



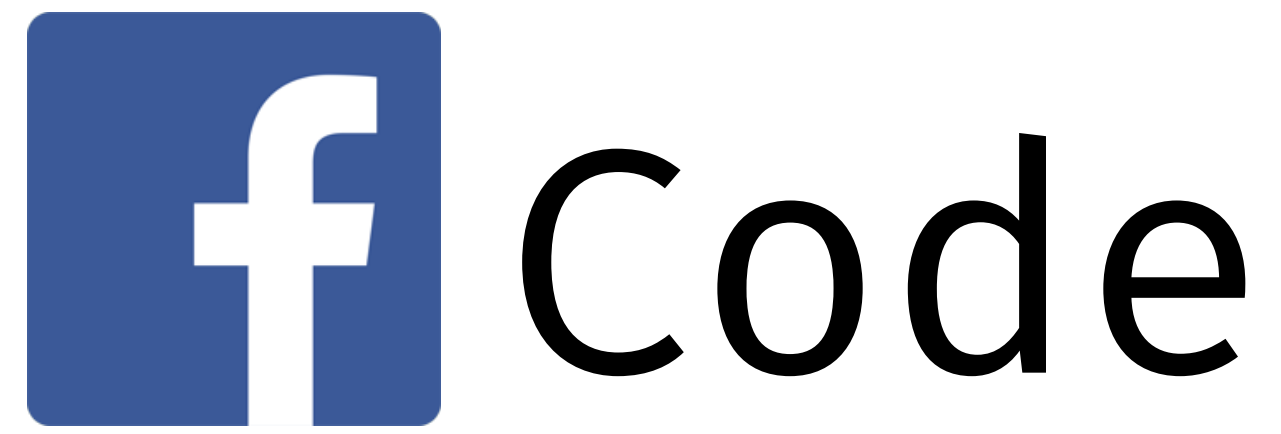
DVA ROKY ELMU



@robinpokorny



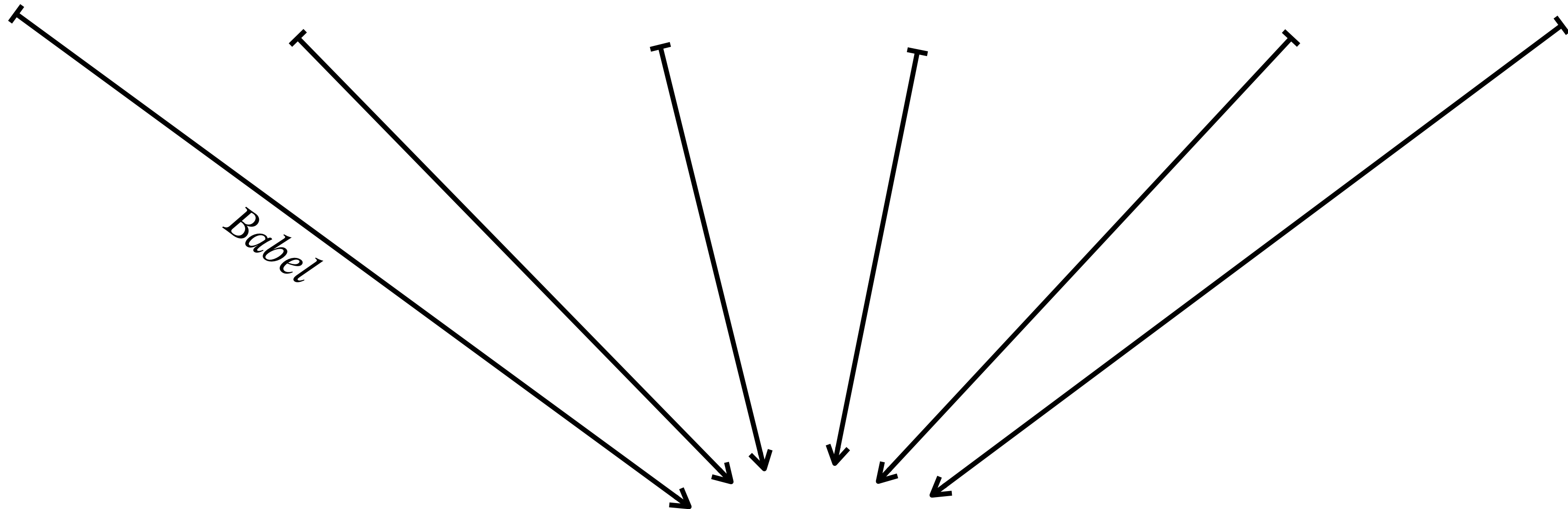
Dobrý pocit



IMMUTABLE



ES200x Elm TypeScript ClojureScript OCaml Dart ...

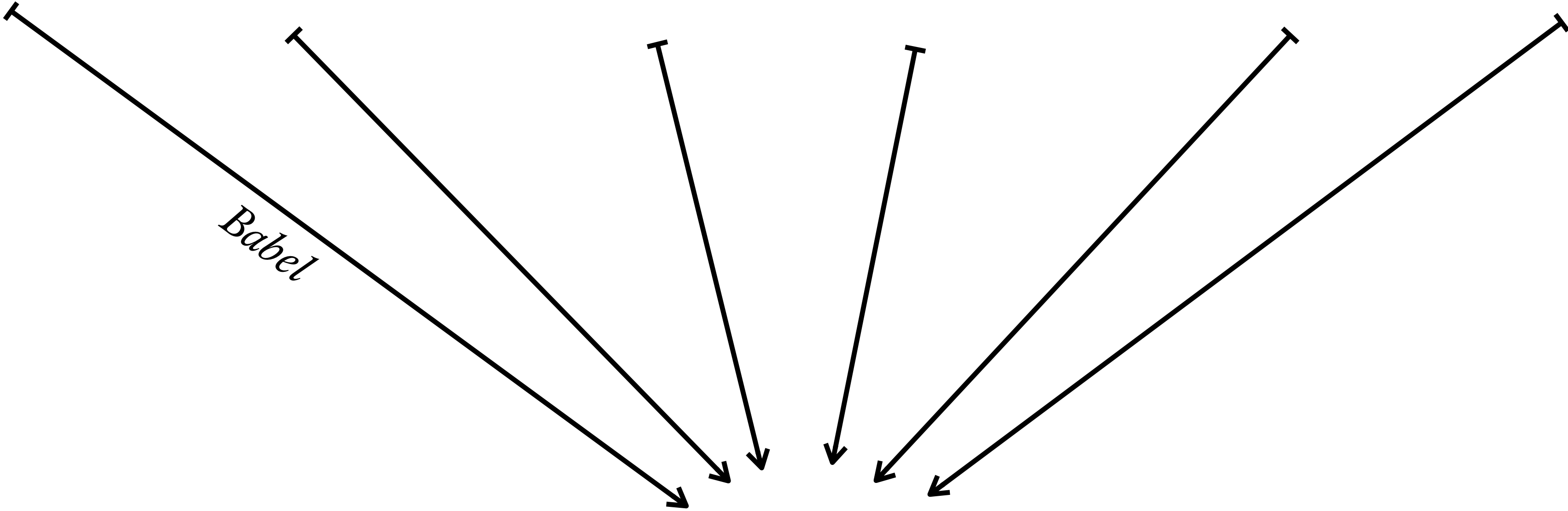


OBR. 17.—Transkompilace



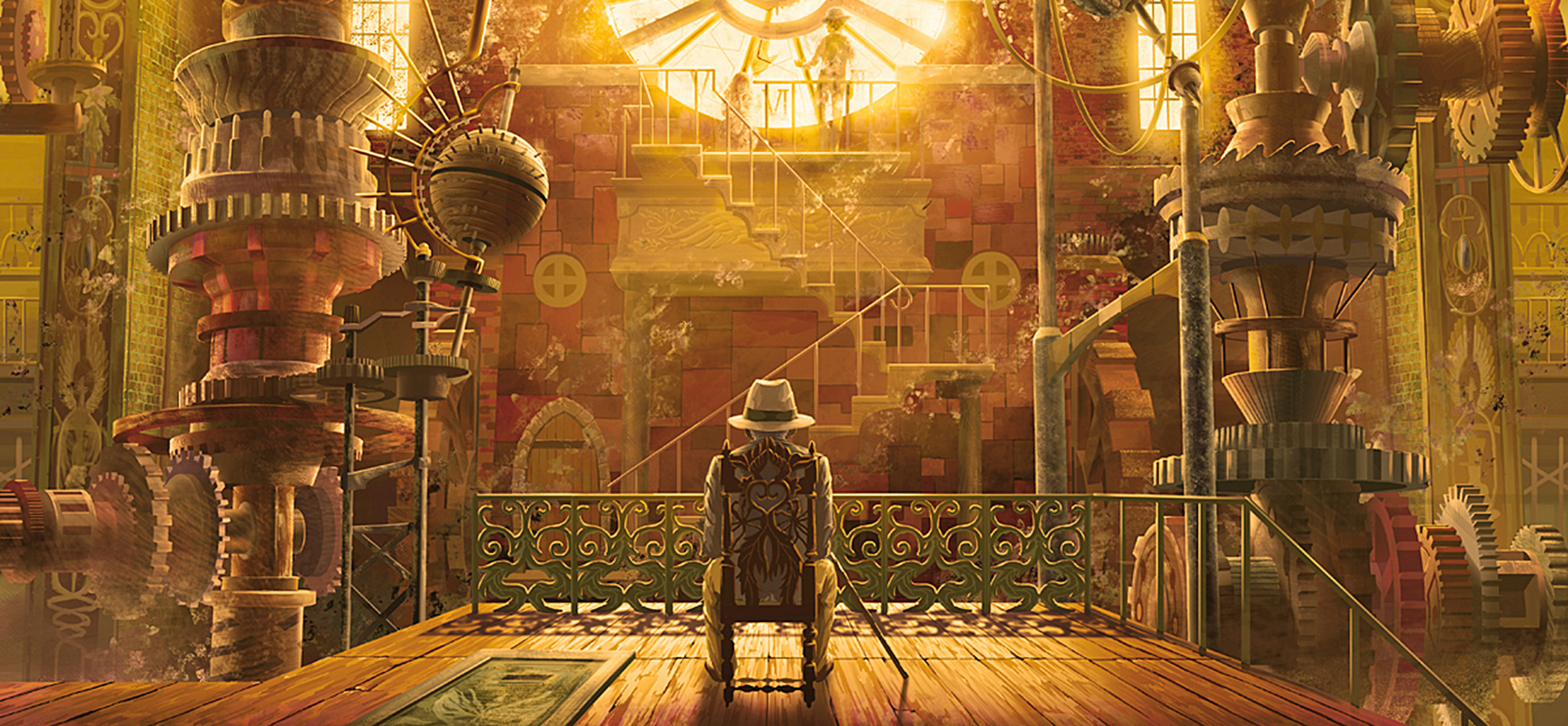
React+ x ElmArchitecture Angular Vue.js Polymer jQuery Ember.js Backbone ...

ES200 x Elm TypeScript ClojureScript OCaml Dart ...



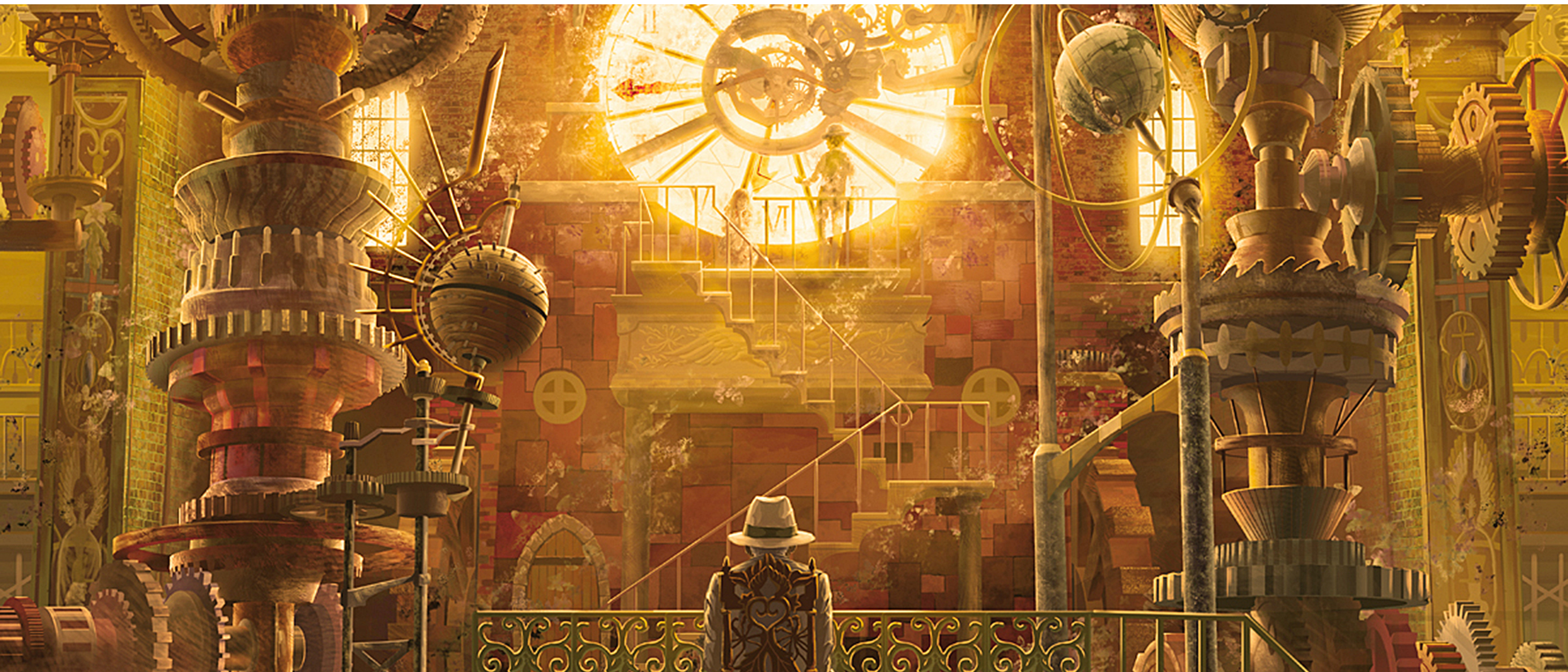
OBR. 17.—Transkompilace





stejný problém + různé cesty $\longrightarrow$ porozumění

Chci po vás 2 ~~roky~~ hodiny



ROK PRVNÍ

Elm

jako

jazyk

update : Msg → Model → (Model, Cmd Msg)

update msg model =

case **msg** of

Add →

{ **model**

 | uid = **model**.uid + 1

 , field = ""

 , entries =

 if String.isEmpty **model**.field then

model.entries

 else

model.entries ++ [newEntry **model**.field]

 }

! []

CheckAll isCompleted →

let

 updateEntry t = { t | completed = isCompleted }

in

{ **model** | entries = List.map updateEntry **model**.entries }

! []

