Elm |> Real Life

jiri.sliva@newired.com

NEWIRED

Pure Functional

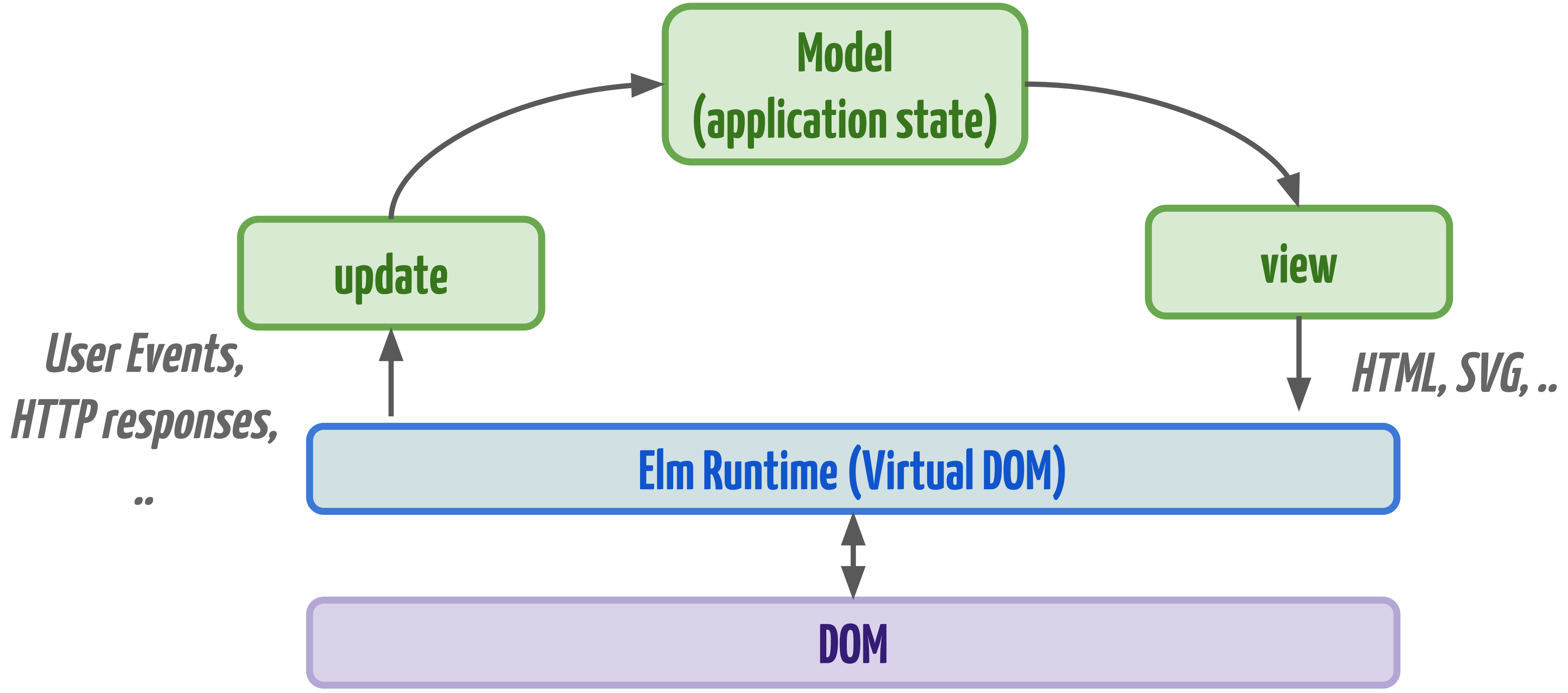
- Immutable data, only pure functions, ...
- ML family syntax *(Haskel for kids)*

```
view : List Article -> Html Msg
view articles =
    let
        viewArticle article =
            div [ class "article" ] [ text article.title ]
    in
    Html.ul
        [ class "article-list" ]
        (List.map viewArticle articles)
```

