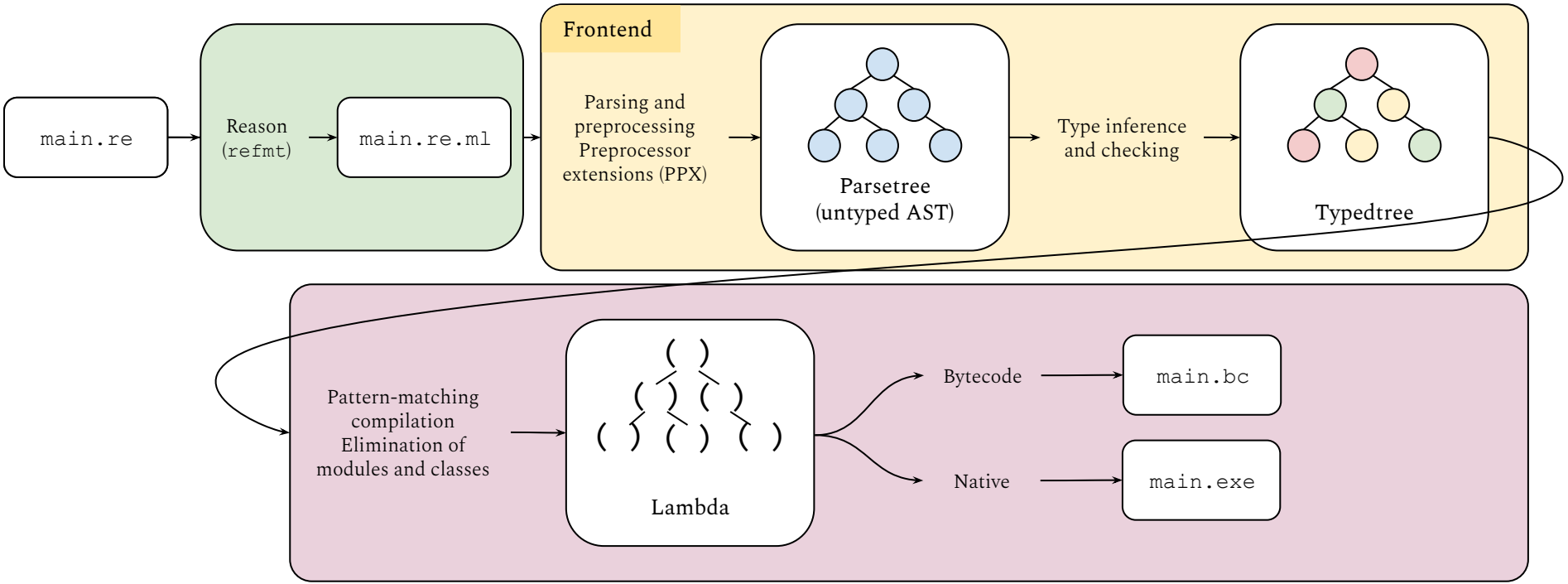
Melange: design goals

- Readable generated code
- Lighter bundles
- 1 OCaml module -> 1 JavaScript module
  - Enables progressive migration of JavaScript applications
  - Seamless consumption of npm packages, Webpack, dynamic imports, NextJS…
- Maximize compatibility with the OCaml Platform
  - Integration with native toolchain
  - Editor integration
  - opam, odoc, dune-release

Melange: compilation model



Reason


main.re
Reason
(refmt)
main.re.ml
Frontend
Parsing and
preprocessing
Preprocessor
extensions (PPX)
Parsetree
(untyped AST)
Type inference
and checking
Typedtree
Pattern-matching
compilation
Elimination of
modules and classes
Lambda
Bytecode
main.bc
Native
main.exe

Reason

OCaml syntax

```
let rec factorial n =
  match n <= 0 with
  | true -> 1
  | false -> n * factorial (n - 1)

let () = Js.log (factorial 6)
```

Reason syntax

```
let rec factorial = (n) =>
  n <= 0
  ? 1
  : n * factorial(n - 1);

Js.log(factorial(6));
```

Reason: design goals

- Friendlier syntax for JavaScript, C developers
- Compatible with any compiler backend (bytecode, native, Js_of_ocaml, Melange)
- First class integration within OCaml tooling (editor, Dune)
- Native support for JSX

Melange 🤝 Reason

```
module Greeting = {
  [@react.component]
  let make = () => {
    <button> {React.string("Hello!")} </button>;
  };
};

ReactDOM.querySelector("#preview")
|> Option.iter(ReactDOM.render(<Greeting />));
```

Runtime representation

Runtime representation

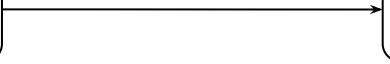
Melange allows to have a closer representation of OCaml values to JavaScript runtime values, which is useful for both debugging and leaner bundles.

Melange	int	nativeint	int32	float	string	array	tuple	record	list	bool
JavaScript	number	number	number	number	string	array	array	object	object	boolean

Records

- Melange records are compiled to JavaScript objects

```
{x = 1; y = 2}
```



```
{x: 1, y: 2}
```

- Can rename fields easily

```
type action = {
  type_ : string [@bs.as "type"]
}

let action = { type_ = "ADD" }
```

```
var action = {
  type: "ADD"
};
```

Tuples

- Tuples compile to JavaScript arrays

```
("hey", "there")
```

→

```
["hey", "there"]
```

- This is useful for writing bindings to JavaScript libraries:

```
let () = React.useEffect2
(fun () -> None) (foo,
bar)
```

→

```
React.useEffect(function
() {}, [foo, bar]);
```

Other shared types

- Array
- Boolean
- String
- Floats
- Integers
- First-class integration with JavaScript regular expressions

✨✨✨ Smaller gap between compile and runtime types => simpler bindings & faster debugging ✨✨✨

Integration with Dune and opam

Create a library

- Install melange

```
$ opam install melange
```

- Add a new library / modify an existing one

```
(library
 (name foo)
 (modes melange))
```

```
(library
 (name foo)
 (modes :standard melange))
```

Create an app

```
(melange.emit
 (target app)
 (alias my-app)
 (libraries lib)
 (module_systems es6))
```

```
$ dune build @my-app
```

Publish in opam

- It’s just an opam file, can check template: https://github.com/melange-re/melange-opam-template
- Leverage existing tooling like https://github.com/tarides/dune-release/
- Some packages start being published in the opam public repository, for example https://opam.ocaml.org/packages/reason-react/

Available libraries and PPXs

Available libraries and PPXs

- reasonml/reason-react
- melange-community/melange-webapi
- melange-community/melange-jest
- melange-community/melange-json
- davesnx/styled-ppx
- ahrefs/melange-react-intl-ppx
- ahrefs/melange-atdgen-codec-runtime
- ahrefs/melange-recharts

What's next?

What's next?

- Increased compatibility with the OCaml Platform
  - Upgrade compiler-libs to OCaml 5
- Developer Experience
  - Improve editor integration, error locations on the React PPX
  - Improve opam / npm interaction
- Documentation and Branding
- Publication of more libraries and PPXs in the opam public repository

Resources

Resources

- Official docs: https://melange.re/
  - Playground: https://melange.re/unstable/playground/
- Discord: https://discord.gg/reasonml
- Getting started with ReasonReact and Melange: https://dev.to/psb/getting-started-with-reasonreact-and-melange-13hd

Slides

bit.ly/melange-zaragoza-2023

Thank you!