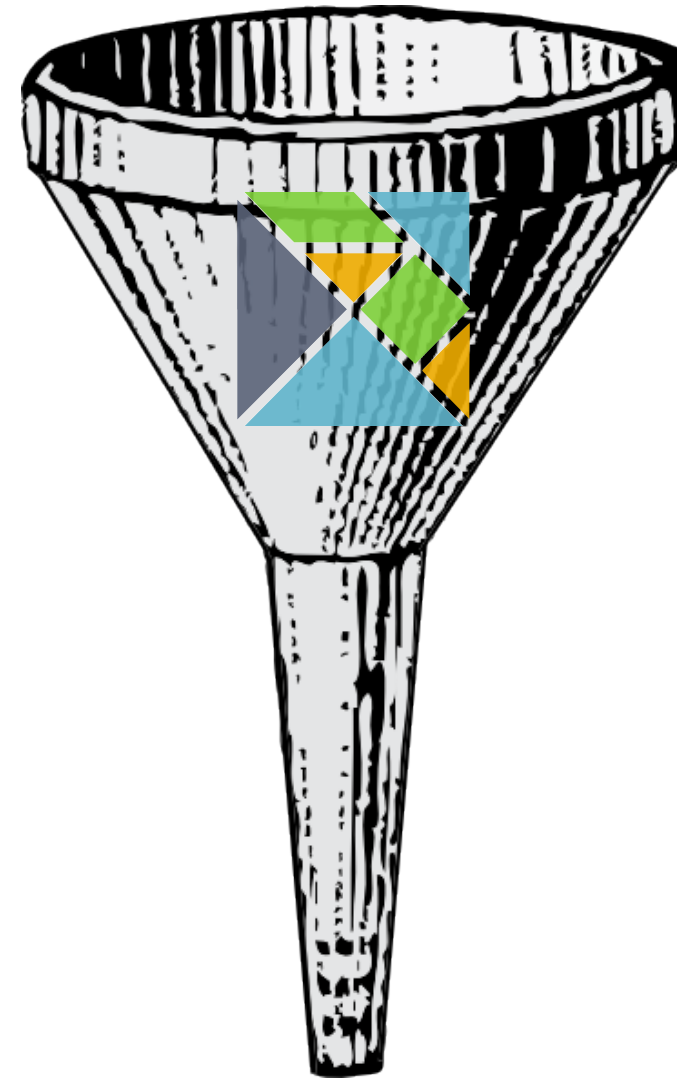


An advanced, purely functional programming language



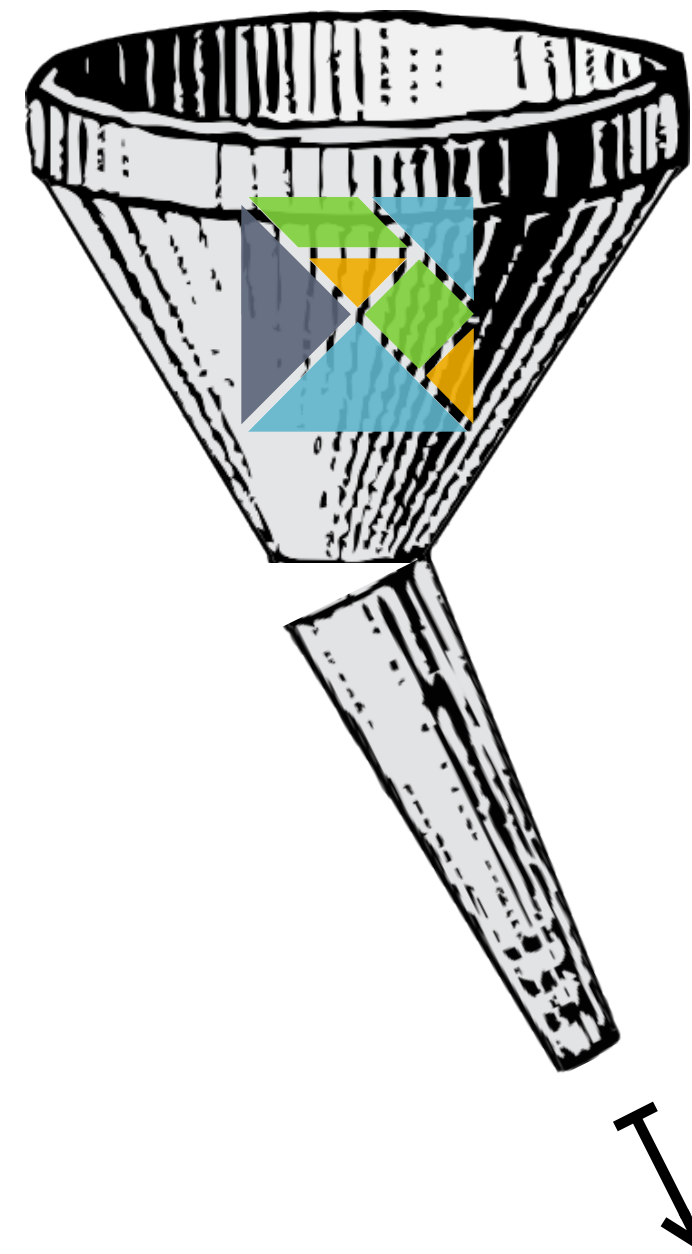
A delightful language for reliable webapps.

Main.elm



OBR. 31.—Transkompilace Elmu

Main.elm

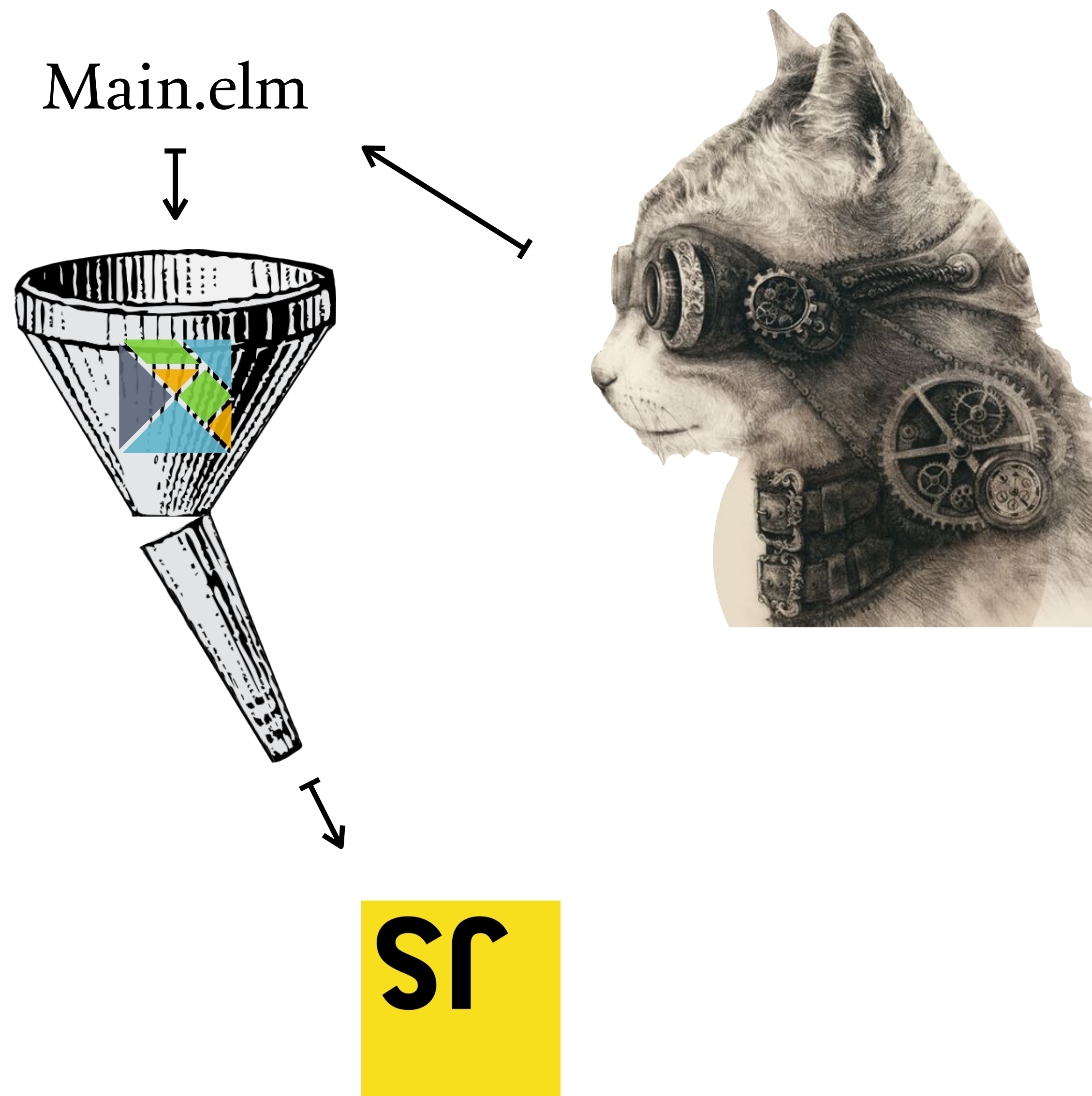


OBR. 32.—Ztracené miliony

„Předávají argument
něčemu, co není funkce!

Možná zapomněli na
závorky? Nebo čárku?“

OBR. 33.—Perfektní hlášky




```
> "foo" "bar"  
-- TYPE MISMATCH -----
```

You are giving an argument to something that is not a function!

```
3|   "foo" "bar"  
      ^^^^
```

Maybe you forgot some parentheses? Or a comma?




```
> "foo" + "bar"
-- TYPE MISMATCH -----
The left argument of (+) is causing a type mismatch.
```

```
3|   "foo" + "bar"
   ^^^^
```

(+) is expecting the left argument to be a:

number

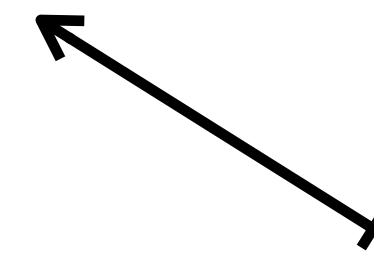
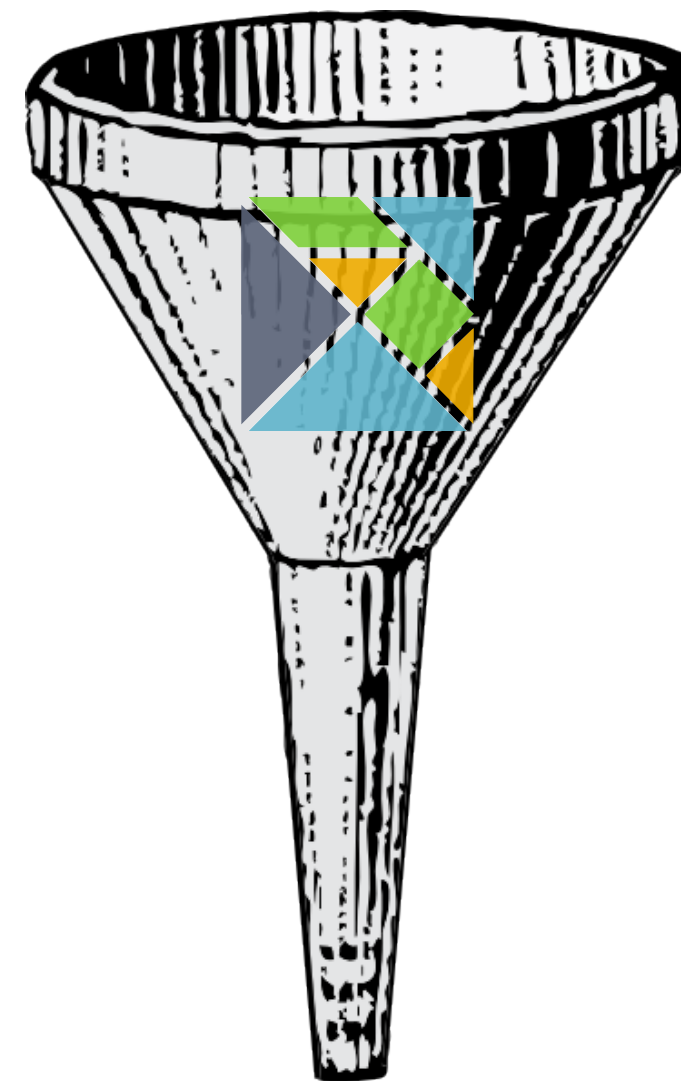
But the left argument is:

String

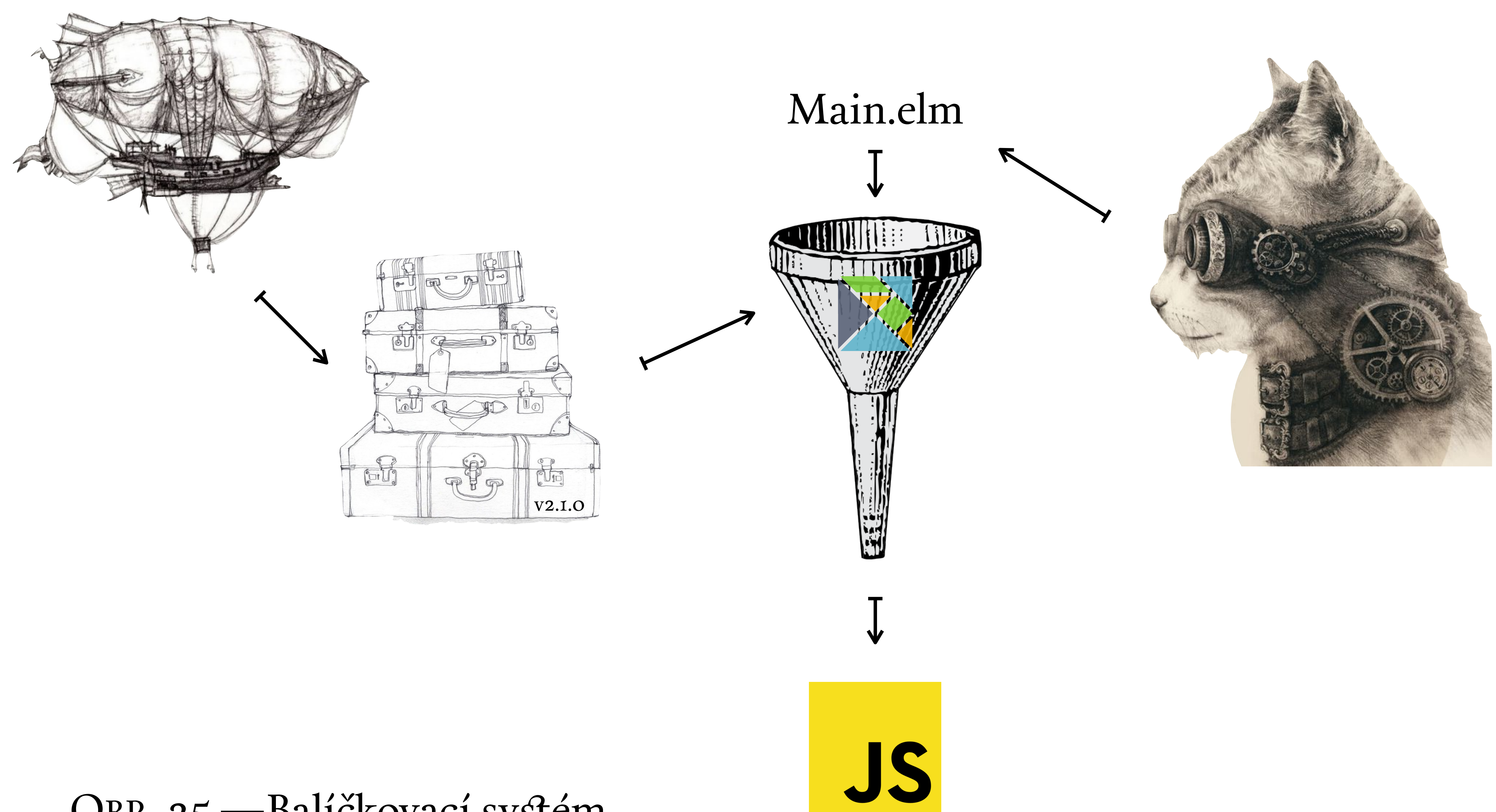
**Hint: To append strings in Elm, you need to use the
(++) operator, not (+).**
<<http://package.elm-lang.org/...>>