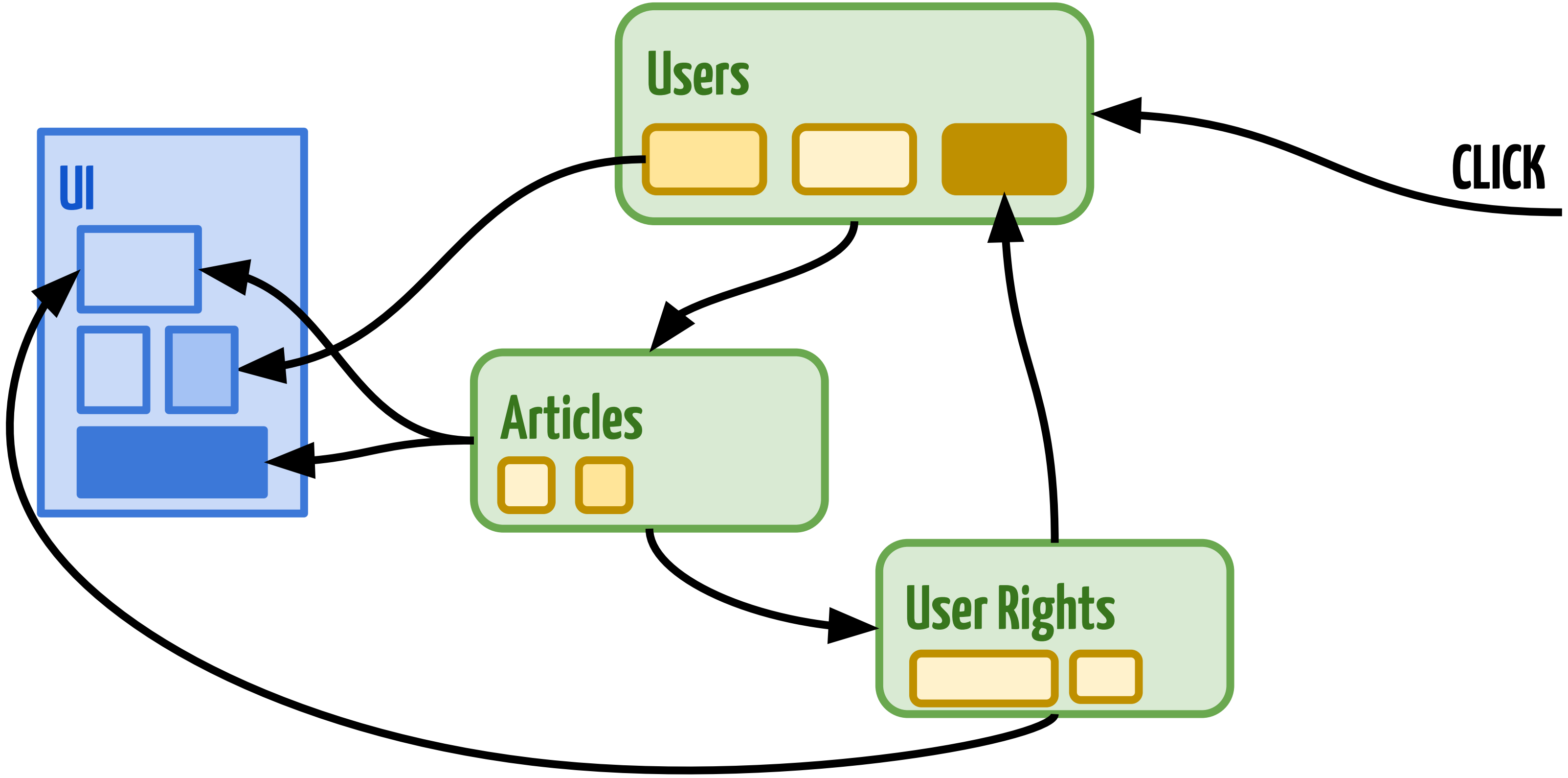
Syntax

- Types: String, Int, Boolean, Lists, Records
- Type aliases
- Custom Types
- Pattern matching (if-else, case)
- Functions

