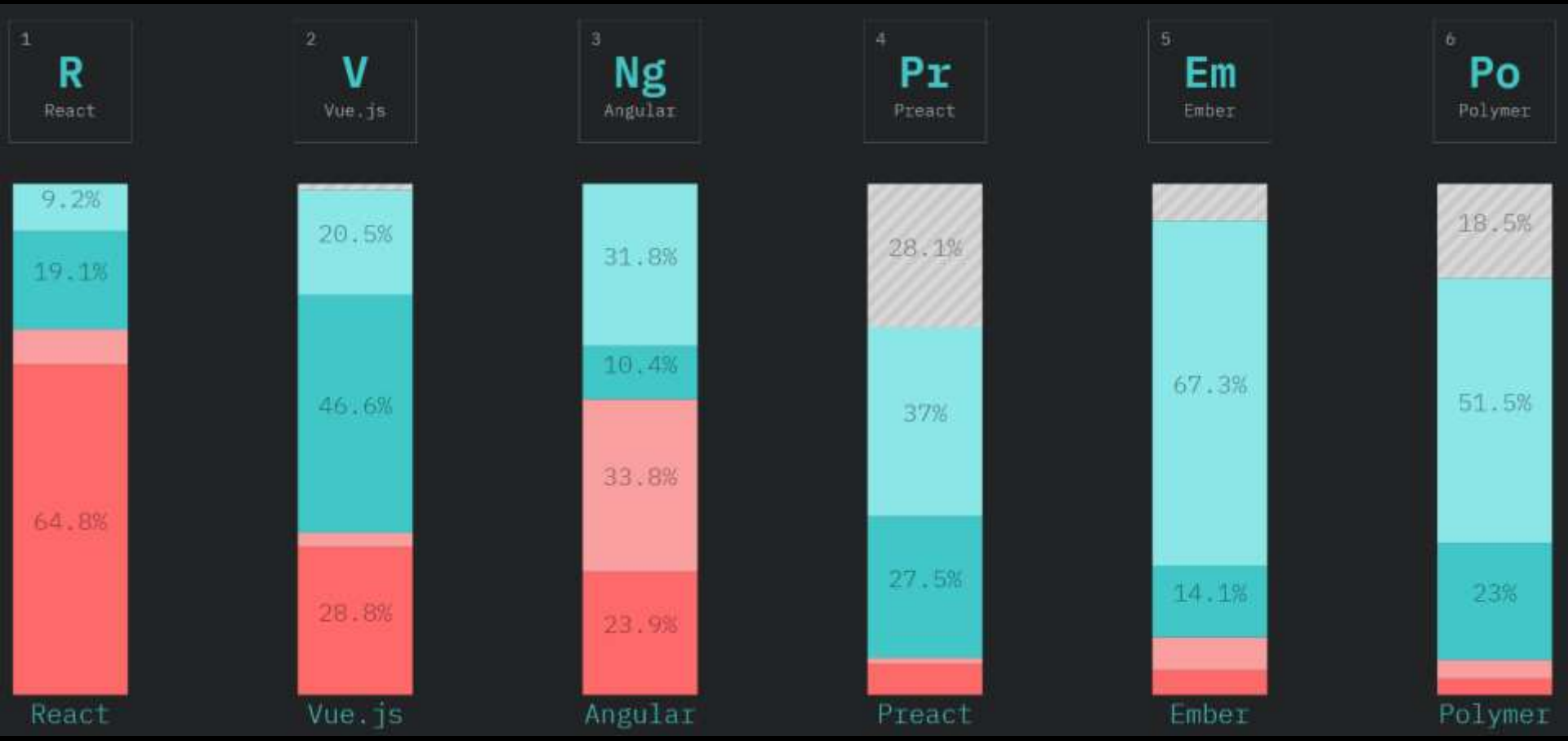
Dezvoltarea aplicațiilor Web la nivel de client



inginerie software în contextul JavaScript

"There are two ways to write error-free programs; only the third one works."

Alan J. Perlis

Ce instrumente software și biblioteci JS pot fi folosite în contextul programării Web la nivel de client?

instrumente

Editare de cod și dezvoltare de aplicații Web
Depanare
Testare
Documentare a codului
Compresie
Optimizare

Dr. Sabin Buraga ▲ profs.info.uaic.ro/~busaco

instrumente: editare

Editoare + medii integrate (IDE) pentru *desktop*:
Atom Editor, Brackets, JS Development Tools (Eclipse),
Sublime Text, Visual Studio Code, WebStorm etc.

Disponibile pe Web – în *cloud*:
Cloud9 IDE, Codepen, Codio, JS Bin, JS Fiddle, Plunker,…

unele oferă și partajarea codului-sursă cu alți dezvoltatori

a se considera și **github.com/sorrycc/awesome-javascript#editors**

instrumente: depanare

Utilizarea consolei JS

obiectul **console** oferit de *browser*

console.spec.whatwg.org (*Living Standard*, 25 oct. 2018)

metode utile: **log ()**, **error ()**, **warn ()**, **info ()**, **assert ()**, **time ()**, **timeEnd ()**, **trace ()**, **group ()**, **groupEnd ()**, **table ()**, **dir ()**, **dirxml ()**

developer.mozilla.org/Web/API/Console

Basic console logging functions - LS

Usage | % of all users

Global | 91.12% + 0.27% = 91.39%

Method of outputting data to the browser's console, intended for development purposes.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

de studiat și M. Burgess, *Beyond console.log()*, nov. 2018

medium.com/@mattburgess/beyond-console-log-2400fdf4a9d8

Notes | Sub-features (1) | Known issues (0) | Resources (4) | Feedback

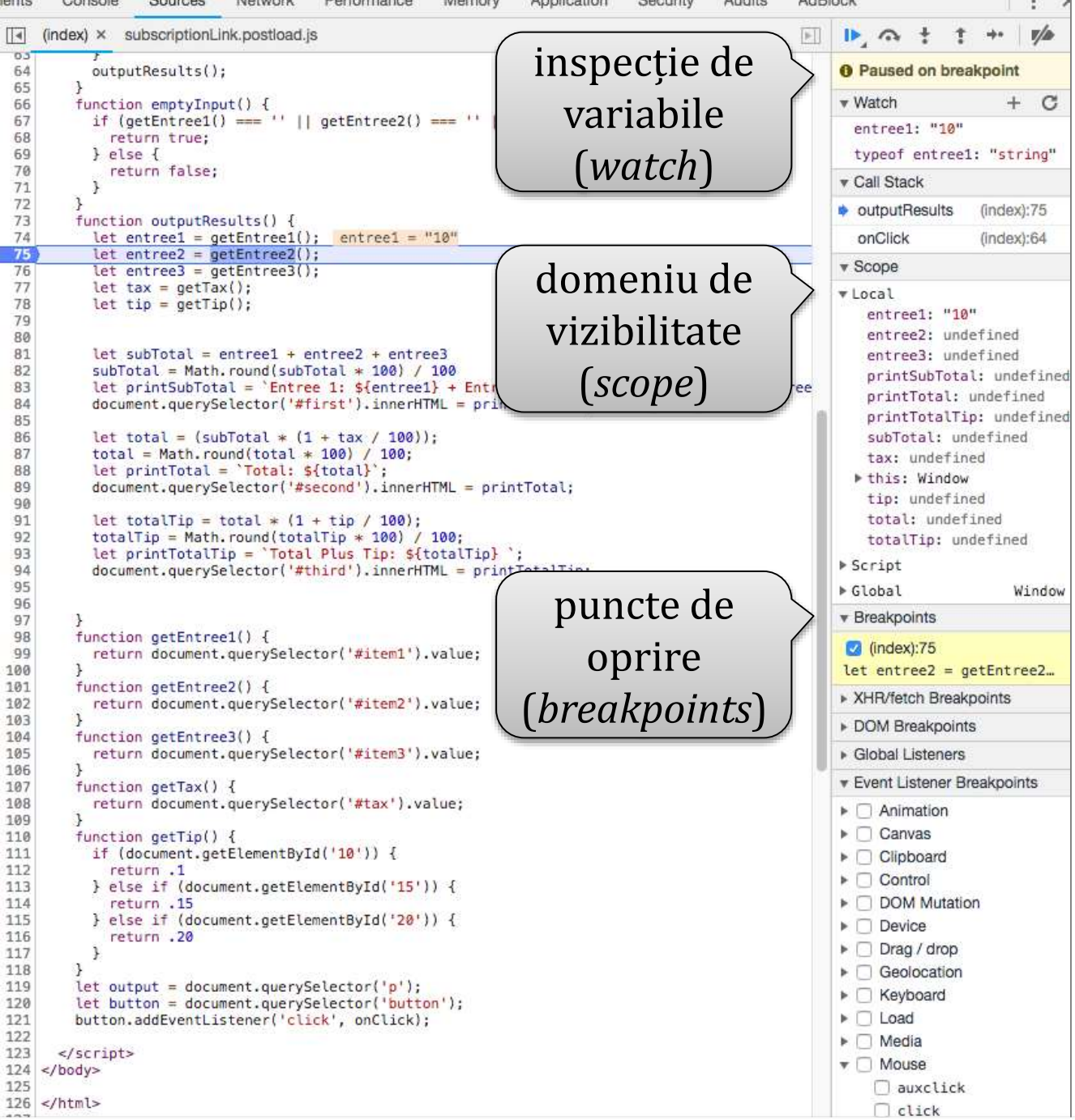
- Specification [console.spec.whatwg.org] ref
- Chrome console reference [developer.chrome.com] ref info
- MDN Web Docs - Console [developer.mozilla.org] ref info