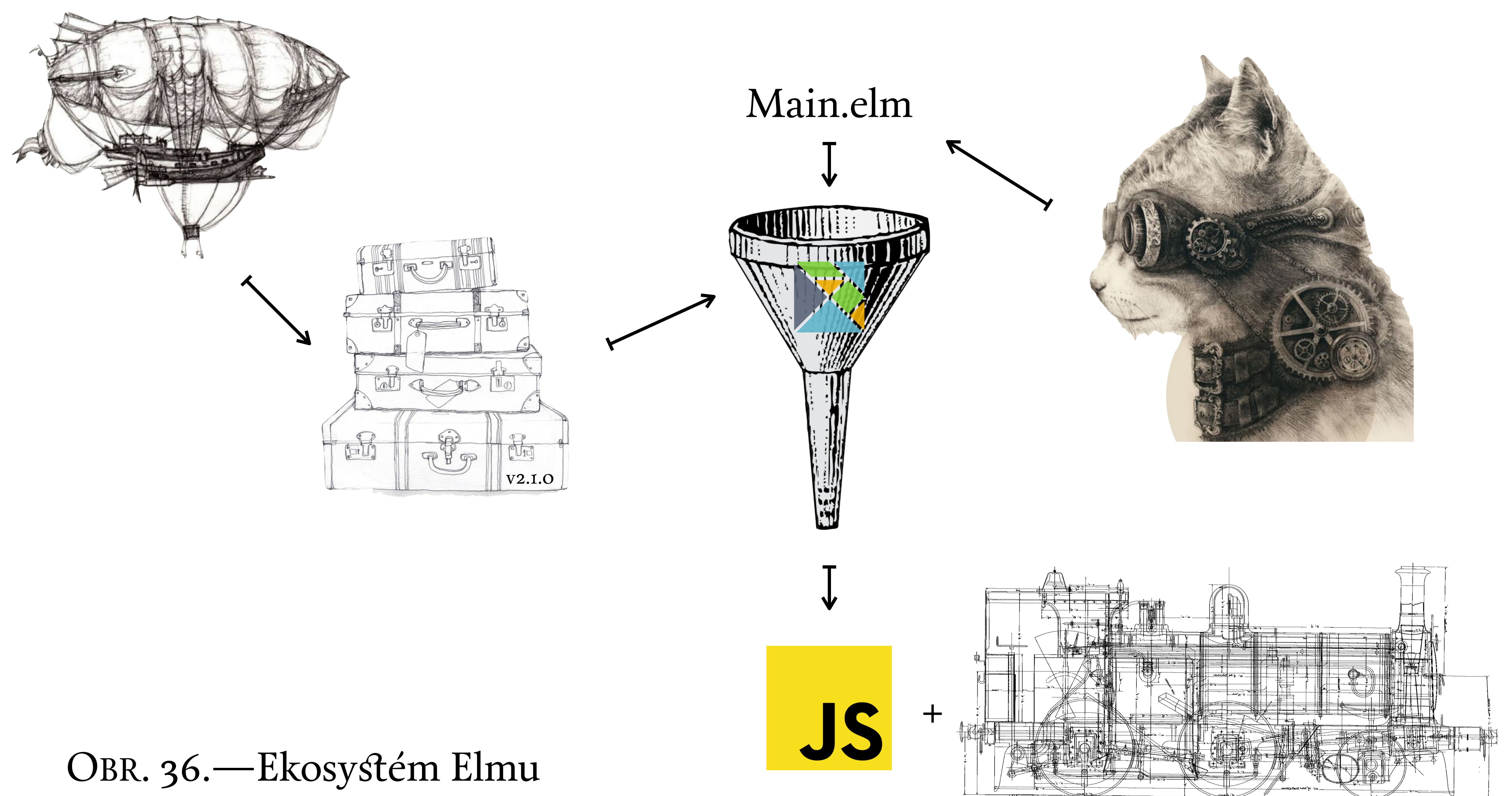
Main.elm



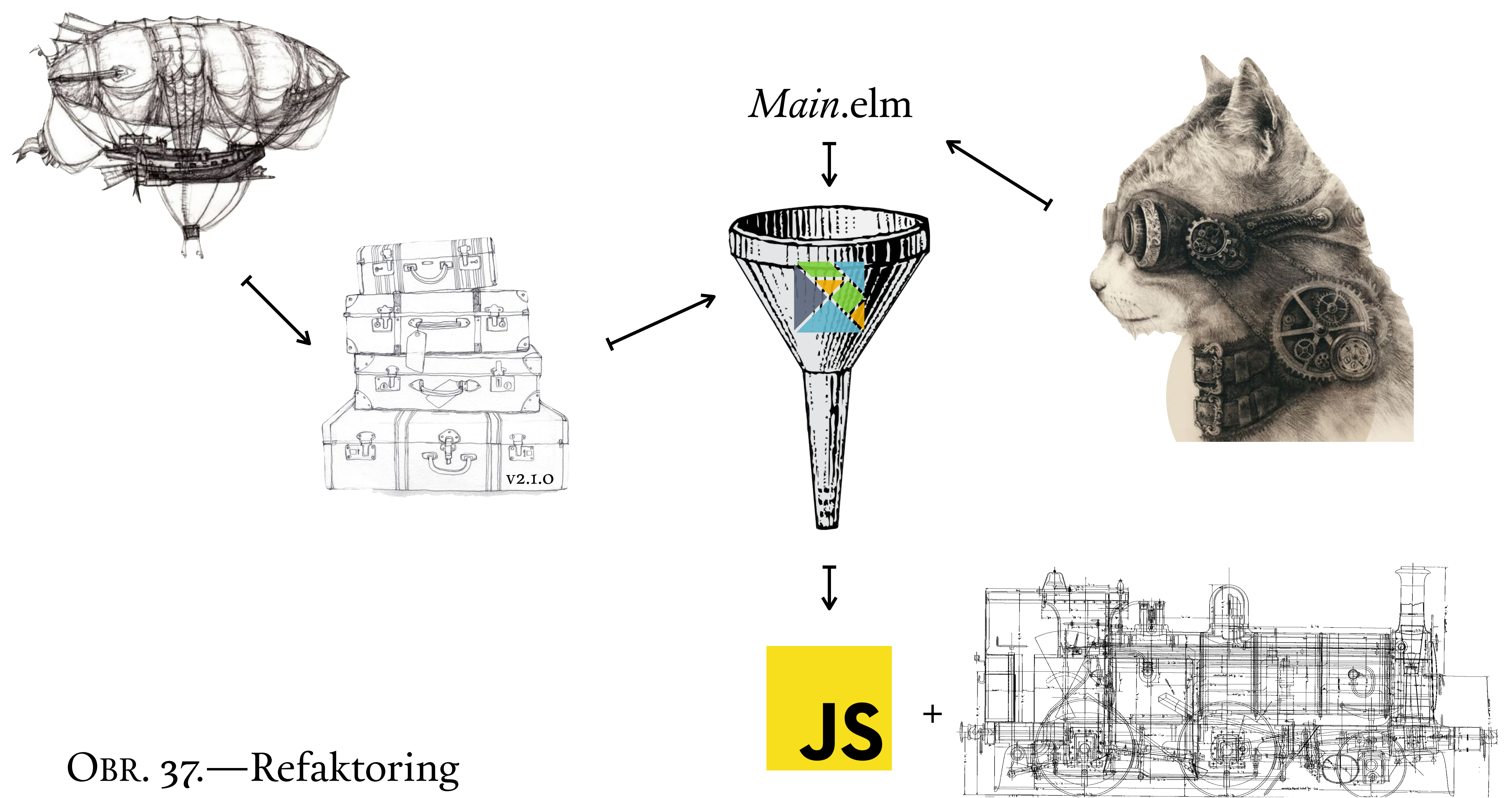
OBR. 34.—Navrácené miliony



OBR. 35.—Balíčkovací systém



OBR. 36.—Ekosystém Elmu



ROK DRUHÝ

Elm

jako

přístup


```
Date.now()
```

```
Math.random() * (6 - 1) + 1
```

```
const resp = await fetch(...)
```

```
element.focus()
```

```
window.innerWidth
```


`Date.now()`

„řekni mi čas“

`Math.random() * (6 - 1) + 1`

„hod' kostkou a řekni mi, co vyšlo“

`const resp = await fetch(...)`

„zaběhni na tuhle adresu
a doneš, co ti dají“

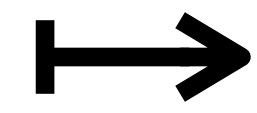
`element.focus()`

„přesuň kurzor támhle“

`window.innerWidth`

„změř mi aktuální šířku okna“

Cmd



Msg

Cmd



Msg

„hod’ kostkou a řekni mi, co vyšlo“

„výsledek hodu je 4“

Cmd



Msg

Random.generate **NewFace** (Random.int 1 6)

NewFace 4

Cmd



Msg

„Pro přednášku x mi z API zjisti
twitter přednášejícího
a řekni mi jej společně s x “

„Twitter přednášejícího
přednášky x je y “

Cmd



Msg

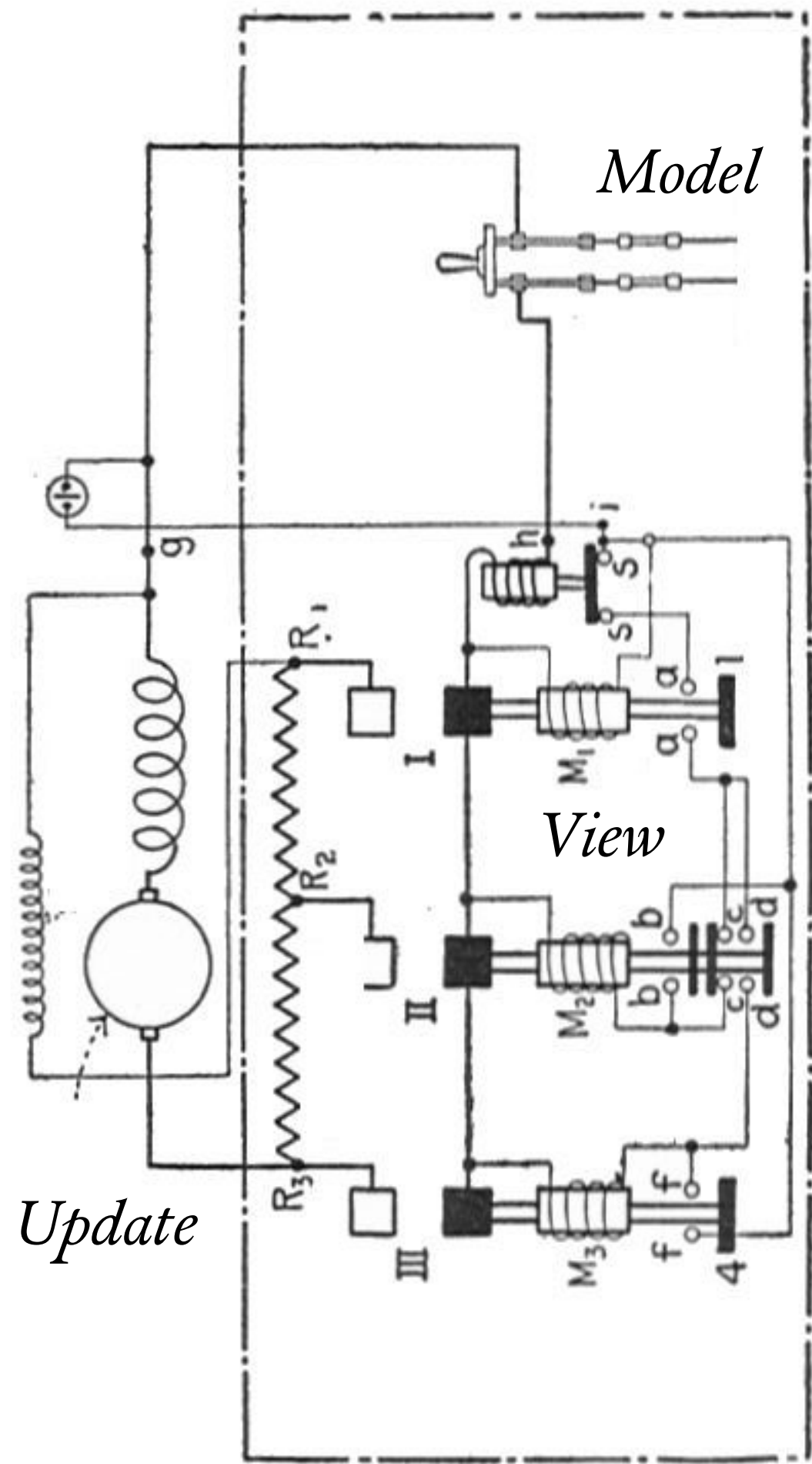
getSpeakerTwitter “elm”

SpeakerTwitter “elm” “@robinpokorny”

```
getSpeakerTwitter : String → Cmd Msg
getSpeakerTwitter talkId =
  let
    url = "https://api.devel.cz/konference/" ++ talkId

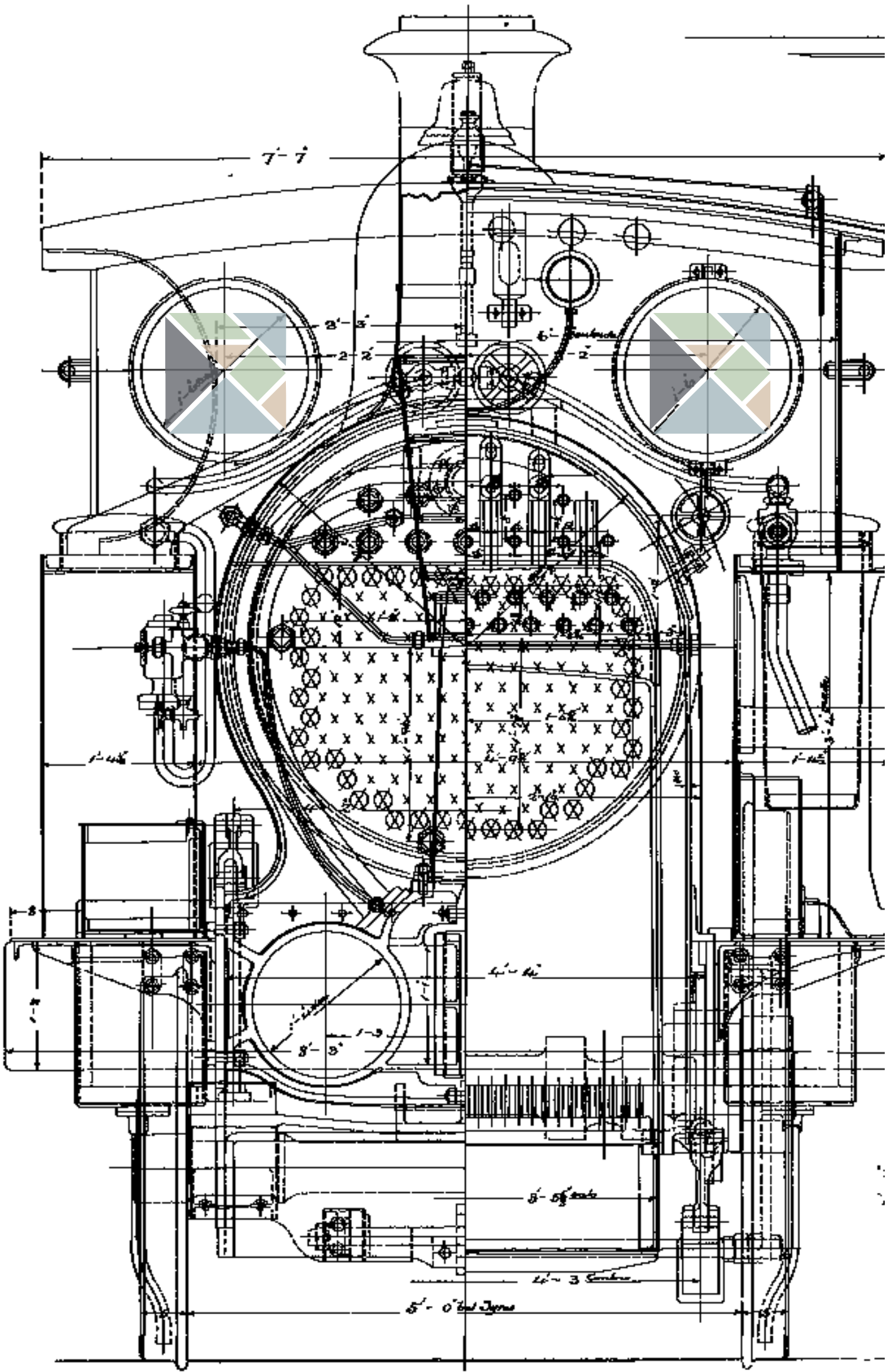
    request =
      Http.get url decodeTwitterUrl                                -- 1
  in
    Http.send (SpeakerTwitter talkId) request                      -- 2

decodeTwitterUrl : Decode.Decoder String
decodeTwitterUrl =
  Decode.at ["speaker", "twitter"] Decode.string                    -- 3
```