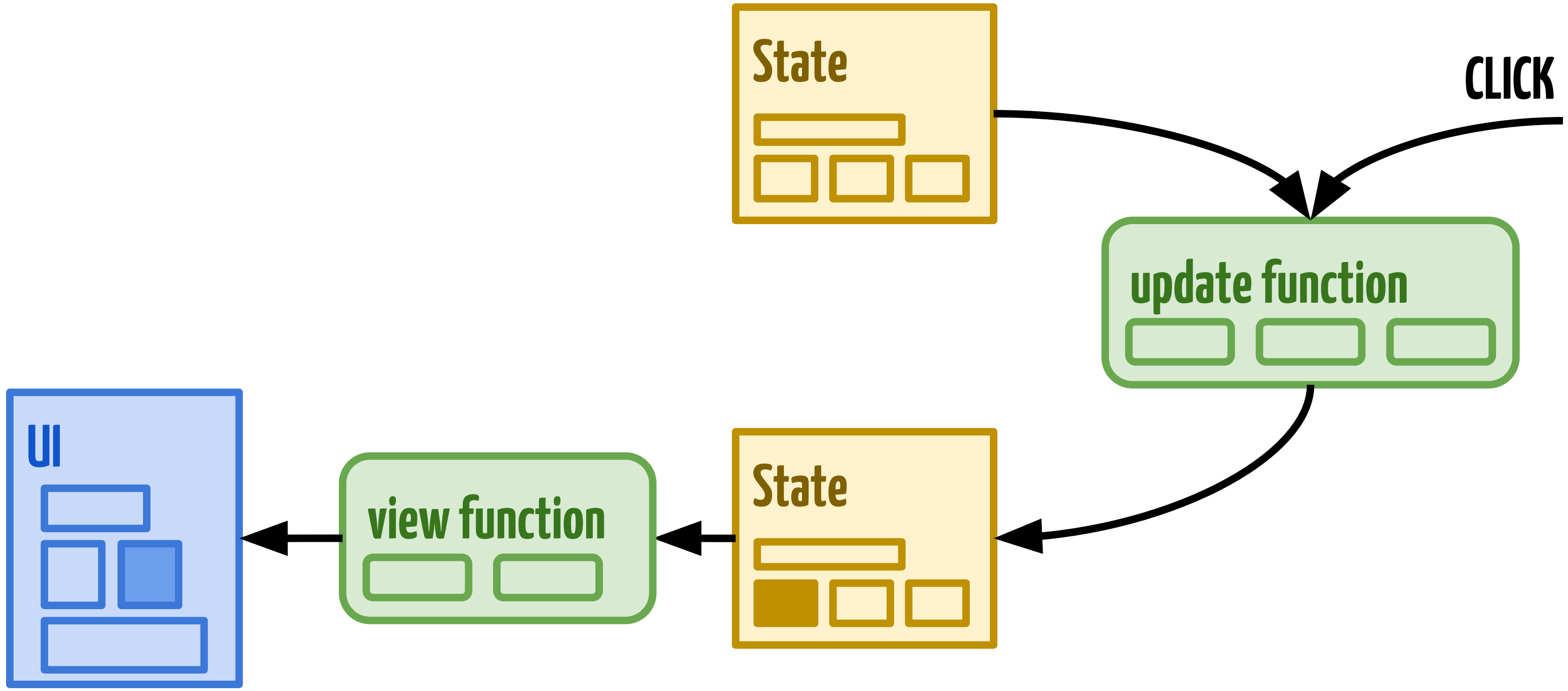
Model – Update – View



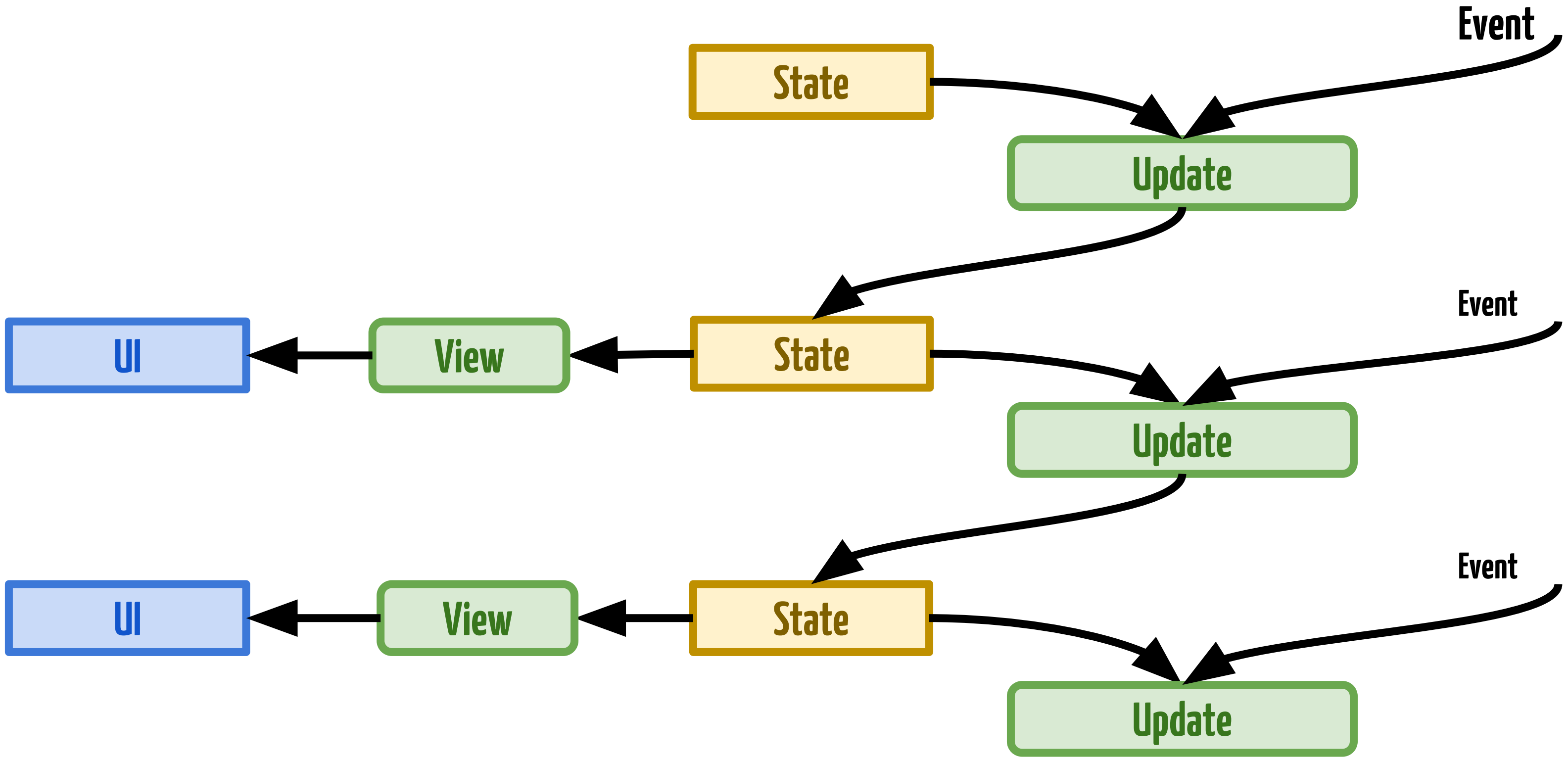
Inside *common application™*



Inside Elm Application



Elm Application Timeline



Demo!

1. Hello World
2. Update Model
3. Decomposition
4. Commands
5. Decoders
6. Tasks
7. JavaScript Interoperability – Ports & WebComponents
8. Unit tests

Interoperability with JavaScript – Ports (1)

Elm

```
port setProfile : String -> Cmd msg

update : Msg -> Model -> ( Model, Cmd Msg )
update msg model =
    case msg of
        SetProfile ->
            ( model, setProfile "pythagoras" )
        ..
```

JavaScript

```
  var app = Elm.Main.init({
    node: document.getElementById('elm')
  });

  app.ports.setProfile.subscribe(function(id) {
    console.log("Elm is asking for saving Profile with id: " + id)
    localStorage.setItem("profile", id);
  });
```

Interoperability with JavaScript – Ports (2)

Elm

```elm
port onLastAccess : (String -> msg) -> Sub msg

subscriptions : Model -> Sub Msg
subscriptions _ =
         onLastAccess OnLastAccess

type Msg
    = OnLastAccess String

update : Msg -> Model -> ( Model, Cmd Msg )
update msg model =
    case msg of
        OnLastAccess lastAccess ->
            ( { model | lastAccess = lastAccess }, Cmd.none )
        ..
```

JavaScript

```javascript
app.ports.onLastAccess.send(lastAccess);
```

WebComponents – Dispatch Custom Events

```
class CustomEditor extends HTMLElement {
  constructor() {
