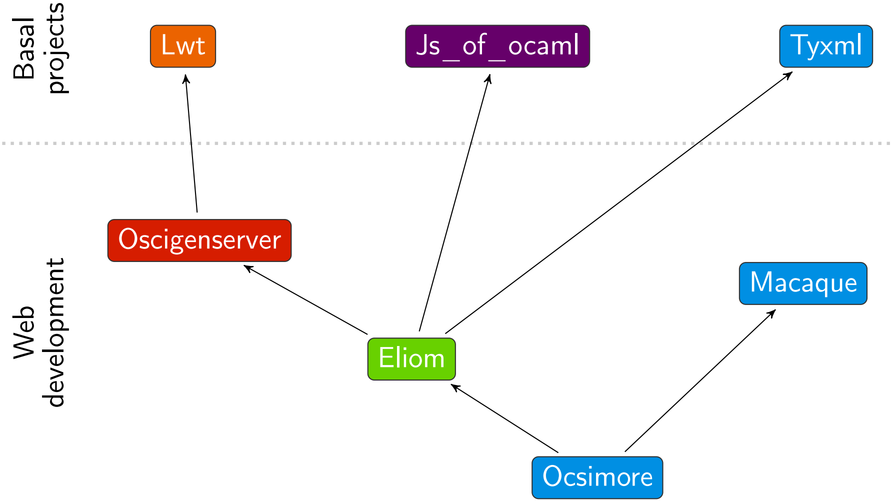
# The Ocsigen-Project

## The State of the Art, and the Prospects

Benedikt Becker
Vincent Balat

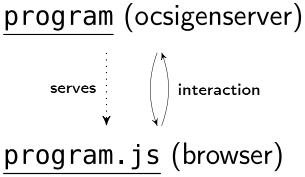
OUD, September 14, 2012

# The Ocsigen-Zoo



http://ocsigen.org/chat3

# eliomc



`program` (ocsigenserver)

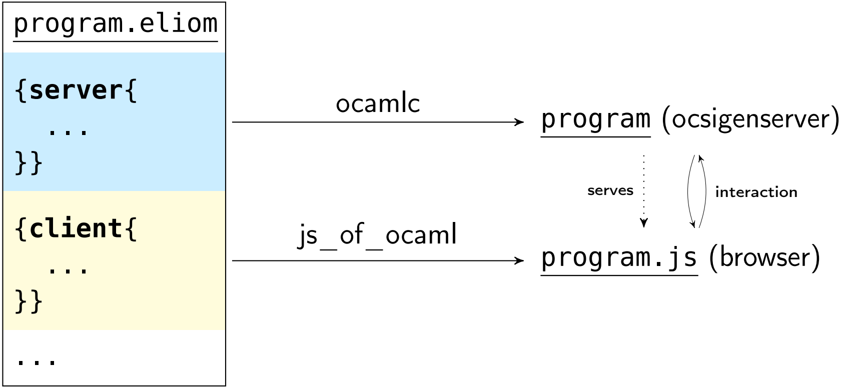
serves

interaction

`program.js` (browser)

# eliomc

## The compilation process



```
program.eliom

{server{
   ...
}}

{client{
   ...
}}

...
```

ocamlc

program (ocsigenserver)

serves

interaction

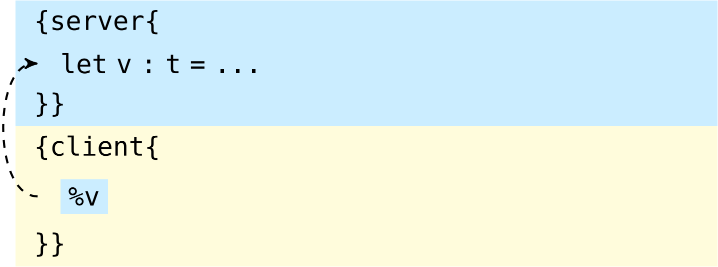
js_of_ocaml

program.js (browser)

# Injections

## Exchange between sections

- Injections



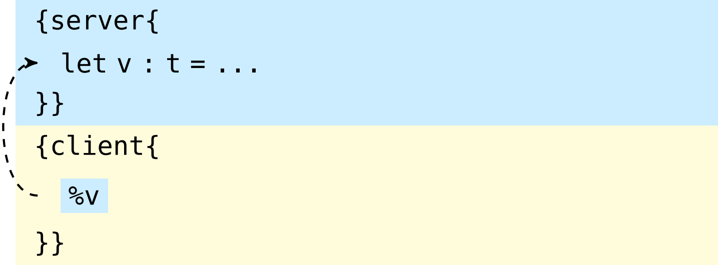
```
{server{
  let v : t = ...
}}
{client{
  %v
}}
```

  - The value as of the *delivery* of the program

# Injections

## Exchange between sections

- Injections



```
{server{
  let v : t = ...
}}
{client{
  %v
}}
```

  - The value as of the *delivery* of the program

- Example: Chat, reflecting the list of users

```
{server{
  ... let users : user signal = ...
}}
{client{
  let () = Lwt_react.S.keep
    (React.S.map (reflect_users users_ul) %users)
}}
```

# Client values

- Client values

  ```
  {server{
    ...
    let v : α client_value = {{ ... }} in
    ...
  }}
  ```

  - Abstract on server, synonym on client
  - Guess, what is a type of: `{{ %(x : t client_value) }}`

# Client values

- Client values

  ```
  {server{
    ...
    let v : α client_value = {{ ... }} in
    ...
  }}
  ```

  - Abstract on server, synonym on client
  - Guess, what is a type of: `{{ %(x : t client_value) }}`

- Example: Chat, browser event handler on the server

  ```
  {server{
    let dialog_widget dialog =
      let close =
        {{ fun ev -> close_dialog %dialog }}
      in div [ ...;
        span ~a:[a_onclick close] [pcdata "×"]]
  }}
  ```

## Batteries included

- Services

  *Abstraction of HTTP resources*
- References

  *Server-side state arranged by client*
- Client application

  *Persistent across navigation*
- Communication

  *Buses, reactive data*
- Remote procedure calls
- . . .

http://ocsigen.org