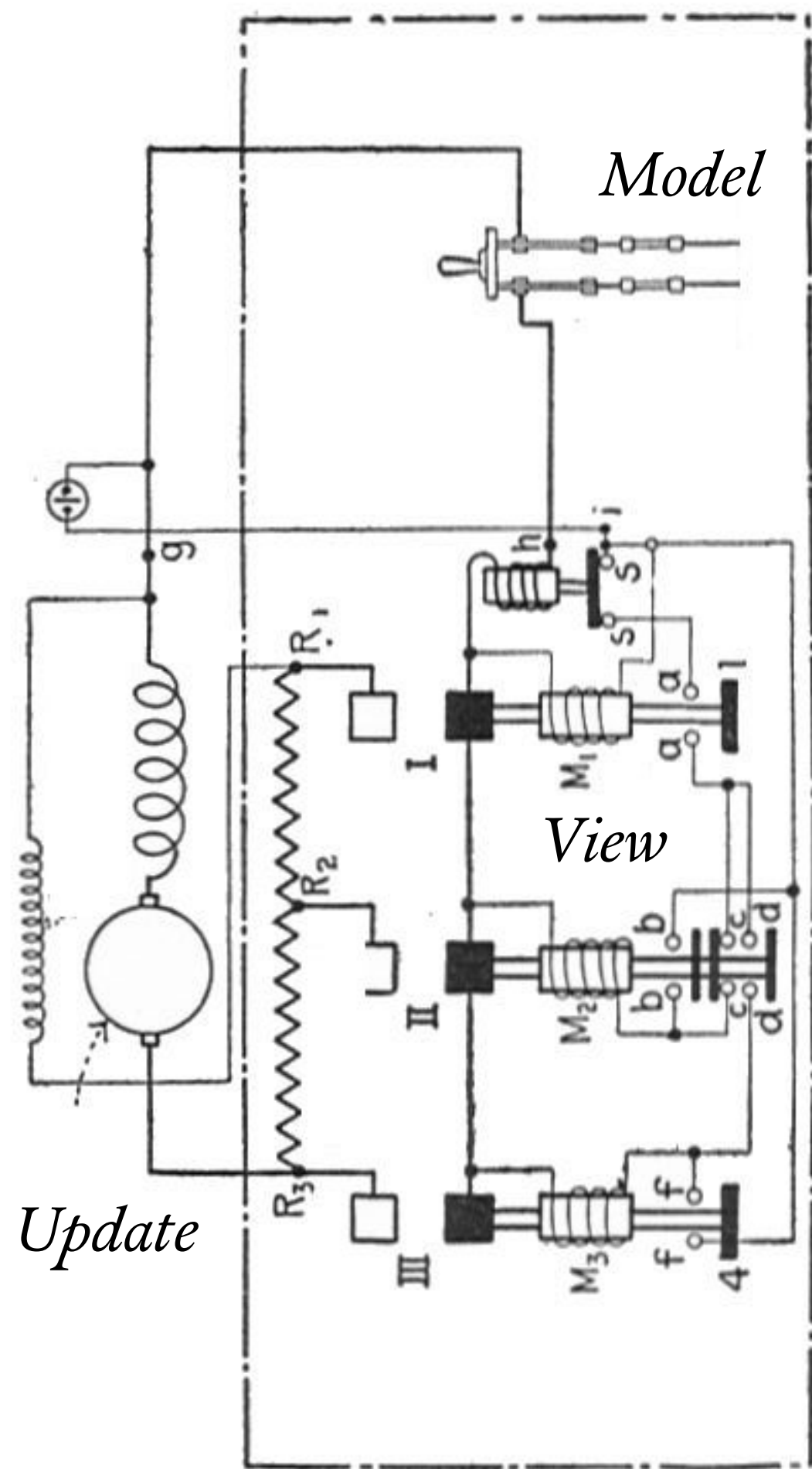



OBR. 42.—Aplikace

onClick Msg_I



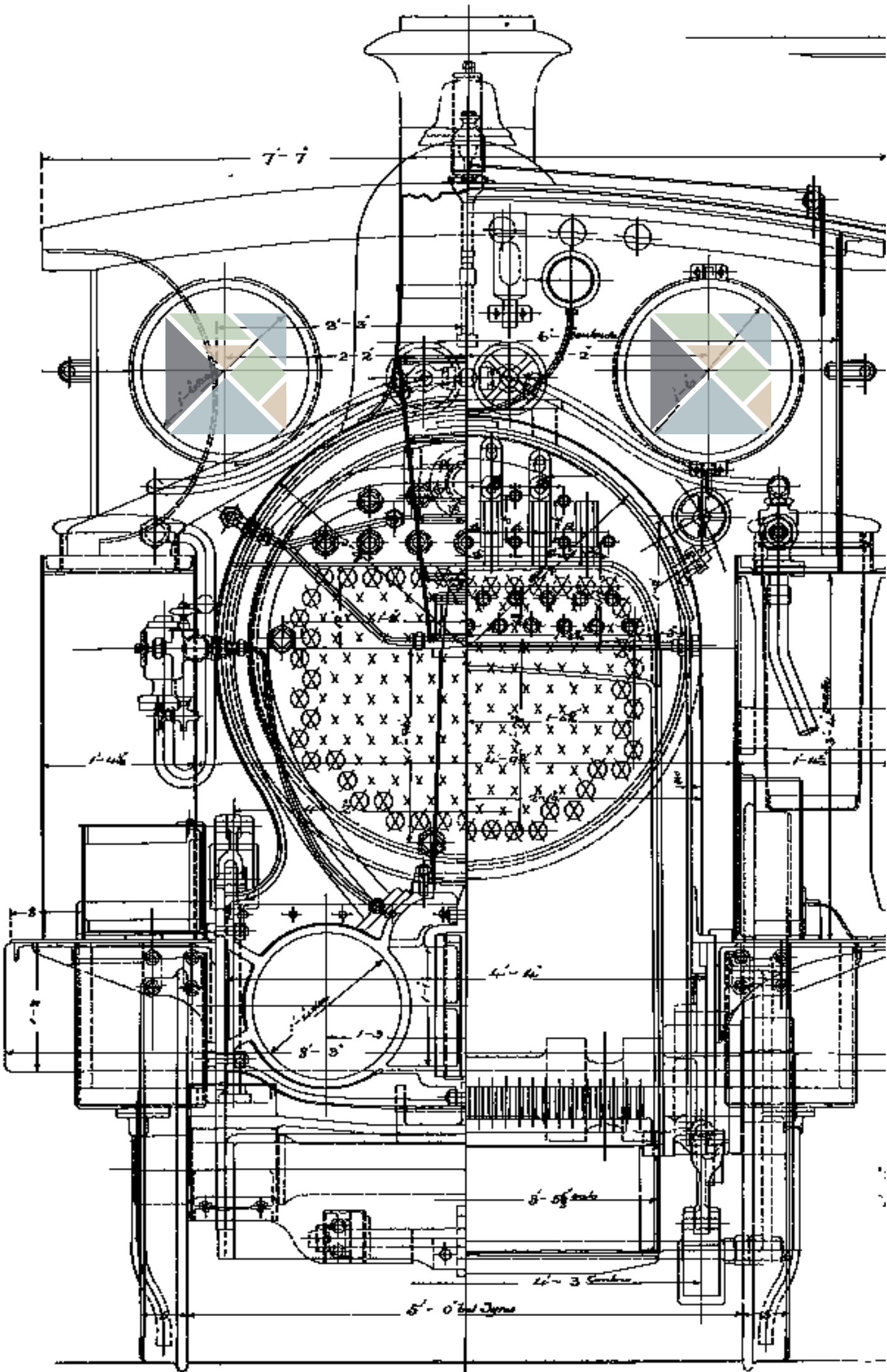
OBR. 18.—Elm Runtime



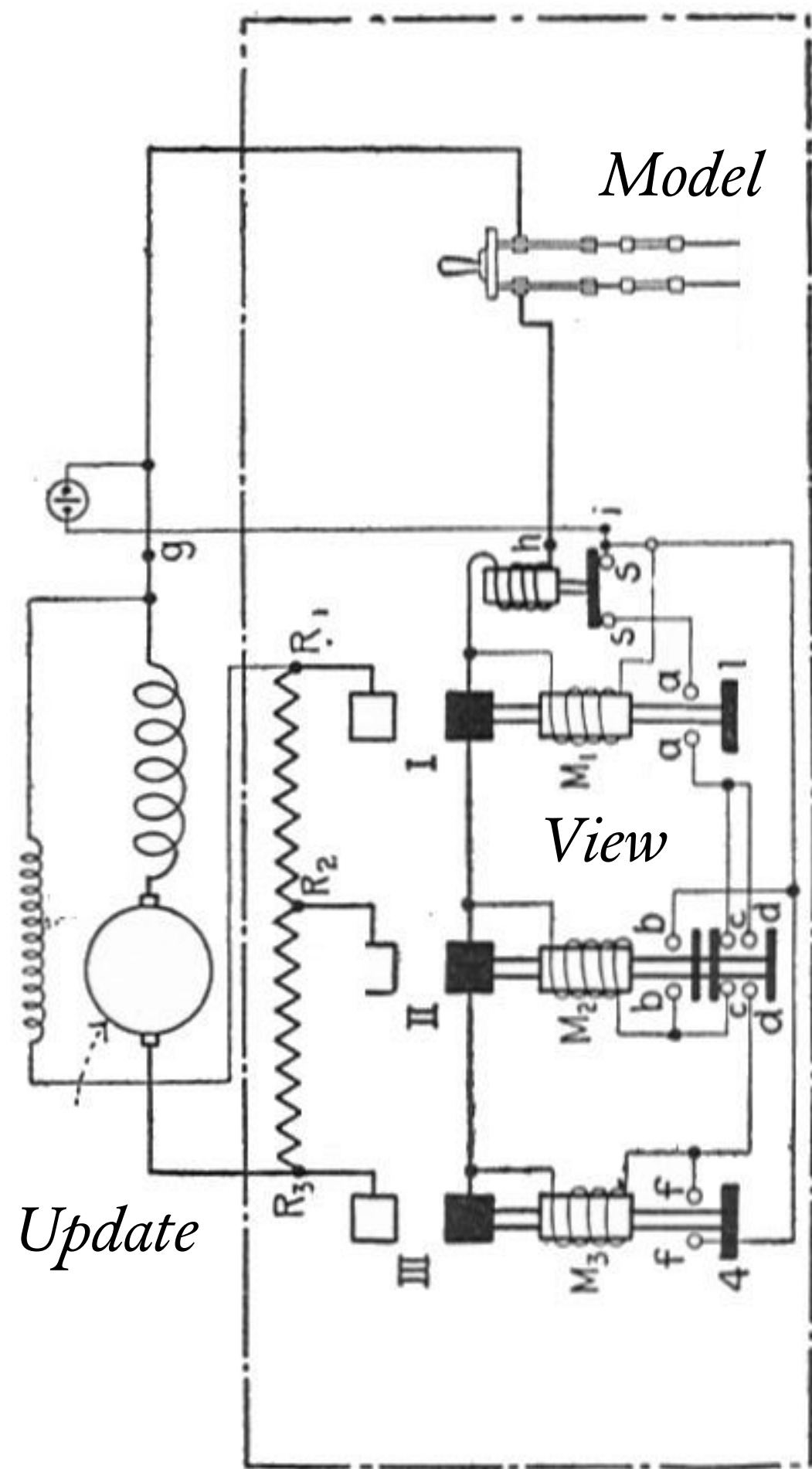
OBR. 42.—Aplikace

onClick Msg_I

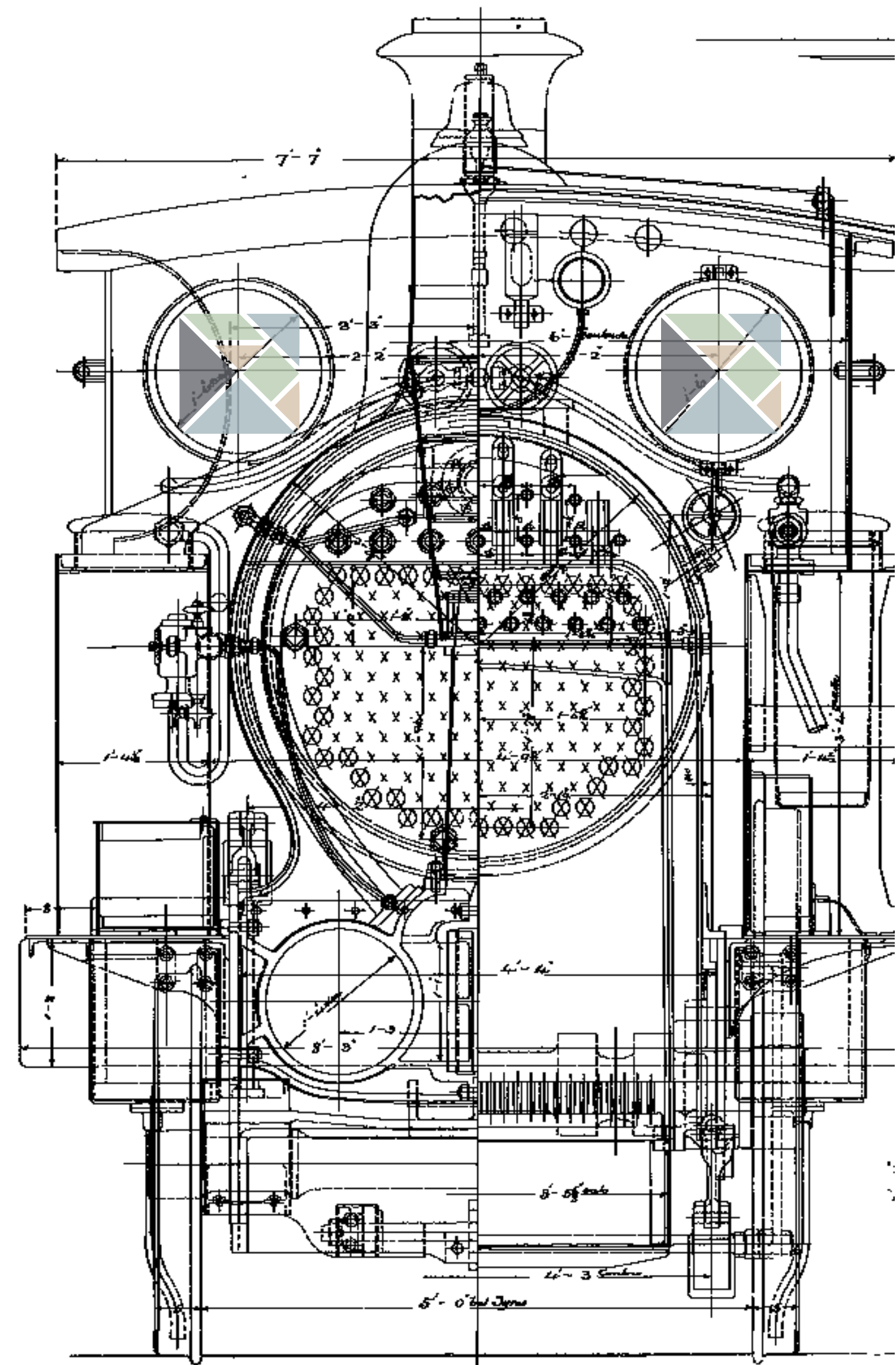