instrumente: depanare

Instrumentele de depanare

incluse în navigatoarele Web moderne:
developers.google.com/web/tools/chrome-devtools/
developer.mozilla.org/docs/Tools
docs.microsoft.com/en-us/microsoft-edge/devtools-guide
developer.apple.com/safari/tools/

disponibile ca *proxy* – exemplu:
Telerik Fiddler – www.telerik.com/fiddler



inspecție de variabile (*watch*)

domeniu de vizibilitate (*scope*)

puncte de oprire (*breakpoints*)

stiva apelurilor (*call stack*)

puncte de oprire vizând transferuri asincrone, DOM, evenimente,…

medium.com/datadriveninvestor/stopping-using-console-log-and-start-using-your-browsers-debugger-62bc893d93ff

instrumente: testare

Verificarea corectitudinii codului JavaScript

construcția **"use strict";** indică interpretorului
că se va utiliza varianta strictă a limbajului

disponibilă începând cu ECMAScript versiunea 5

developer.mozilla.org/Web/JavaScript/Reference/Strict_mode

Dr. Sabin Buraga ▲ profs.info.uaic.ro/~busaco

instrumente: testare

Verificarea corectitudinii codului JavaScript

erorile de programare (*e.g.*, crearea accidentală
a variabilelor globale, nume identice de proprietăți etc.)
vor conduce la emiterea de excepții

instrumente: testare

Verificarea corectitudinii codului JavaScript

sunt interzise diverse facilități:
numere exprimate în baza 8,
folosirea construcțiilor with, arguments.callee
etc.

instrumente: testare

Modul de procesare strict:

modifică semantica programelor

este implicit folosit pentru modulele de cod ES6
module code is always strict mode code

nu este impus de vreun *browser* Web

```
"use strict";

varAiurea = "Ah!";
obiect = {prop: 1, prop: true};
// eroare de sintaxă
console.error (varAiurea);
```

Undeclared 'varAiurea'.
`varAiurea = "Ah!";`
Undeclared 'obiect'.
`obiect = {prop: 1, prop: true}; // eroare de sintaxă`
Duplicate 'prop'.
`obiect = {prop: 1, prop: true}; // eroare de sintaxă`
Unexpected space between 'error' and '('.
`console.error (varAiurea);`
Undeclared 'varAiurea'.
`console.error (varAiurea);`

Verificare statică a codului

JSLint – **www.jslint.com**

a se consulta și **github.com/douglascrockford/JSLint**

instrumente: testare

Verificare statică a codului

alte instrumente utile:

JSHint – **www.jshint.com**

ESHint – **eslint.org**

JS Inspect (*detects copy-pasted & structurally similar code*)

github.com/danielstjules/jsinspect

```
1   // Hello.
2   //
3   // This is JSHint, a tool that helps to detect errors and potential
4   // problems in your JavaScript code.
5   //
6   // To start, simply enter some JavaScript anywhere on this page. Your
7   // report will appear on the right side.
8   //
9   // Additionally, you can toggle specific options in the Configure
10  // menu.
11
12  $(function() { // cod rulat atunci cand browser-ul e pregatit de interactiune
13    $("div#ajutor").hide(); // initial, textul de ajutor nu e afisat
14    $("#zar1").hide();      // initial, nu sunt afisate valorile...
15    $("#zar2").hide();      // ...zarurilor (nu au fost inca 'aruncate')
16  });
17
18  function Zar (valoareZar) {       // 'clasa' privitoare la managementul unui zar
19      /* proprietati */
20    this.valoareZar = valoareZar; // valoarea curenta a zarului
21    /* metode */
22    this.obtineZar = function (){ // furnizeaza valoarea curenta a zarului
23      return this.valoareZar;
24    };
25    this.aruncaZar = function (){ // simuleaza actiunea de 'aruncare' a zarului
26      // varianta apeland serviciul Web oferit de random.org
27      var rezultat;
28      $.ajax({
29        'url': 'http://www.random.org/integers/?num=1&min=1&max=6&base=10&col=1&forma
30        'async': false,         // transfer sincron (de ce?)
31        'success': function (date) { rezultat = date; }
32      }).fail(function() {
33        // apelul a esuat, semnalam la consola acest aspect
34        console.log ('Cererea nu a putut fi satisfacuta de random.org');
35        // ...si calculam valoarea zarului in mod clasic
36        rezultat = Math.round (Math.random() * 5) + 1;
37        }
38      );
39      return this.valoareZar = parseInt (rezultat, 10);
40    };
41  }
42
43  function Joc (scorInitial){       // clasa referitoare la implementarea jocului
44    /* proprietati */
45    this.scorCurent = scorInitial;// scorul curent
46    this.zar1 = new Zar (0);      // instantele celor 2 zaruri
47    this.zar2 = new Zar (0);
```

opțiuni JSHint

CONFIGURE

Report

- ✓ Cyclomatic complexity
- ✓ Unused variables
- ✓ Undefined variables

Warn

- ✓ About == null
- ✓ About debugging code
- ✓ About unsafe for..in
- ✓ About arguments.caller and .callee
- ✓ About assignments if/for/...
- ✓ About functions inside loops
- ✓ About eval
- ✓ About unsafe line breaks
- ✓ About potential typos in logical operators
- ✓ When code is not in strict mode
- ✓ When new is used for side-effects

Assume

- ✓ Browser
- ✓ NodeJS
- ✓ jQuery
- ✓ Development (console, etc.)
- ✓ New JavaScript features (ES6)
- ✓ Mozilla JavaScript extensions
- ✓ Older environments (ES3)

instrumente: testare

Suport pentru testare, inclusiv *unit testing*

exemplificări:
AVA, Cypress, dom-testing-library, Flow, Jasmine, Jest, Mocha, PhantomJS, Qunit, Sinon.js, Tape, TestCafe etc.
github.com/sorrycc/awesome-javascript#testing-frameworks

de studiat și *Let's Code: Test-Driven JavaScript*
www.letscodejavascript.com

instrumente: testare

Testare *cross-browser* a aplicațiilor Web

exemple:

Browsera – **www.browsera.com**

Browserling – **www.browserling.com**

Browser Sandbox – **turbo.net/browsers**

Browser Shots – **browsershots.org**

instrumente: documentarea codului

Standarde de redactare a codului JavaScript

exemplificări:

Code Conventions for JavaScript

www.crockford.com/javascript/code.html

Principles of Writing Consistent, Idiomatic JavaScript

github.com/rwaldron/idiomatic.js

instrumente: documentarea codului

Standarde de redactare a codului JavaScript

ghiduri specifice – câteva exemple:

Airbnb JavaScript Style Guide

github.com/airbnb/javascript

Google JavaScript Style Guide

google.github.io/styleguide/jsguide.html

React Native Code Style

facebook.github.io/react-native/docs/style.html

instrumente: documentarea codului

Standarde de redactare a codului JavaScript

instrumente utile (formatatoare a codului-sursă):
JS Beautifier – **jsbeautifier.org**
Prettier – **prettier.io**
esformatter – **github.com/millermedeiros/esformatter**

```
define ('my/toy', function () {
   /**
    * A module representing a toy.
    * @exports my/toy
    * @version 1.0
    */
    var toy = {
        /** A property of the module. */
        color: "black",

        /** @constructor */
        Toy: function(size) {
            /** A property of the class. */
            this.size = size;
        }
    };
    return toy;
});
```

marcaje (adnotări) speciale în cadrul comentariilor:

@abstract
@access
@alias
@async
@extends
@class
@interface
@callback
@event
@emits
@listens
@function
@property
@module
@requires
...

@author
@copyright
@license
@summary
@description
@version
@example
@since
@see
@todo
@deprecated

documentarea codului: JSDoc Toolkit – usejsdoc.org

instrumente: compresie de cod

Instrumente privind compresia/minimizarea

exemple:

Javascript Compressor – **javascriptcompressor.com**

Minify – **www.minifier.org**

UglifyJS (Mihai Bazon, absolvent FII) – **lisperator.net/uglifyjs/**

instrumente: optimizare javascript

Transformarea codului JS într-unul optimizat

exemple:
Closure Compiler – **developers.google.com/closure/**
Optimize – **github.com/nolanlawson/optimize-js**

instrumente: pachete

Managementul de pachete JavaScript pentru dezvoltarea de aplicații Web la nivel de client

căutare, instalare, compilare, verificare a dependențelor

exemple:
Bower, jspm, npm, Yarn etc.

github.com/sorrycc/awesome-javascript#package-managers

instrumente: pachete

Managementul de pachete JavaScript pentru dezvoltarea de aplicații Web la nivel de client

crearea de conglomerate de cod/resurse (*bundles*)

exemplificări:
browserify, Rollup (specific ES6), webpack

github.com/sorrycc/awesome-javascript#bundlers

instrumente: fluxuri de activități

Suport pentru fluxuri de activități (*workflow*-uri), eventual realizate automat

exemple:
Brunch, Grunt, Yeoman, Phantom.js, Plop, Selenium

instrumente: machete de redare

Sisteme de management al machetelor de redare a conținutului (*templating engines*)

exemplificări:
Dust.js, EJS, Handlebars, Mustache.js,...

github.com/sorrycc/awesome-javascript#templating-engines

Dr. Sabin Buraga ▸ profs.info.uaic.ro/~busaco

instrumente: programare

Biblioteci pentru programarea funcțională

Lodash – **lodash.com**
Ramda – **ramdajs.com**
Underscore – **underscorejs.org**

resurse de interes la **github.com/stoeffel/awesome-fp-js**

Dr. Sabin Buraga ▲ profs.info.uaic.ro/~busaco

instrumente: programare

Biblioteci pentru programarea reactivă

vizează procesarea fluxurilor de date asincrone pe baza paradigmei funcționale

adoptă șablonul de proiectare *Observer*

www.learnrxjs.io

instrumente: programare

Biblioteci pentru programarea reactivă

exemple:

RxJS – **github.com/ReactiveX/rxjs**

Cycle – **cycle.js.org**

MobX – **mobx.js.org**

Most.js – **github.com/cujojs/most**

instrumente

Suport pentru creșterea performanței

asm.js (Mozilla, 2012—2014)
subset JavaScript ce poate fi utilizat ca limbaj de nivel scăzut, eficient – în spiritul limbajului de asamblare

rulează nativ în Chrome, Edge și Firefox

asmjs.org

Dr. Sabin Buraga ▲ profs.info.uaic.ro/~busaco

instrumente

Suport pentru creșterea performanței

WebAssembly – wasm (în lucru, 20 noiembrie 2018)
limbaj de programare proiectat pentru execuție eficientă la nivel de (*browser*) Web