    const shadow = this.attachShadow({ mode: 'open' })
    const input = document.createElement("input")
    shadow.appendChild(input)

    const _this = this
    input.addEventListener("input", function () {
      _this.dispatchEvent(new CustomEvent("editorChanged", { detail: input.value }));
    })
  }
}

customElements.define('custom-editor', CustomEditor)
```

```
<custom-editor></custom-editor>
```

WebComponents – Handle Custom Events

```
node "custom-editor"
  [ property "editorValue" (Encode.string model.greetings)
  , on "editorChanged" editorChangedDecoder
  ]
  []
```

```
editorChangedDecoder : Decode.Decoder Msg
editorChangedDecoder =
    Decode.map EditorChanged <|
        Decode.at [ "detail" ] <|
            Decode.string
```

```
input.addEventListener("input", function () {
    _this.dispatchEvent(new CustomEvent("editorChanged", { detail: input.value }));
})
```

Tooling

- **Elm-Format** – Accepted by community as *the only one proper format* (used in 99% packages)
- **Elm Analyse** – Static analyze tool
- **Elm debugger** – Just compile with `--debug`
- Elm-monitor - Monitor your Elm program with redux-devtools
- IDEs support
  - **Intellij IDEA** – Refactoring, Find usages, Code completion, Type Inference and Type Checking
  - **Visual Studio Code**
  - **Atom**
  - **Vim**

Ecosystem

package.elm-lang.org

- vega-lite
- elm-graphql
- Elm Bootstrap
- elm-mdl - Material Design Library
- feathericons.com