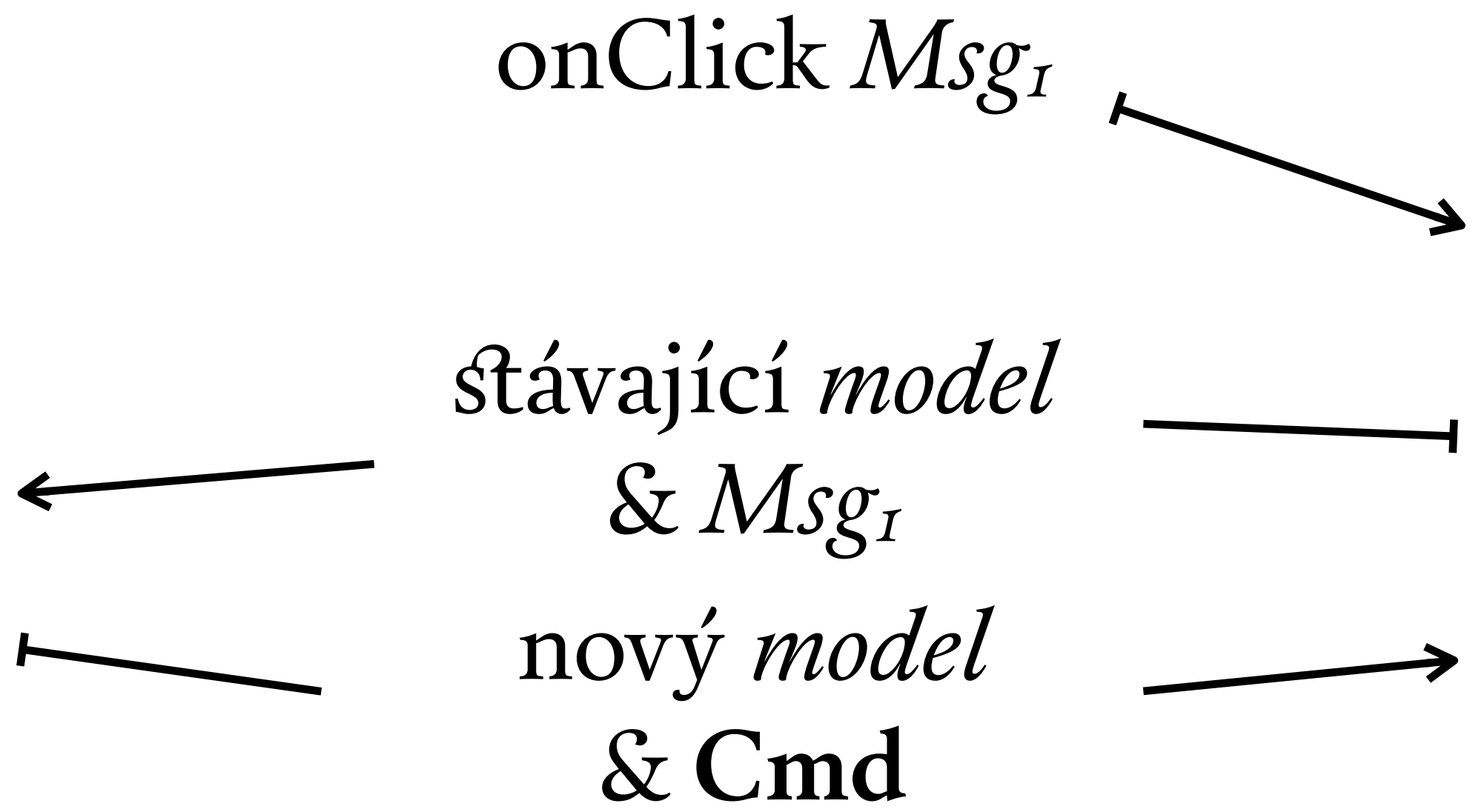
stávající *model*
& Msg_I



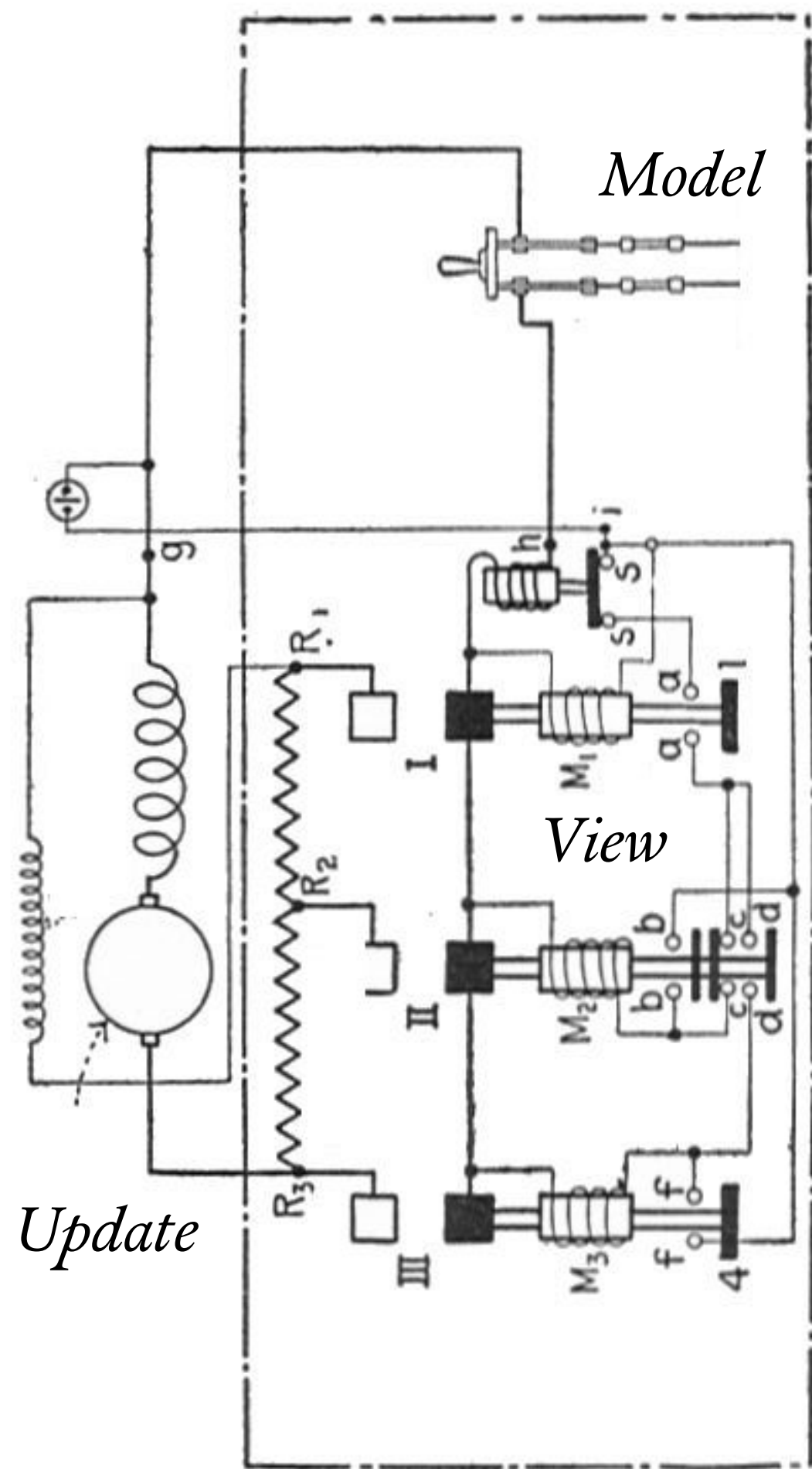
OBR. 18.—Elm Runtime



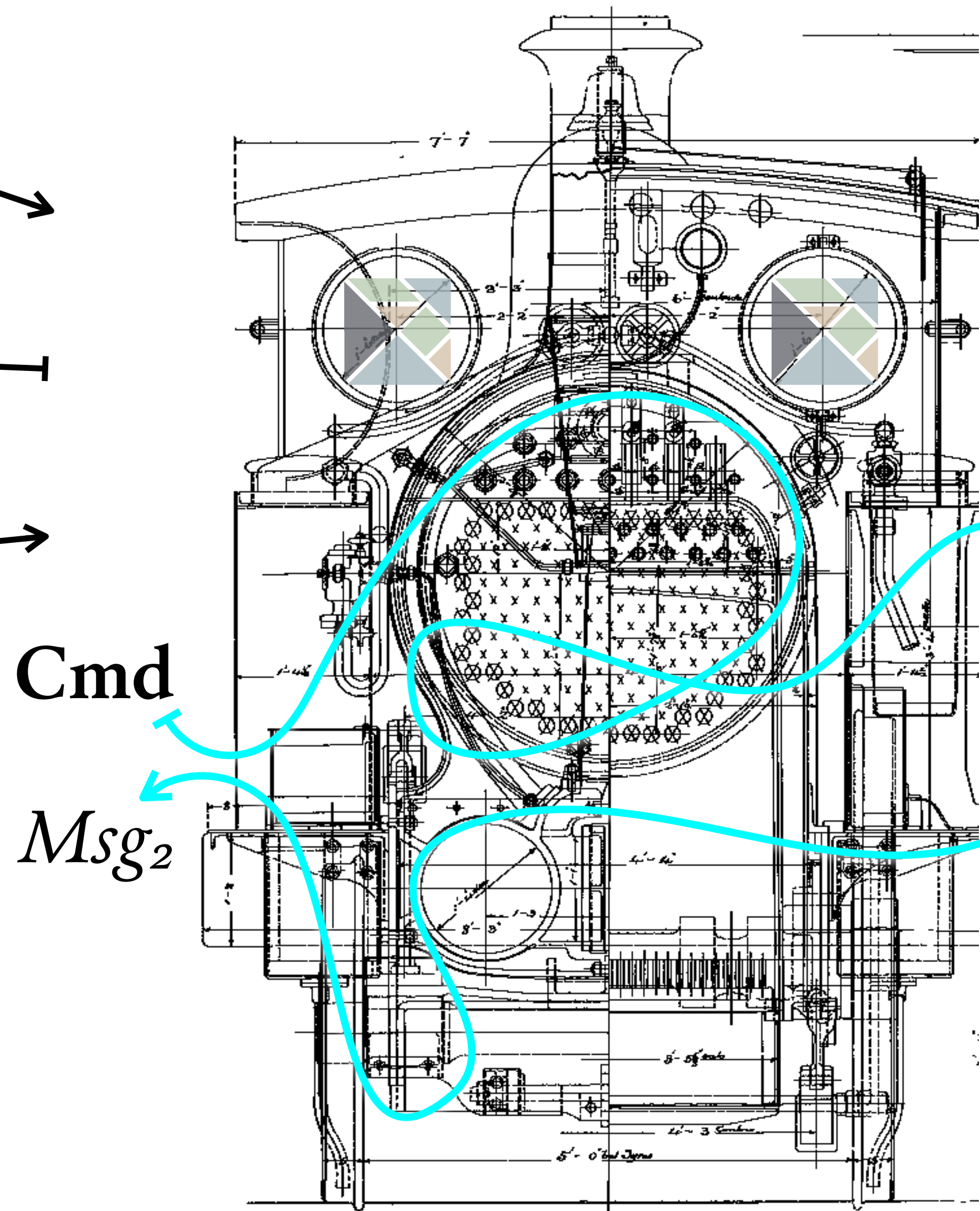
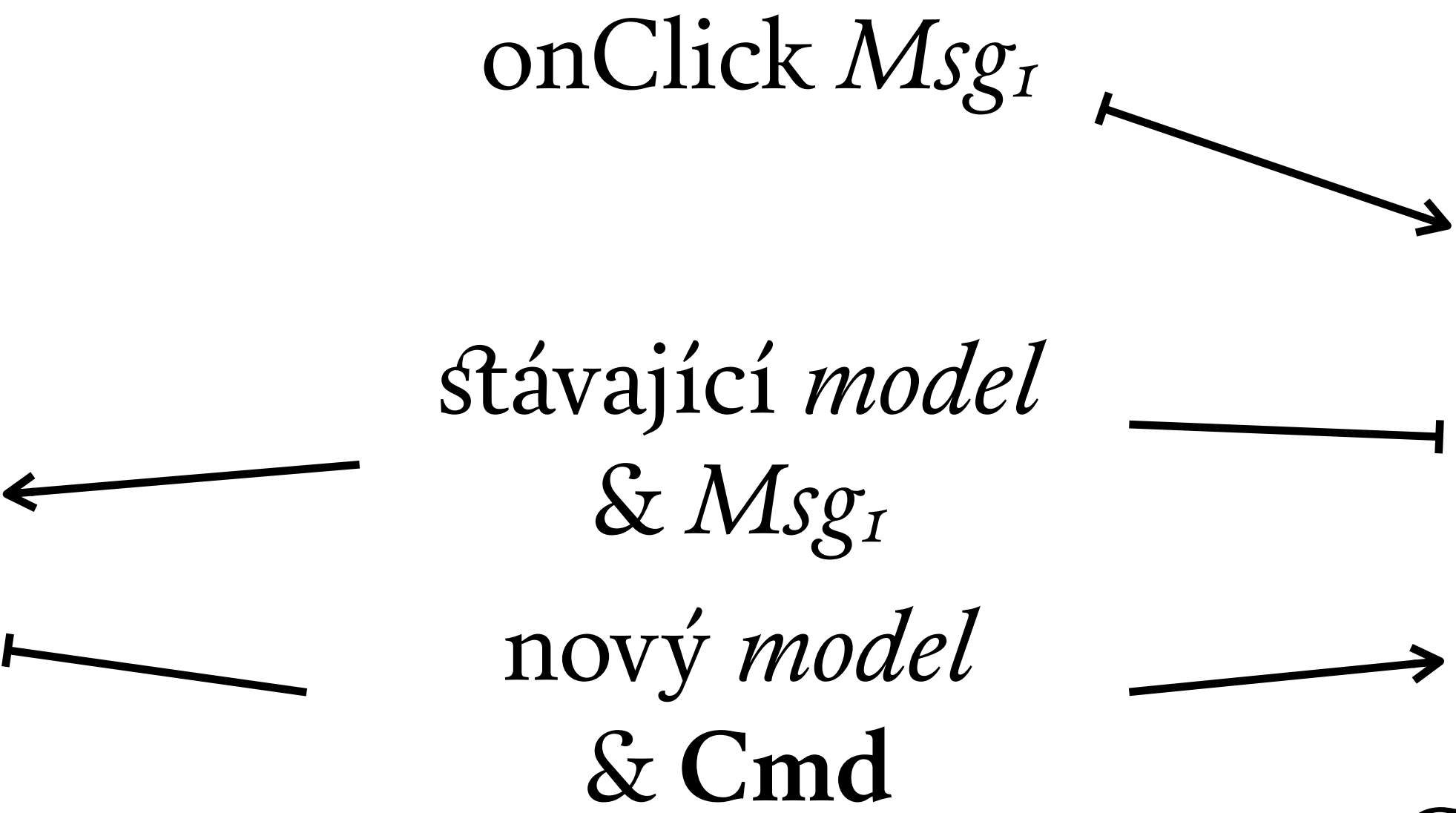
OBR. 42.—Aplikace