safe, portable, low-level code format

webassembly.org • developer.mozilla.org/WebAssembly

Dr. Sabin Buraga ▲ profs.info.uaic.ro/~busaco

instrumente

Suport pentru creșterea performanței

WebAssembly – wasm

include un set restrâns de tipuri de date și operații

permite optimizări la momentul compilării

axat pe realizarea de calcule numerice complexe

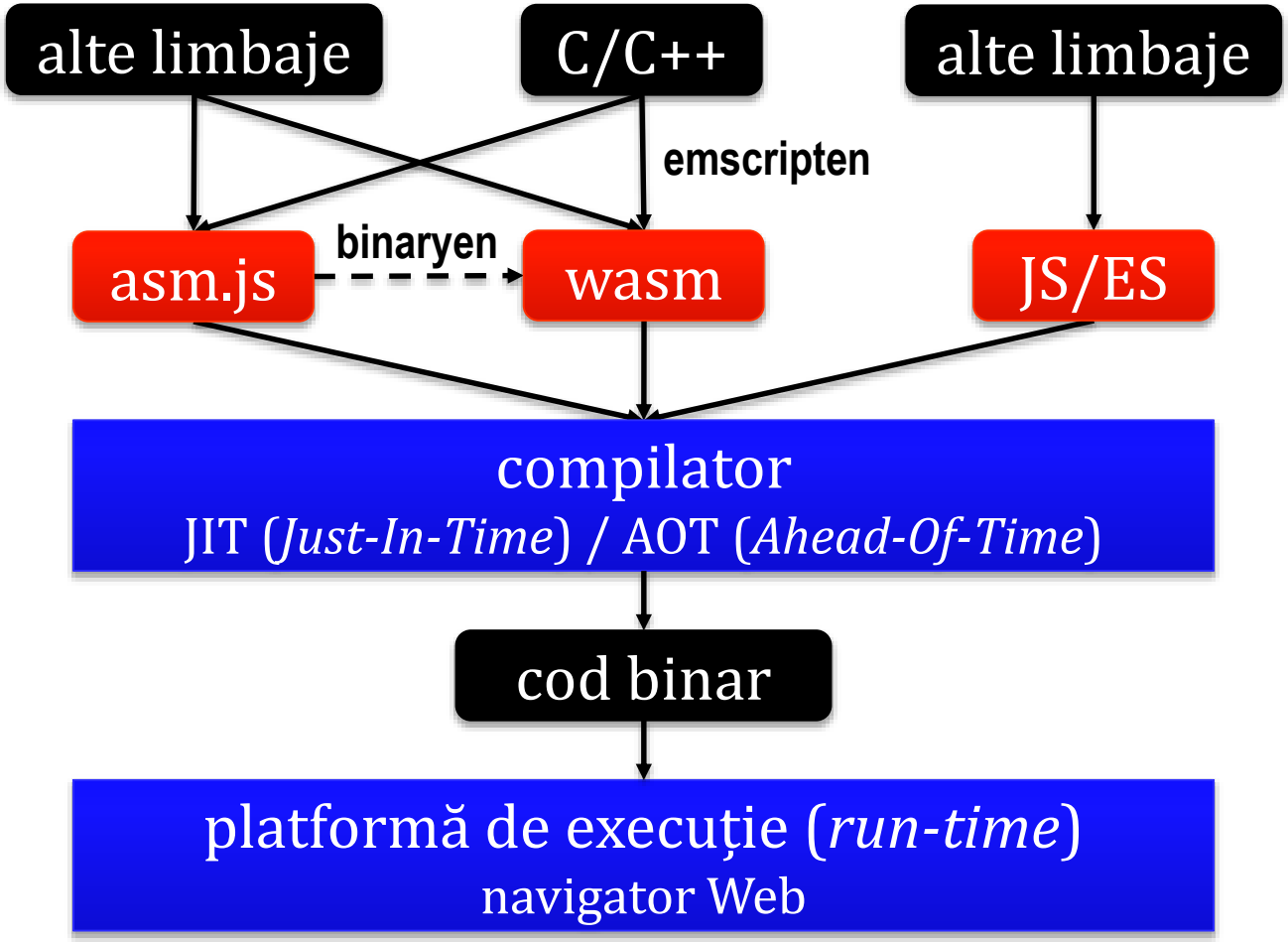
Inside the browser

- Better execution for languages and toolkits that are currently cross-compiled to the Web (C/C++, GWT, …).
- Image / video editing.
- Games.
- Peer-to-peer applications (games, collaborative editing, decentralized and centralized).
- Music applications (streaming, caching).
- Image recognition.
- Live video augmentation (e.g. putting hats on people's heads).
- VR and augmented reality (very low latency).
- CAD applications.
- Scientific visualization and simulation.
- Platform simulation / emulation (ARC, DOSBox, QEMU, MAME, …).
- Language interpreters and virtual machines.
- Developer tooling (editors, compilers, debuggers, …).
- Remote desktop. VPN. Encryption
- Local web server.
- Fat client for enterprise applications (e.g. databases).

Outside the browser

- Game distribution service (portable and secure).
- Server-side compute of untrusted code.
- Server-side application.
- Hybrid native apps on mobile devices.

diverse studii de caz:
webassembly.org/docs/use-cases/


alte limbaje
C/C++
alte limbaje
emscripten
binaryen
asm.js
wasm
JS/ES
compilator
JIT (*Just-In-Time*) / AOT (*Ahead-Of-Time*)
cod binar
platformă de execuție (*run-time*)
navigator Web

instrument online:

WebAssembly Playground – **ast.run**

instrumente

Utilizarea unui modul wasm via API-ul disponibil
webassembly.org/docs/js/

```
fetch ("factorial.wasm") // încărcare asincronă a codului WebAssembly
.then (response => { response.ArrayBuffer (); })
.then (buffer => {
    var tablouDate = new Uint32Array (buffer);
    var fact = WebAssembly.instantiate (tablouDate);
    var rezultat = fact.factorial (5);
    ...
});
```

WebAssembly - OTHER

Usage % of all users

Global 80.2%

WebAssembly or "wasm" is a new portable, size- and load-time-efficient format suitable for compilation to the web.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

Notes | Known issues (0) | Resources (7) | Feedback

- Specification [github.com] ref
- WebAssembly on MDN [developer.mozilla.org] info
- Official site [webassembly.github.io] ref demo info tutorial

instrumente

Alte limbaje – compilate în JS – pentru dezvoltarea de aplicații Web la nivel de client:

CoffeeScript (Jeremy Ashkenas, 2009)
coffeescript.org

dialecte CoffeeScript:
CoffeeScriptRedux – **github.com/michaelficarra/CoffeeScriptRedux**
LiveScript – **livescript.net**

instrumente

Alte limbaje – compilate în JS – pentru dezvoltarea de aplicații Web la nivel de client:

TypeScript (Microsoft, 2012)
www.typescriptlang.org

instrumente + resurse:
github.com/dzharii/awesome-typescript

instrumente

Portarea altor aplicații în JavaScript

Emscripten – compilator LLVM generând cod JS/wasm
(*e.g.*, programe C sau C++ care se compilează
în asm.js sau WebAssembly, cod OpenGL în WebGL)

kripken.github.io/emscripten-site/

instrumente

Portarea altor aplicații în JavaScript

JSIL – compilator care transformă codul CIL (*Common Intermediate Language*) al aplicațiilor .NET în programe JavaScript rulând în *browser*

jsil.org

instrumente

Portarea altor aplicații în JavaScript

ClosureScript – Closure ▸ JS: **github.com/clojure/clojurescript**

PyPy.js – Python ▸ asm.js: **pypyjs.org**

Scala.js – compilează programele Scala în cod JS
www.scala-js.org • **www.lihaoyi.com/hands-on-scala-js/**

PureScript – **www.purescript.org**

N-am putea recurge la biblioteci/*framework*-uri JS specifice?

colecții de (micro-)biblioteci JS:

www.javascripting.com

jster.net

microjs.com

Biblioteci JavaScript specializate:

procesarea formularelor Web
facilitarea transferurilor asincrone de date
tehnici criptografice
realizarea de efecte vizuale
generarea de conținut grafic 2D/3D
vizualizarea datelor
dezvoltare de jocuri (*e.g.*, *game engines*)
...

Biblioteci/*framework*-uri populare
conform **2018.stateofjs.com**

front-end: React, Vue.js, Angular, Preact, Ember, Polymer
altele: Svelte, Aurelia, Hyperapp, Backbone, Mithril,...

data layer: Redux, GraphQL, Apollo, MobX, Relay (Modern)
altele: VueX, ember-data, NgRx, RxJS etc.

testing: Jest, Mocha, Jasmine, Enzyme, Karma, Storybook
altele: Cypress, QUnit, Tape, Chai, TestCafe, Protractor,...

mobile & desktop: Electron, React Native, Cordova, Ionic
altele: Flutter, Weex, Quasar, Expo etc.

Biblioteci/*framework*-uri specializate – exemple:

prelucrare/generare de date în diverse formate:
Js Barcode, jsPDF, JS Xlsx, MathJax, Numeric, Papa Parse, pdfmake, Psd.js, Qrcodejs, SheetJS, STDlib, Superstruct

baze de date: Alasql, Juggling DB, Knex, Local Forage, Loki JS, PouchDB, Typeorm, Watermelon Db,…

internaționalizare: Globalize, i18next, Lingui, Polyglot

suport pentru OAuth: hello.js, JSO, Salte Auth

proiectarea interfeței cu utilizatorul: Anypixel, Ant Design, Ace, Deck.js, Materialized, Reveal, Styled Components, Semantic UI, Slate etc.

dezvoltare de jocuri: Cannon, Easystar, LiquidFun, Melon, P2.js, Phaser, Pixi.js, Whitestorm,…

Biblioteci/*framework*-uri specializate – exemple:

rețea: Axios, Faye, Request, PeerJS, SockJS, Superagent,…

arhivare: JSZip, LZ String, ZIP.js

prelucrare audio: Howler, Midi.js, SoundManager2, Tone.js

manipulare de imagini: Anypixel, Bonsai, Caman, Cytoscape, Dagre, Drawingboard, Fabric.js, MetricsGraphics.js, Nude.js, Paper.js, Plotfly, Processing.js

învățare automată: Clmtrackr, Ml5, TensorFlow.js, TFjs,…

rețele neuronale: Brain.js, Convnet.js, Keras.js, Synaptic

hardware (*e.g.*, roboți, IoT, imprimante 3D): Cylon, IoTjs, Johnny-five, Maker.js, Ruff.io etc.

Nu putem adopta diverse șabloane de proiectare pentru JavaScript?