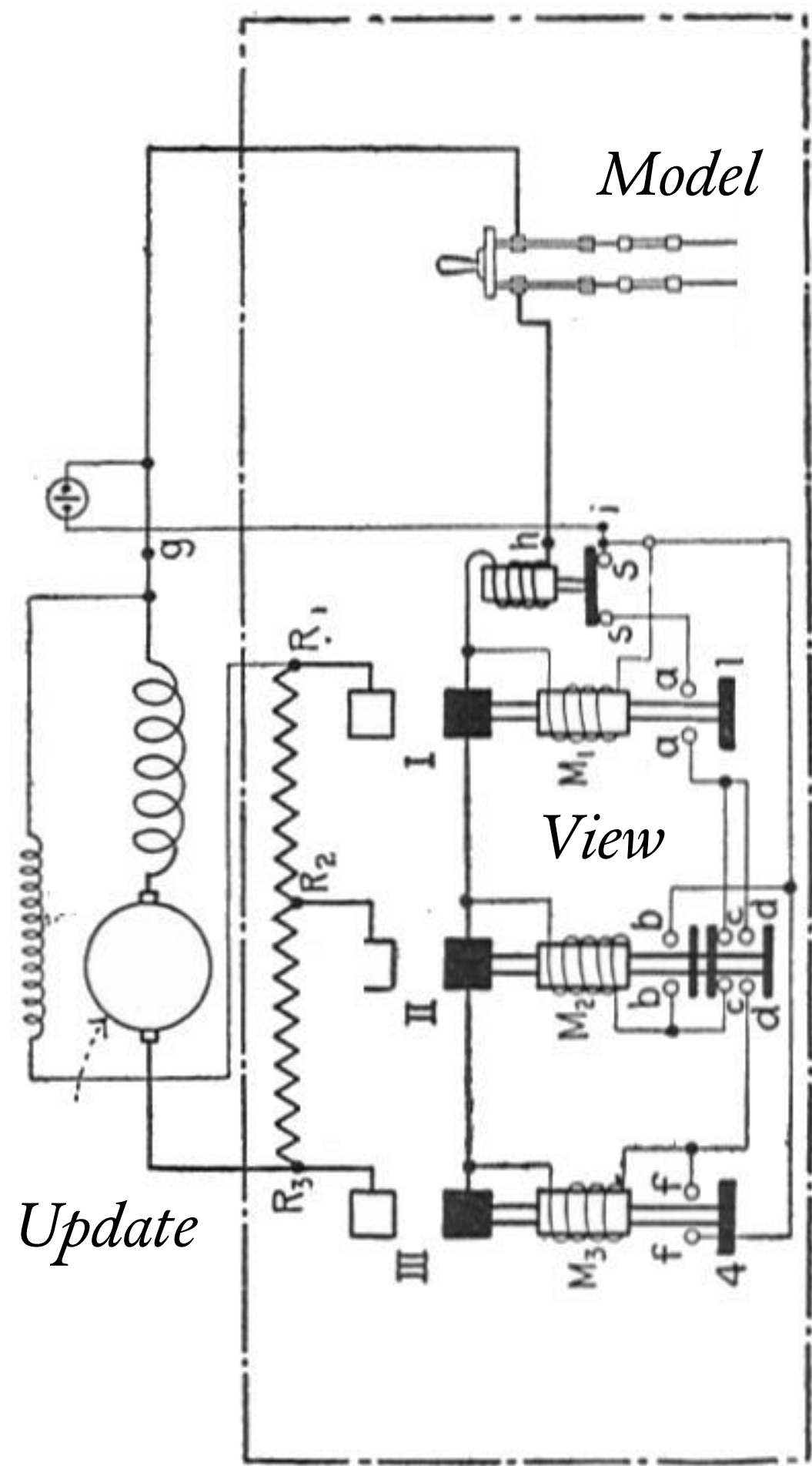
OBR. 18.—Elm Runtime



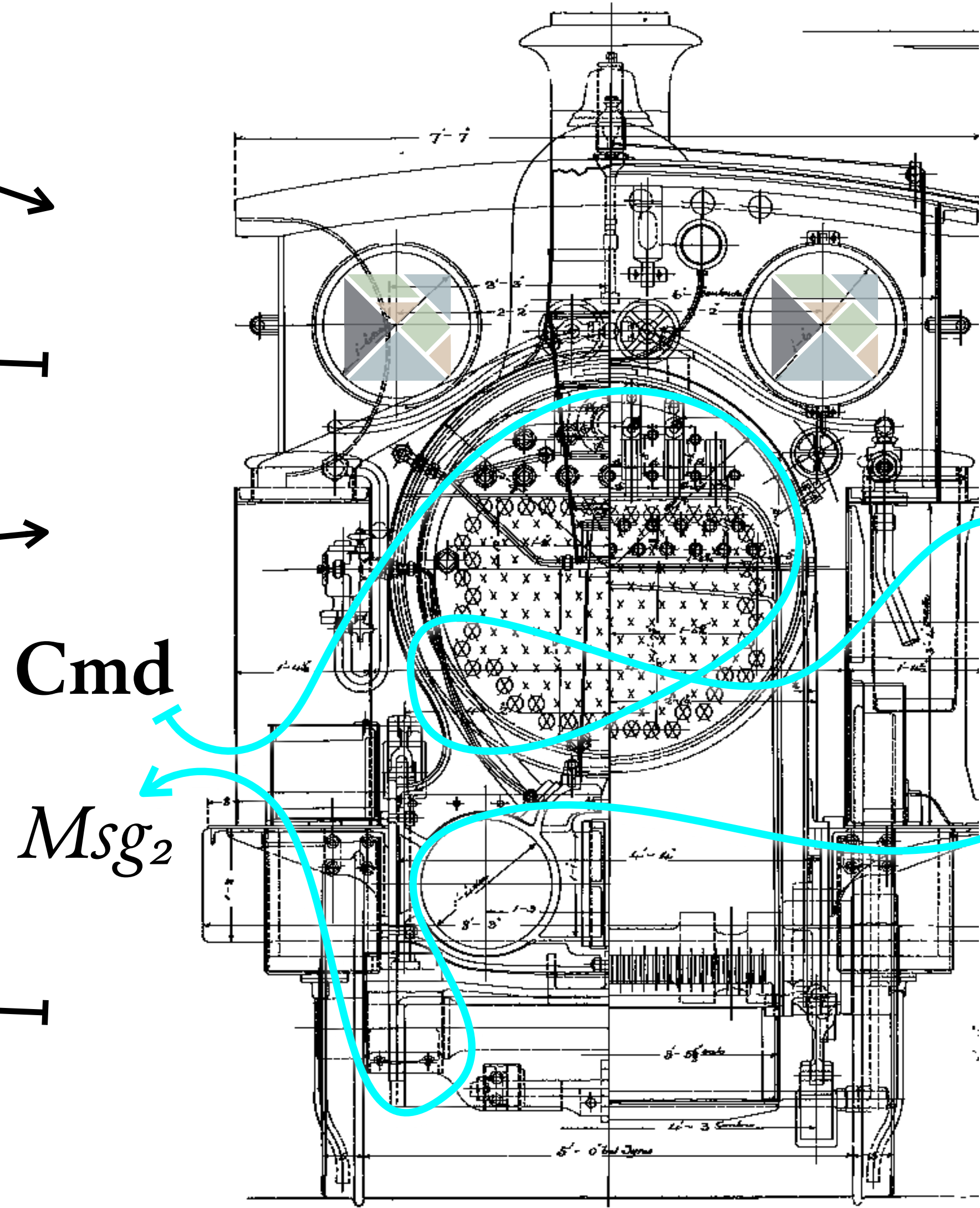
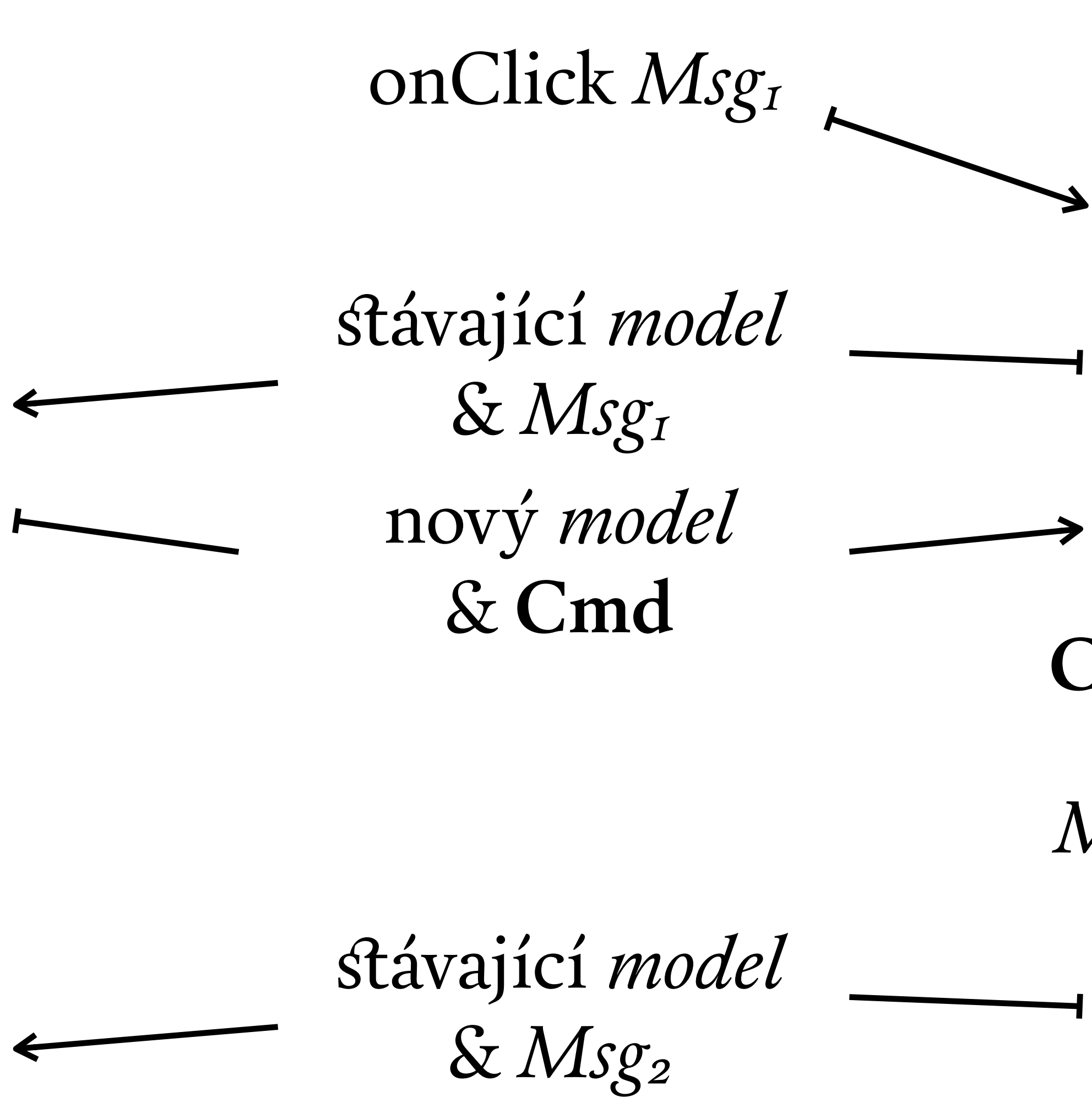
OBR. 42.—Aplikace



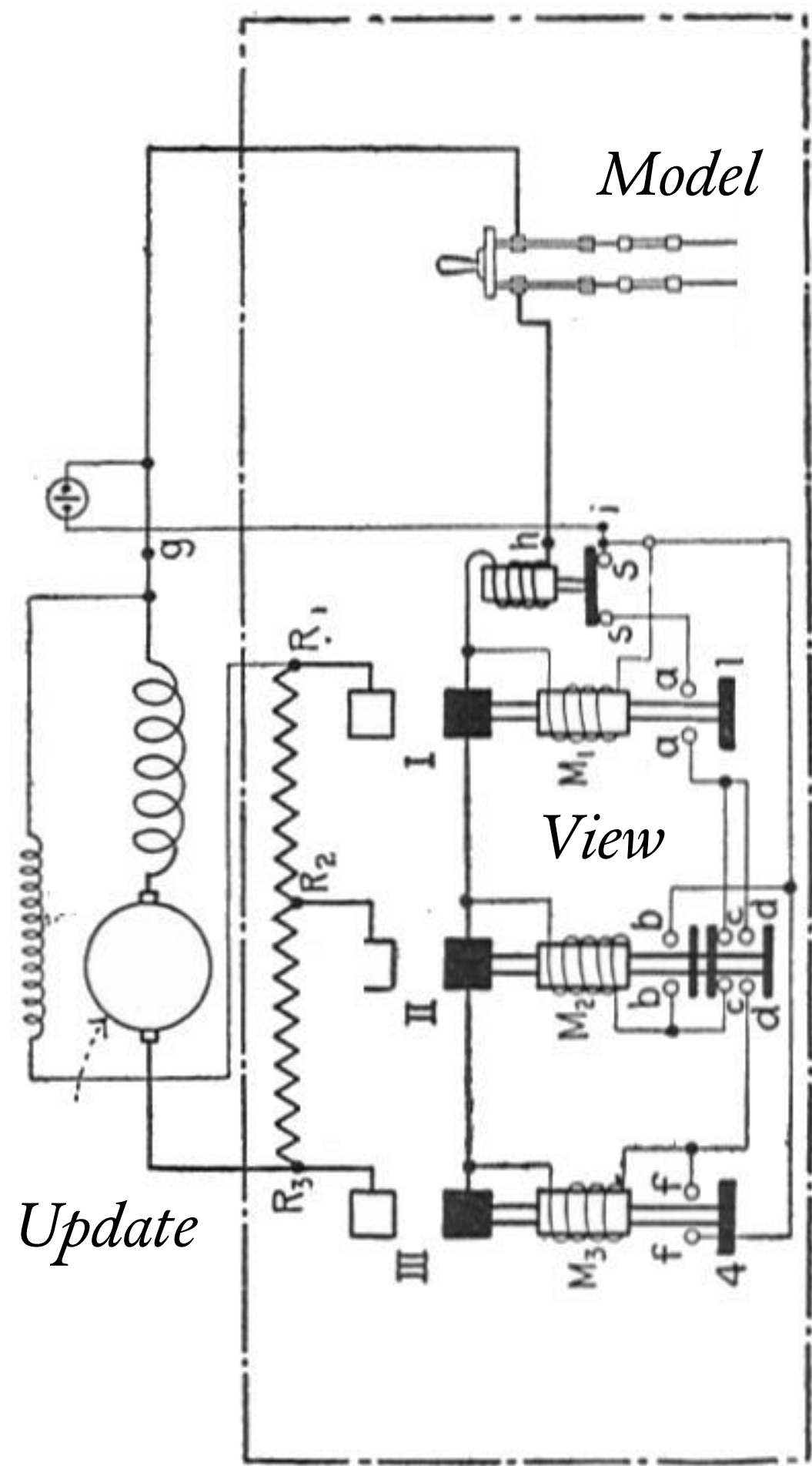
OBR. 18.—Elm Runtime



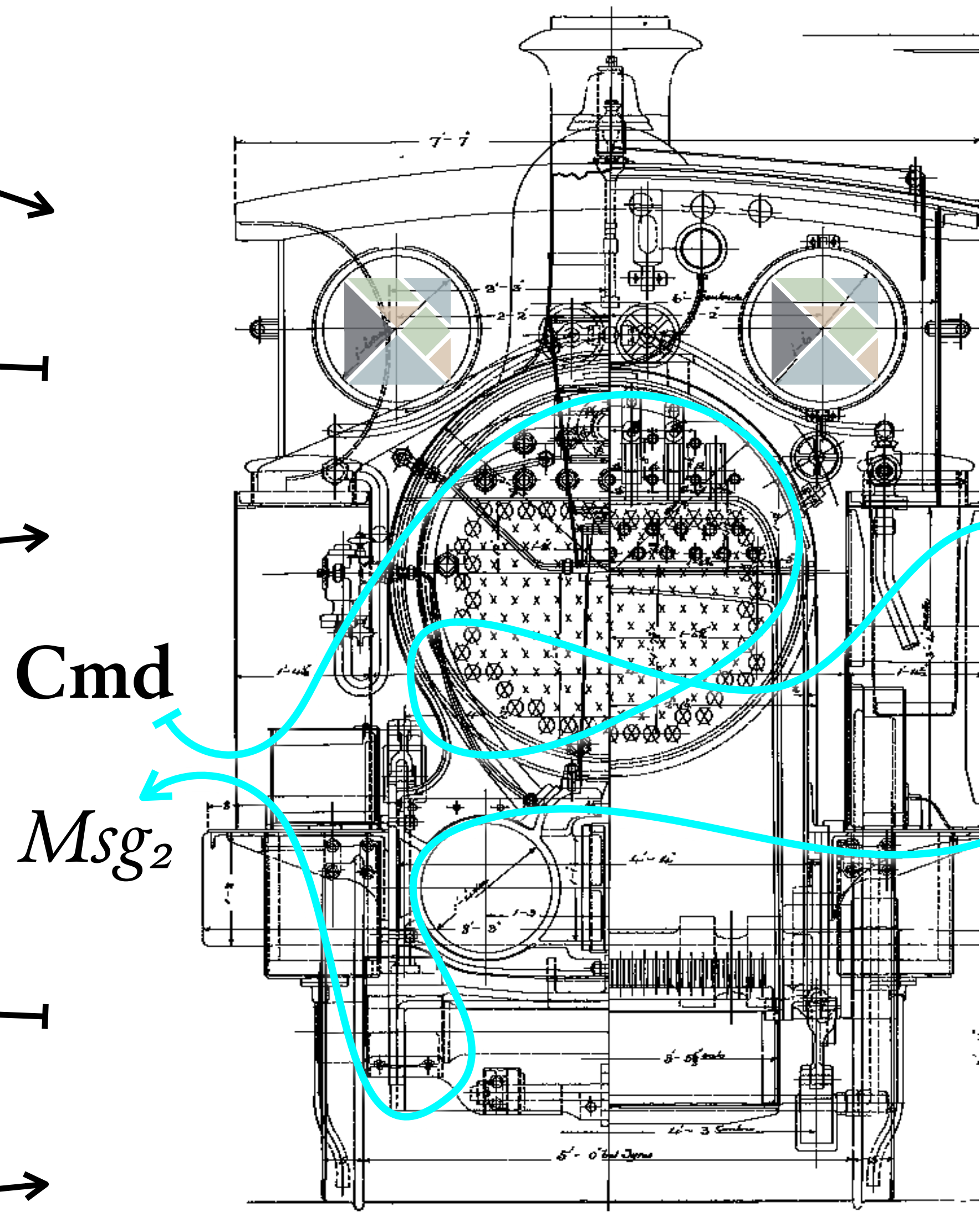
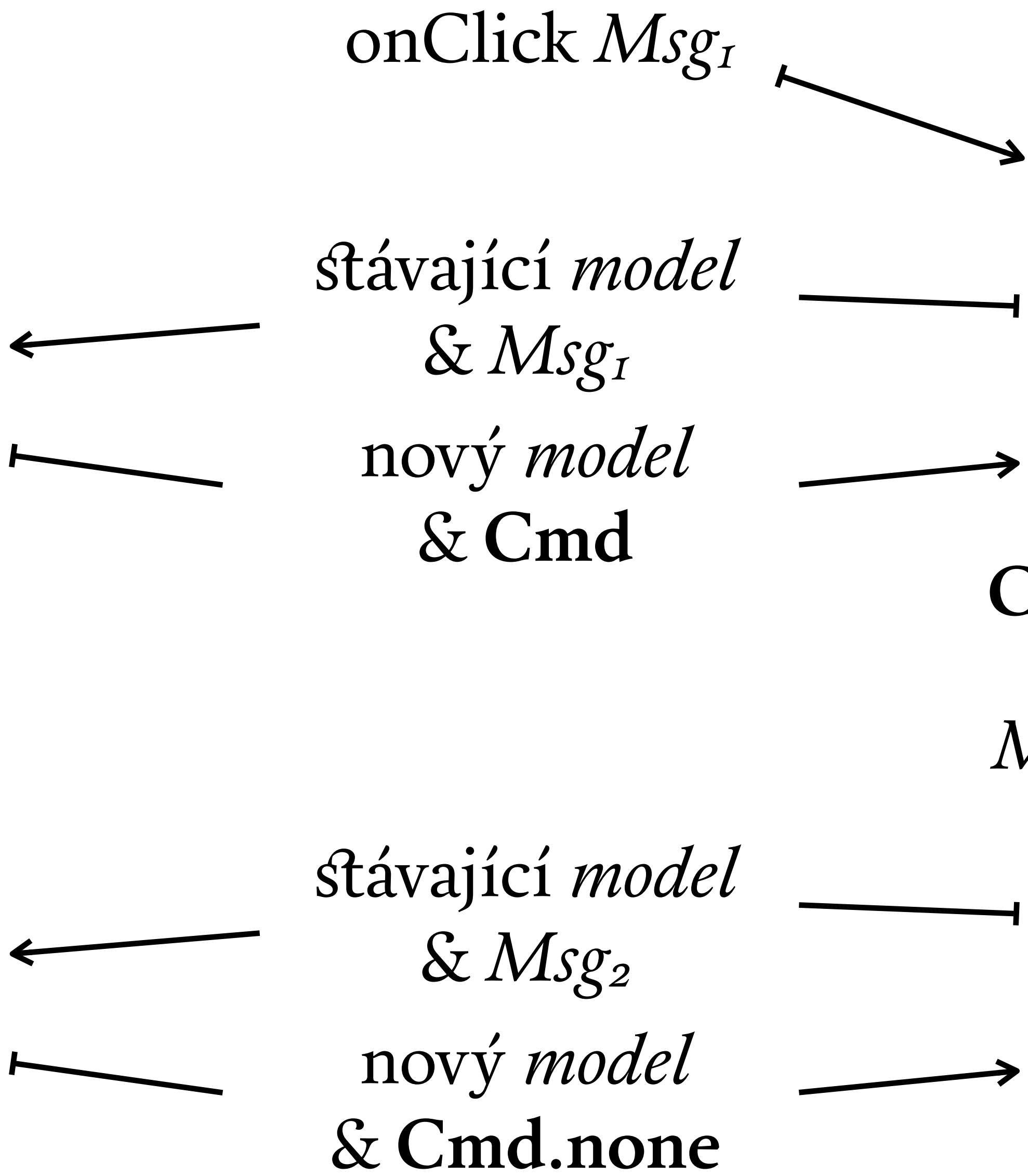
OBR. 42.—Aplikace