Șabloane de proiectare tradiționale

creaționale

Builder, *Prototype*, *Singleton*

structurale

Adapter, *Bridge*, *Decorator*, *Façade*, *Flyweight*, *Proxy*

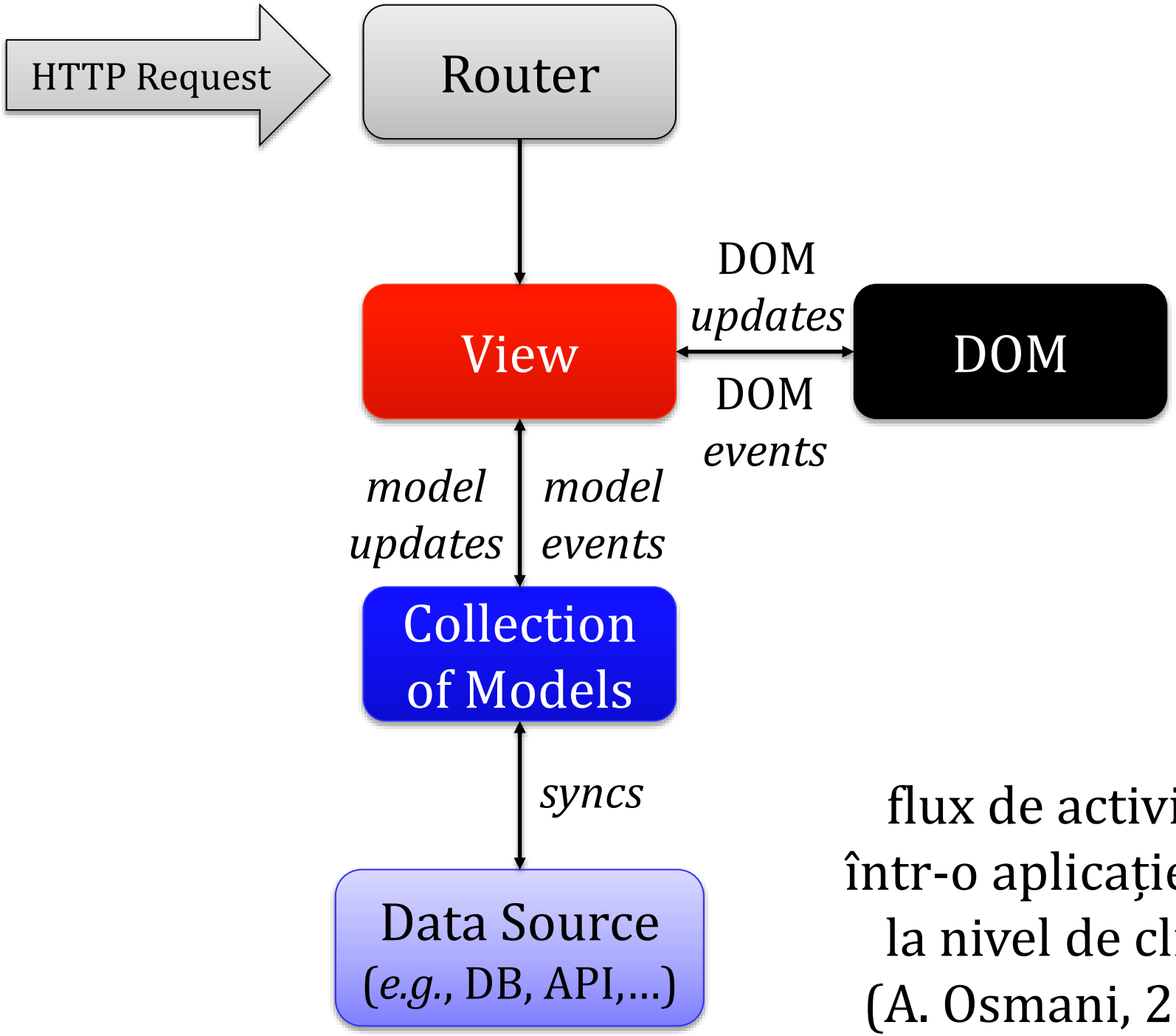
comportamentale

Command, *Iterator*, *Mediator*, *Observer*, *State*, *Visitor*

Șabloane de proiectare MV*

MVC (*Model-View-Controller*)
MVP (*Model-View-Presenter*)
MVVM (*Model View ViewModel*)

pentru detalii, de explorat:
Addy Osmani, *Learning JavaScript Design Patterns* (2017)
www.addyosmani.com/resources/essentialjsdesignpatterns/book/



flux de activități
într-o aplicație MV*
la nivel de client
(A. Osmani, 2013)

Șabloane de proiectare MV*

exemplificări de implementări JS:

Dusan Gledovic, *Basic JS MVC Implementation* (2015)
gist.github.com/gschema/4157554

Camilo Reyes, *The MVC Design Pattern in Vanilla JavaScript* (2017)
www.sitepoint.com/mvc-design-pattern-javascript/

Șabloane de proiectare MV*

utilizare pragmatică via biblioteci/*framework*-uri:

Angular – **angular.io**
Aurelia – **aurelia.io**
Backbone – **backbonejs.org**
Backbone Marionette – **marionettejs.com**
Ember.js – **emberjs.com**

Anumite biblioteci/*framework*-uri pot fi specializate, tratând doar un anumit aspect al MVC/MVVM

privind dirijarea:
router – **github.com/kevindurb/router**

pentru *View* – exemple:
Knockout – **knockoutjs.com**
React – **reactjs.org**
Vue – **vuejs.org**

vizând partea de *Model* – exemplificare:
Breeze.js – **www.getbreezenow.com/breezejs**

Șabloane de proiectare specifice

aspect de interes:
modularizarea codului

în cazul JS clasic, specificarea modulelor se poate realiza via limbajul AMD (*Asynchronous Module Definition*)

github.com/amdjs/amdjs-api

Șabloane de proiectare specifice

aspect de interes:
modularizarea codului

în cazul JS clasic, specificarea modulelor se poate realiza via limbajul AMD (*Asynchronous Module Definition*)

încărcare de cod folosind biblioteci specifice:
curl.js, PINF, RequireJS,...

Șabloane de proiectare specifice

aspect de interes:
modularizarea codului

alternativ, se poate recurge la CommonJS
un format de declarare a modulelor și pachetelor reutilizabile la nivel de server

wiki.commonjs.org/wiki/CommonJS

abordare fără module (monolitică)	**abordare modulară (pachete + module)**
fiecare fragment de cod este implicit global	pachetele expun interfețe publice ușor de înțeles
numele (clase, funcții, constante,…) sunt globale	numele sunt locale pachetului ce le definește
acces direct la implementarea efectivă	detaliile de implementare sunt ascunse
dependența de ordinea încărcării fișierelor	ordinea încărcării fișierelor nu are importanță
dependențele de cod sunt implicite	dependențele de cod sunt declarate explicit
relații nespecificate între fișiere & module	fiecare fișier expune un modul unic
dependențele depind de contextul rulării (uzual, *browser*)	se permite rularea din linia de comandă (*headless browser*)