OBR. 18.—Elm Runtime



OBR. 42.—Aplikace



OBR. 18.—Elm Runtime

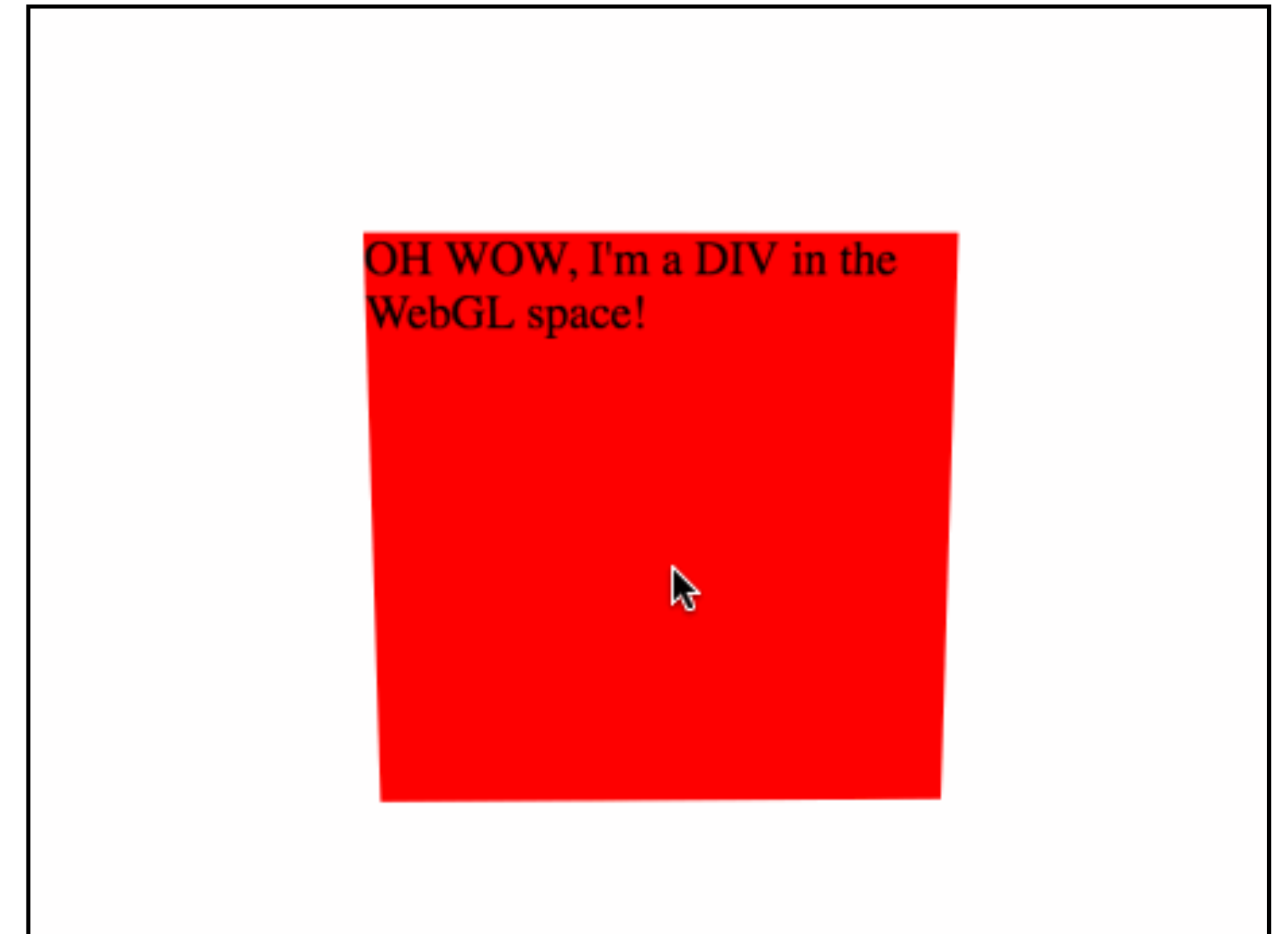
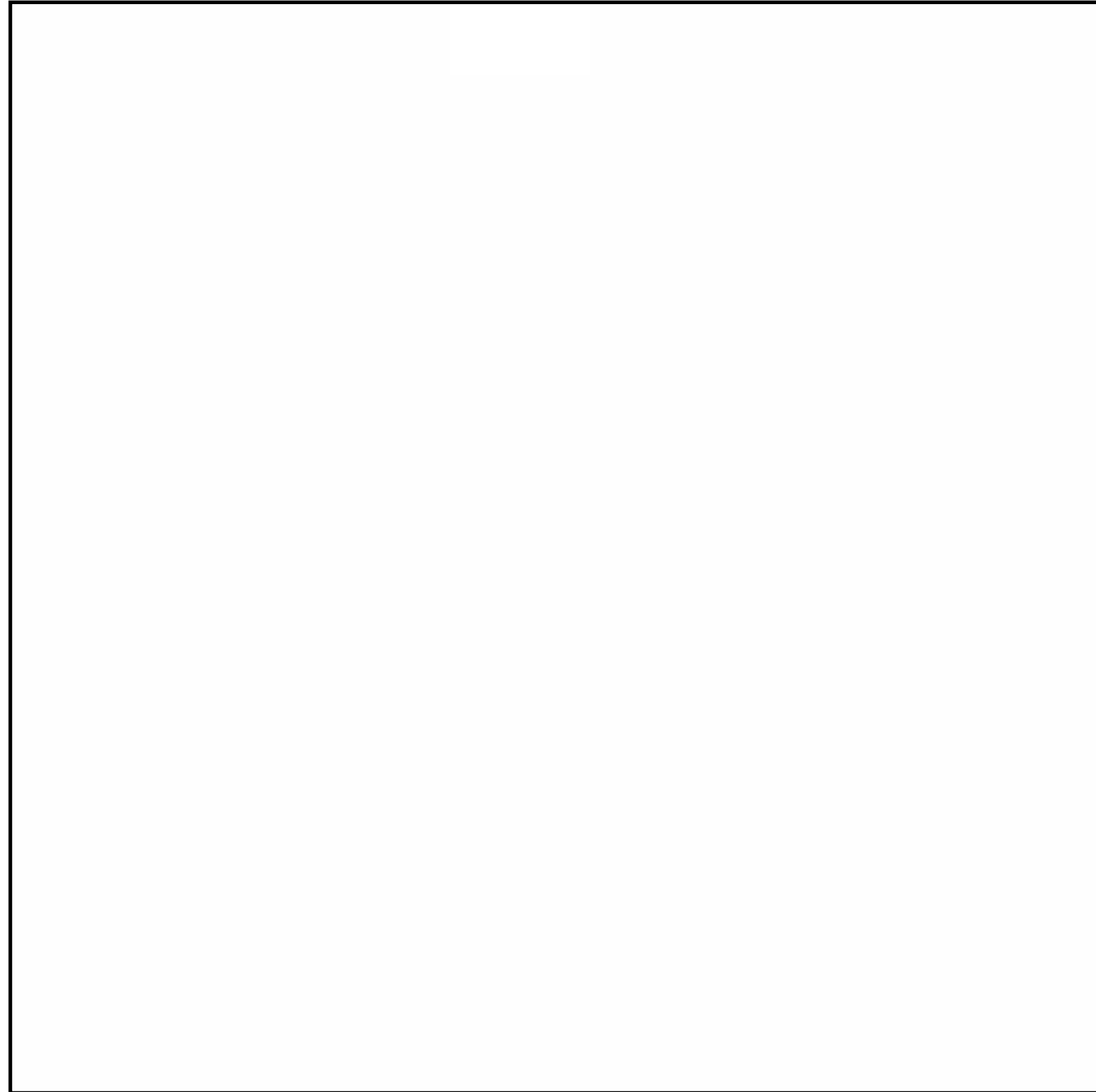
```
import Elm from 'react-elm-components'
import { Todo } from '../dist/elm/todomvc.js'

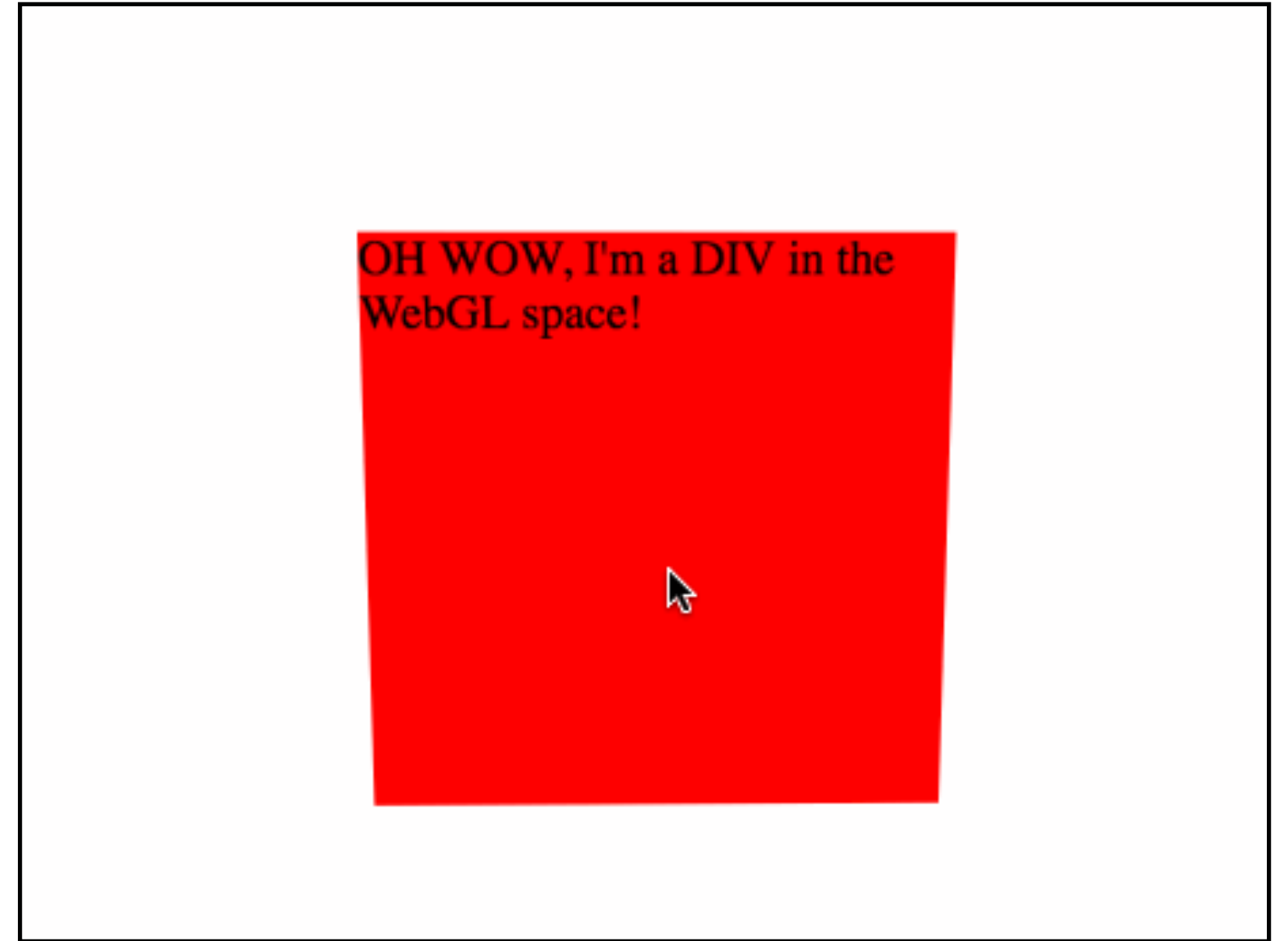
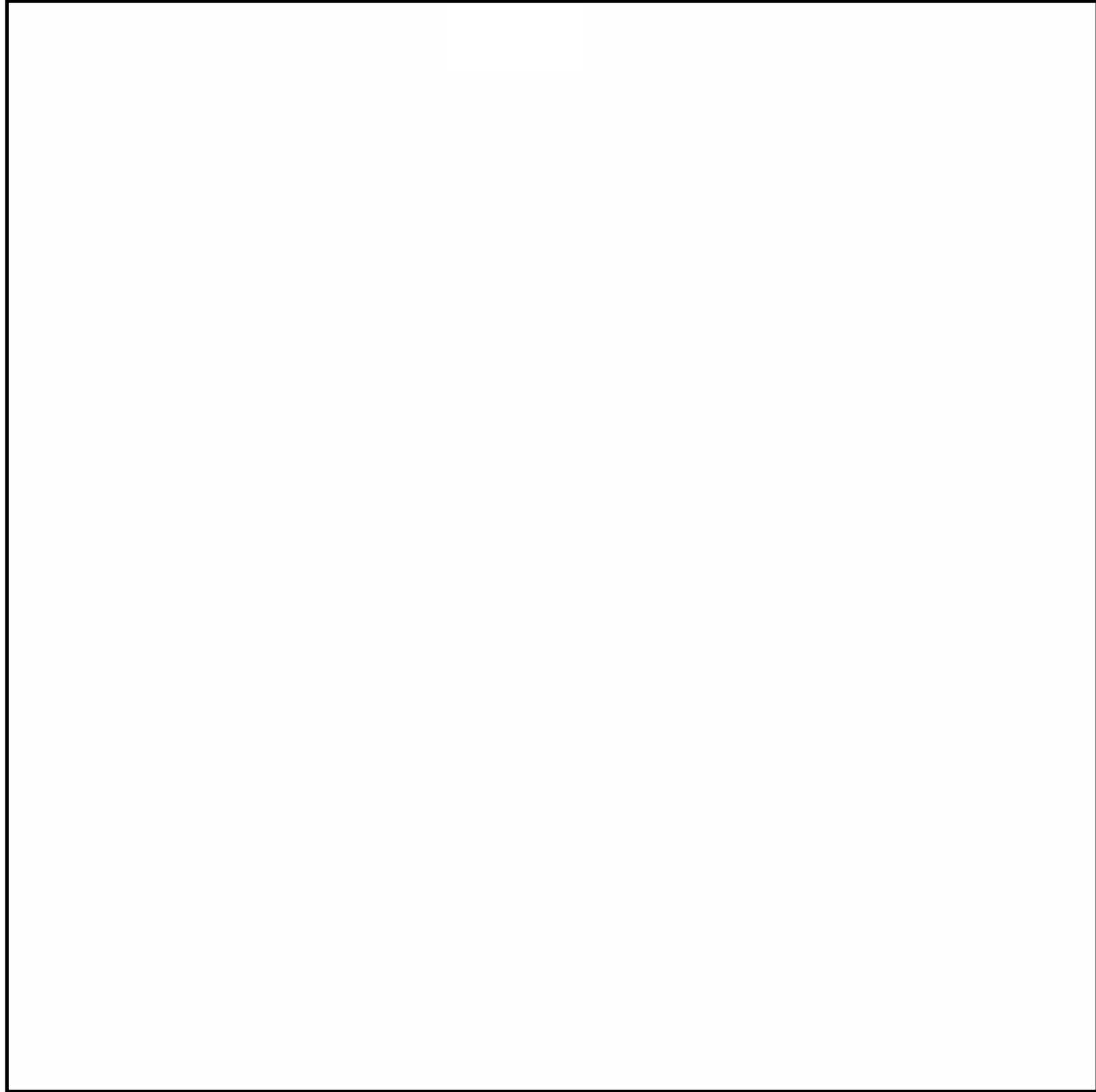
const setupPorts = (ports) => {
  ports.numActiveTodos.subscribe(n) => console.log(n))

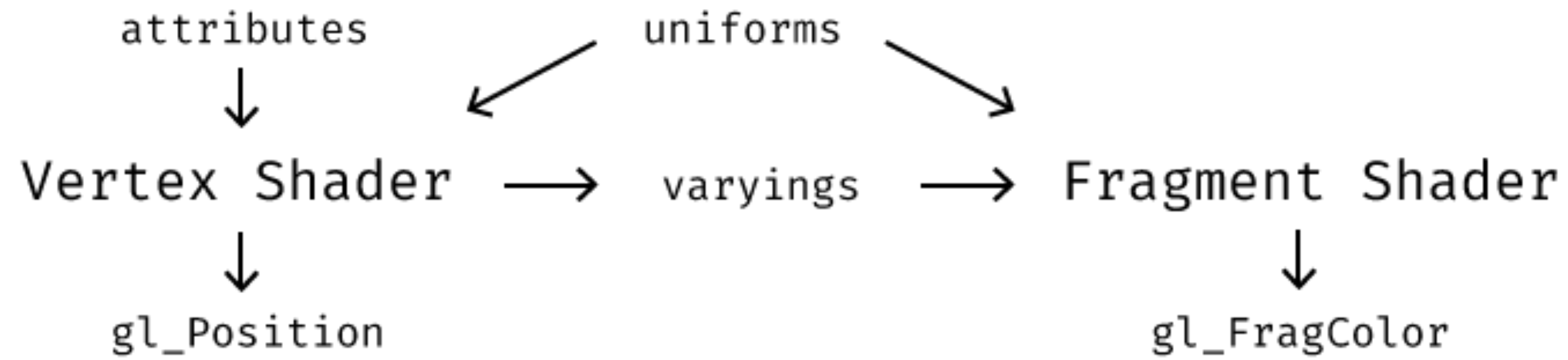
  ports.todos.send("Zarezervovat stůl v Čestru")
}

const TodoContainer = (list) => (
  <div>
    <h1>Hele, mami, Elm v Reactu!</h1>

    <Elm src={Todo} flags={{ todos: list }} ports={setupPorts} />
  </div>
)
```





Float64Array

```
type alias Attributes = { position : Vec3, color : Vec3 }  
type alias Uniforms = { rotation : Mat4 }  
type alias Varyings = { vColor : Vec3 }
```

```
vertexShader : Shader Attributes Uniforms Varyings
```

```
vertexShader = [glsl|  
    attribute vec3 position;  
    attribute vec3 color;  
    varying vec3 vColor;  
    uniform mat4 rotation;  
    void main () {  
        gl_Position = rotation * vec4(position, 1.0);  
        vColor = color;  
    }  
|]  
|]
```


view : Float → **Html** a

view angle =

WebGL.**toHtml**

[width 300

, height 300

, style ["display", "block"]

]

[WebGL.**entity**

vertexShader

fragmentShader

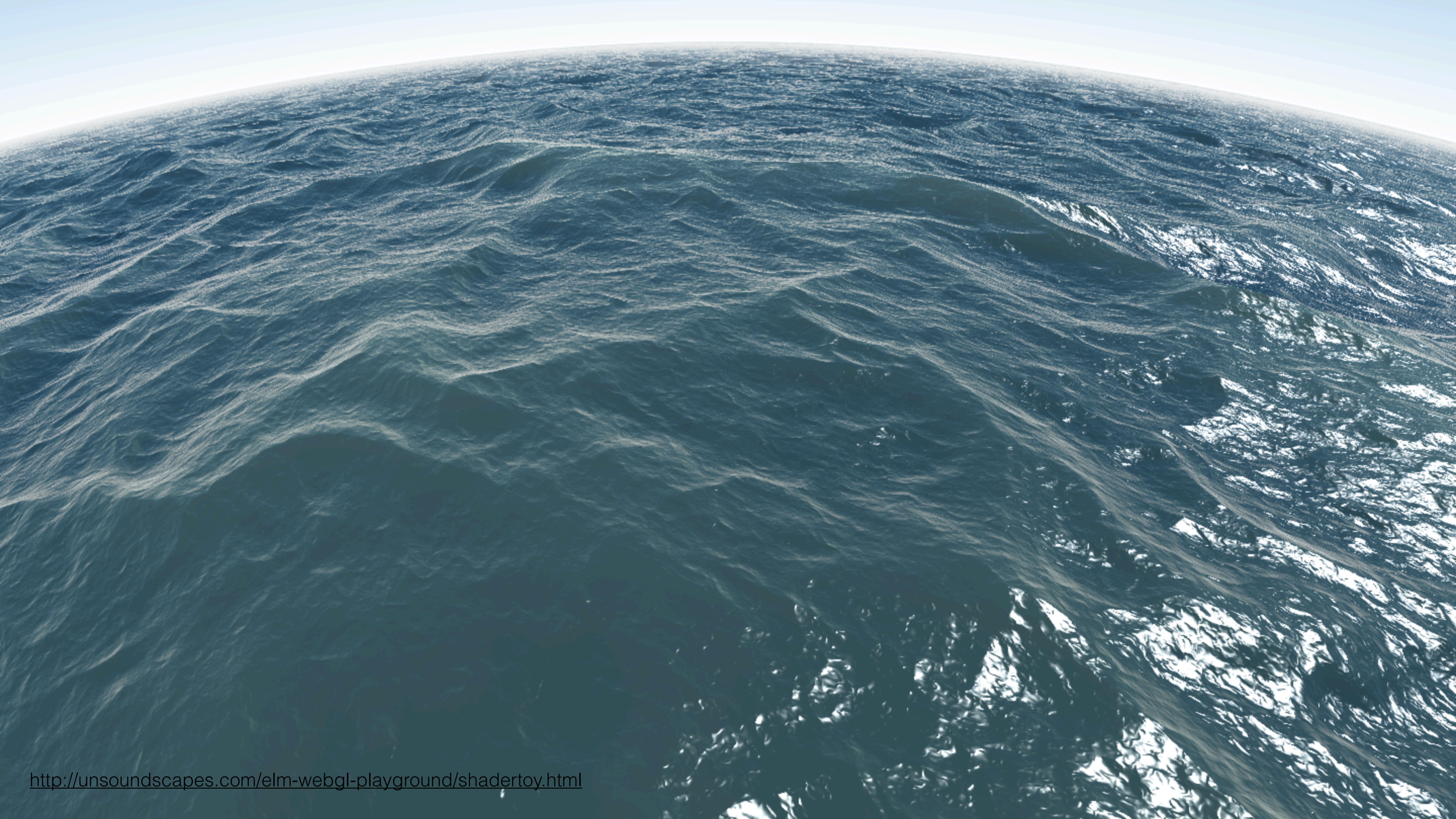
triangle

{ rotation = makeRotate **angle** (vec3 0 1 0) }

]

„Díky Elmu jsem pochopil WebGL.“

Andrey Kuzmin
maintaner elm-webgl







Im





elm
@robinpokorny