Șabloane de proiectare specifice

aspect de interes:
modularizarea codului

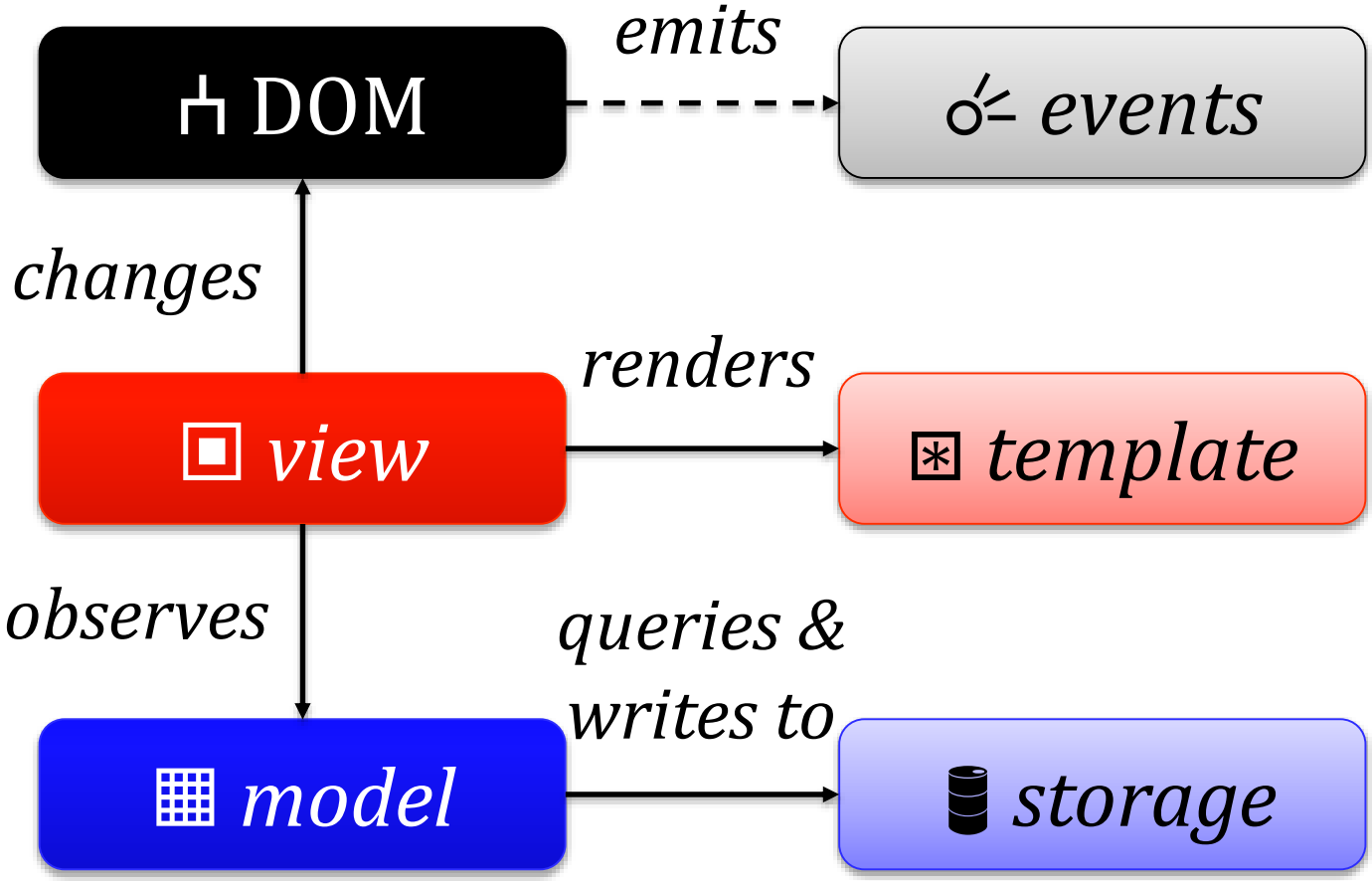
pentru alte resurse de interes, a se consulta
Nicolás Bevacqua, *Module Design*, 2018
ponyfoo.com/articles/module-design
Serg Hospodarets, *Native JavaScript modules*, 2018
slides.com/malyw/native-js-modules

ES6 modules for Web usage:
github.com/webmodules

Șabloane de proiectare specifice

aspect de interes:
SPA (*Single Page Applications*)

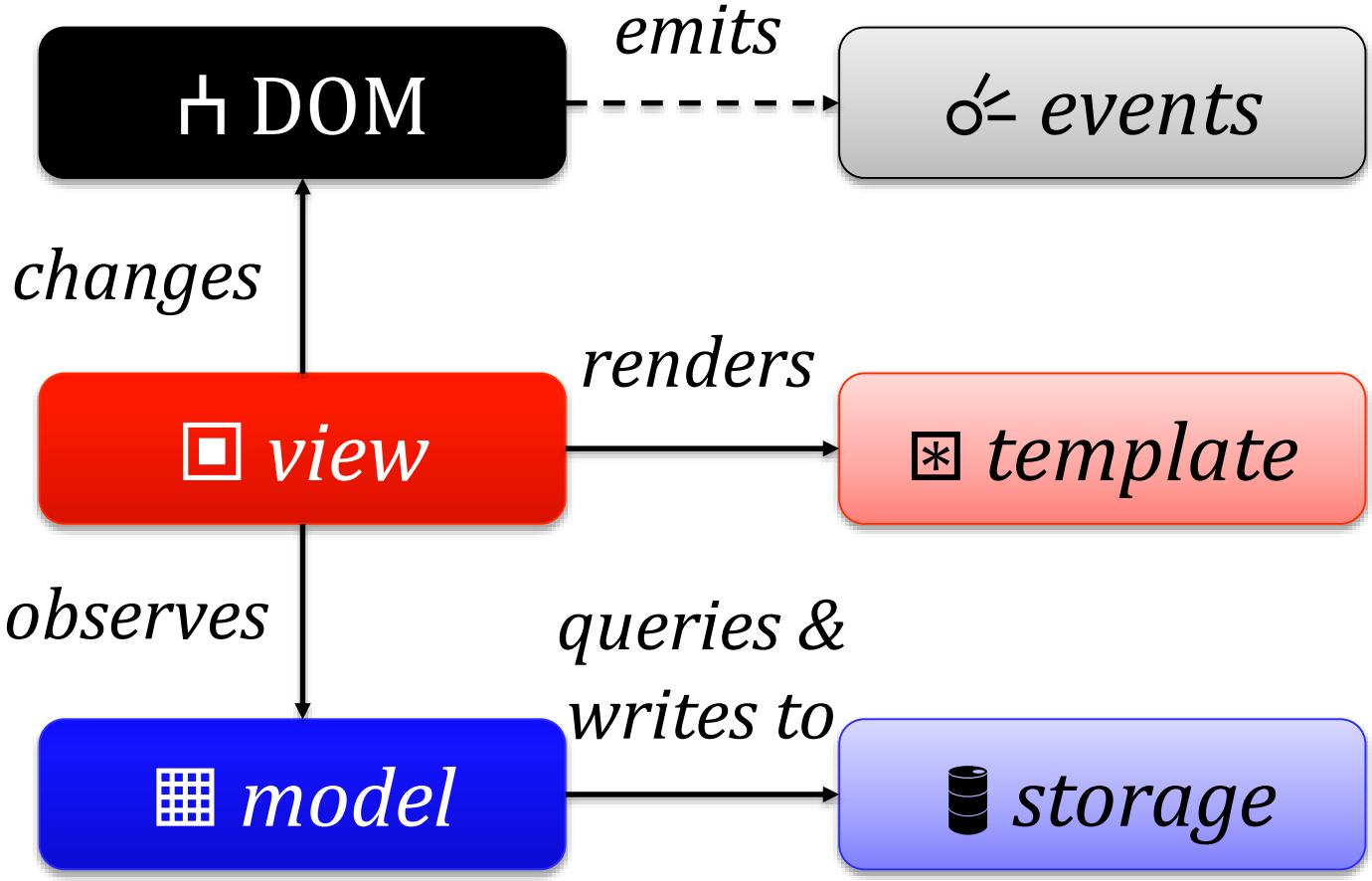
rescrierea dinamică a conținutului paginii Web
– pe baza transferului asincron al datelor –
în urma interacțiunii cu utilizatorul



structura de bază a unei SPA

de studiat Mikito Takada, *Single Page Apps in Depth* (2017)

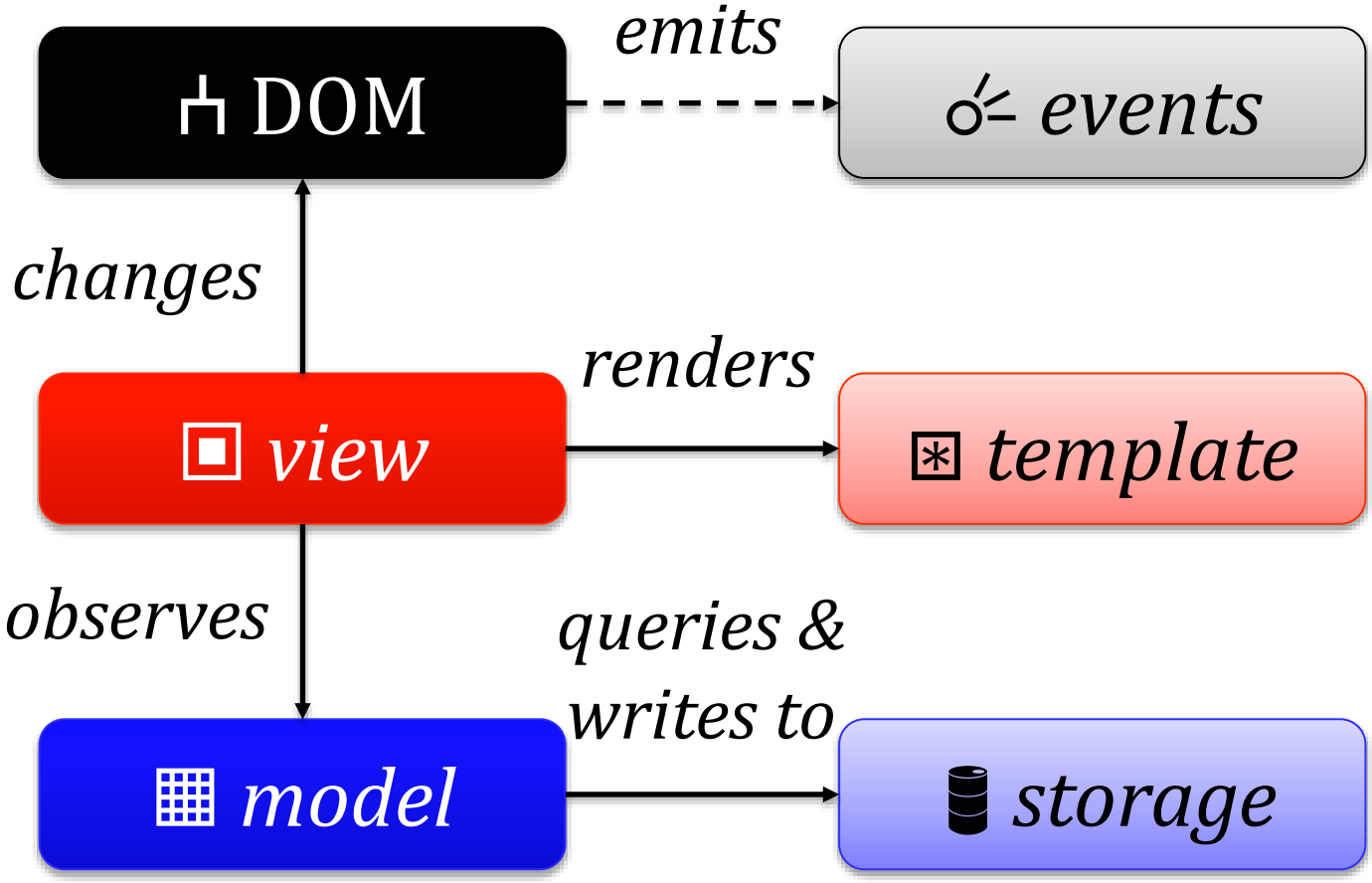
singlepageappbook.com



write-only DOM

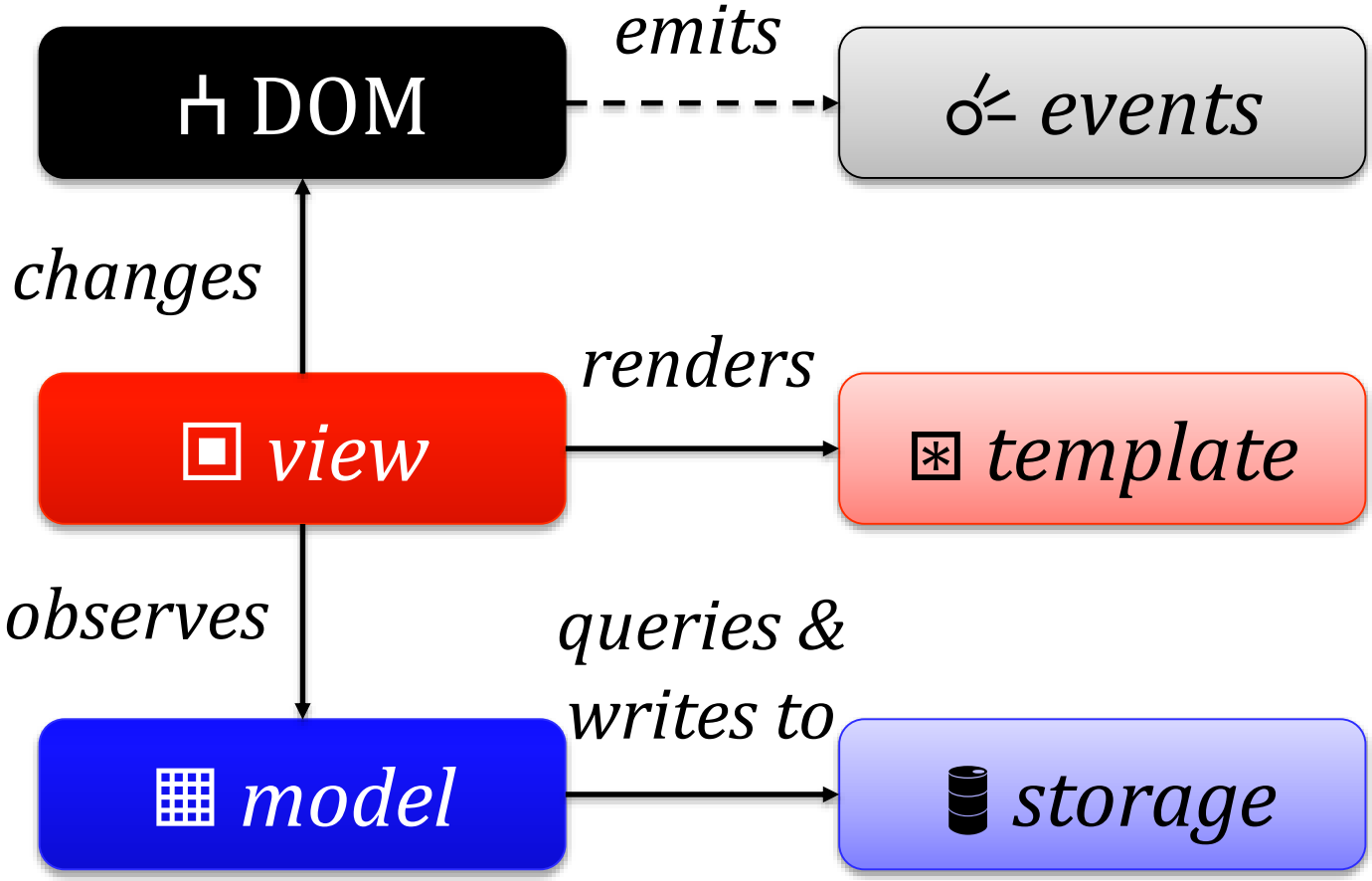
nu sunt preluate date din DOM

managementul stării se face independent de DOM



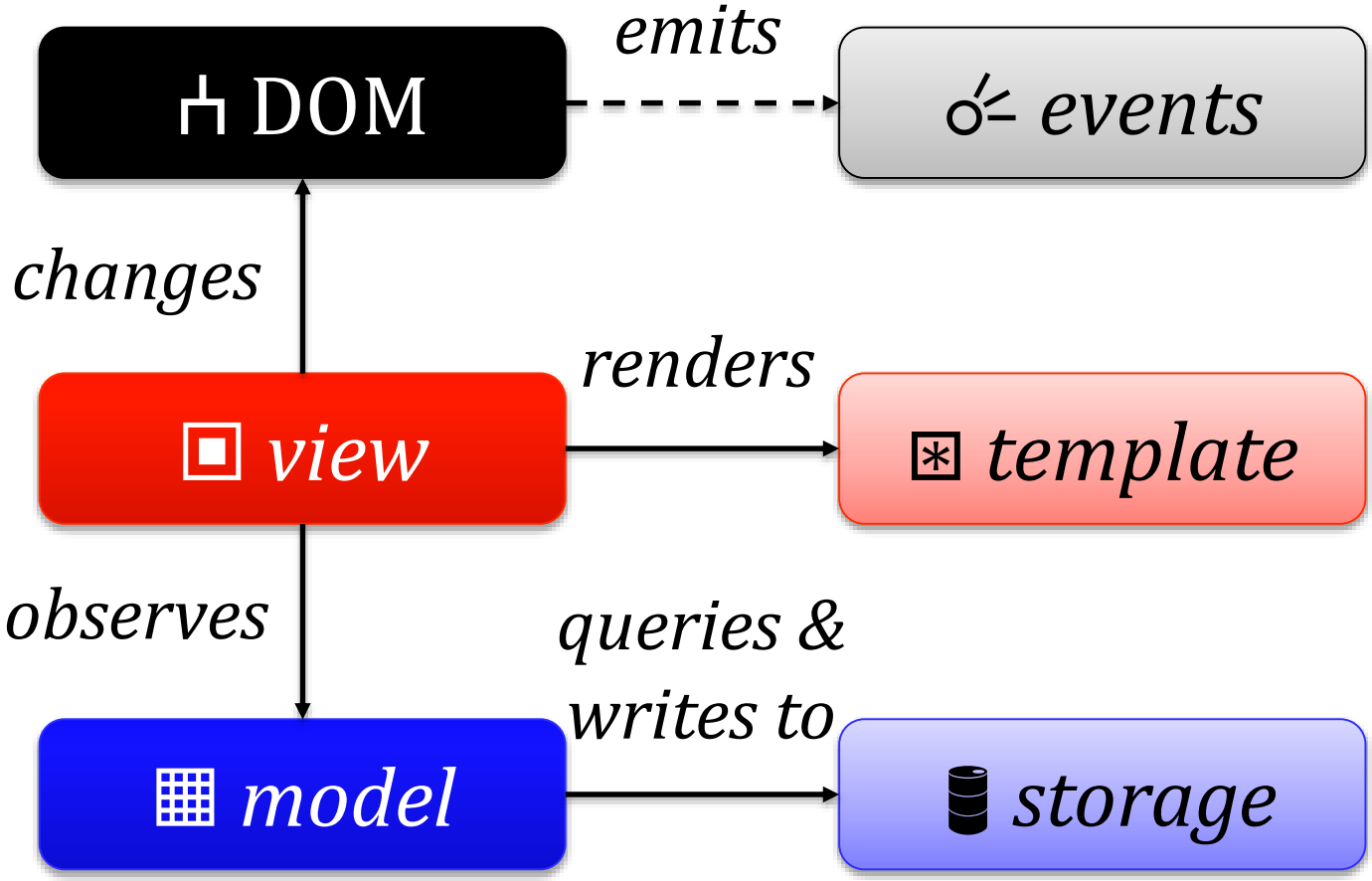
models are the single source of truth

modelele reprezintă toate datele/stările aplicației Web și sunt păstrate în memorie



views observe model changes

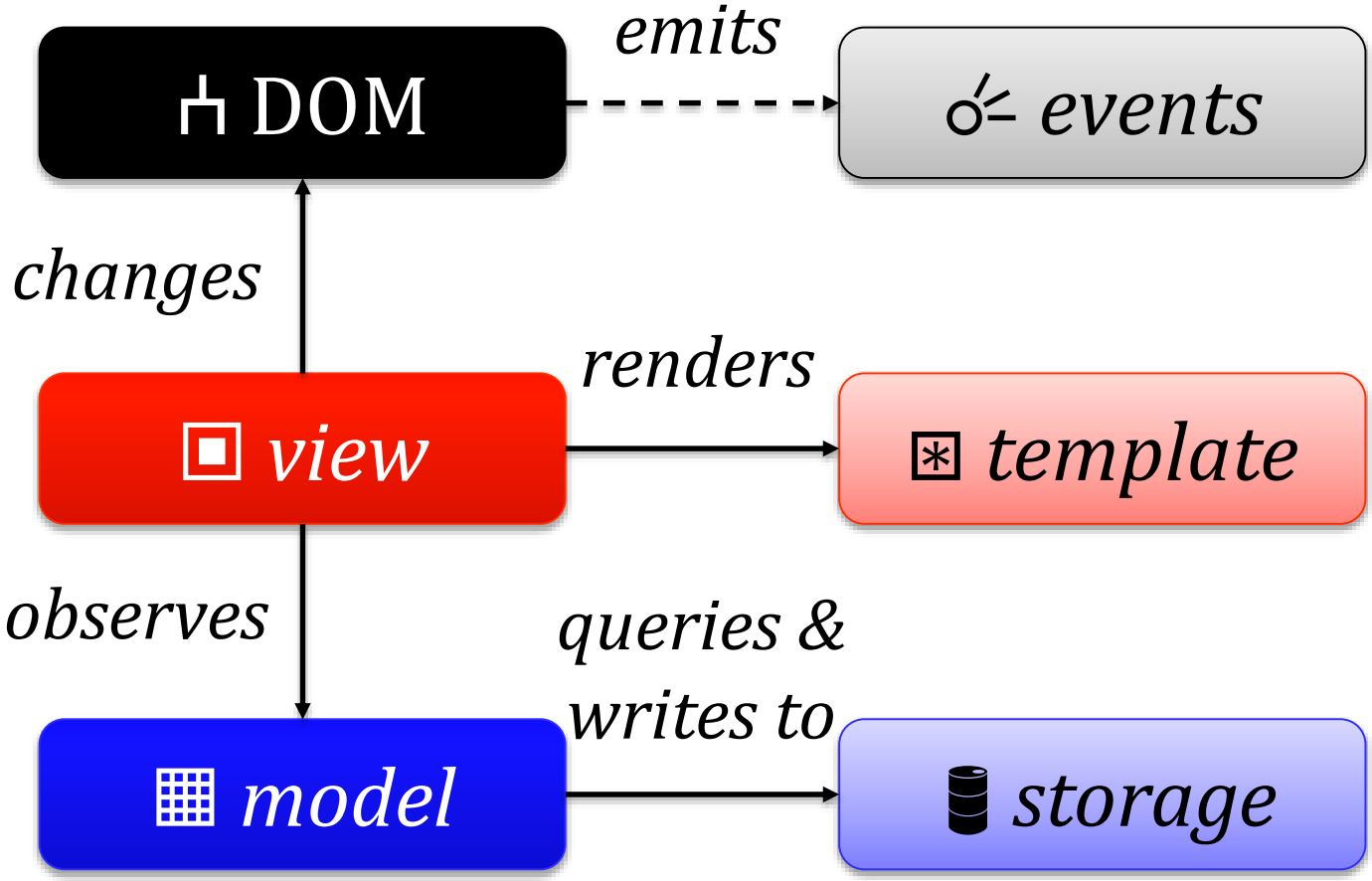
view-urile reflectă conținutul modelelor și primesc notificări de actualizare din partea modelelor



decoupled modules that expose small external surfaces

arhitectura aplicației este compusă din
sub-sisteme (module) independente, specializate

- pachete expunând o interfață simplă publică



minimizing DOM dependent-code

minimizarea și izolarea codului JS
vizând manipularea arborelui DOM

SPA (*Single Page Applications*)

managementul stării aplicației la momentul rulării
URL↔stare

starea curentă a *view*-ului e dependentă de URL
uzual, de ***#fragment-identifier***

tools.ietf.org/html/rfc3986#section-3.5

SPA (*Single Page Applications*)

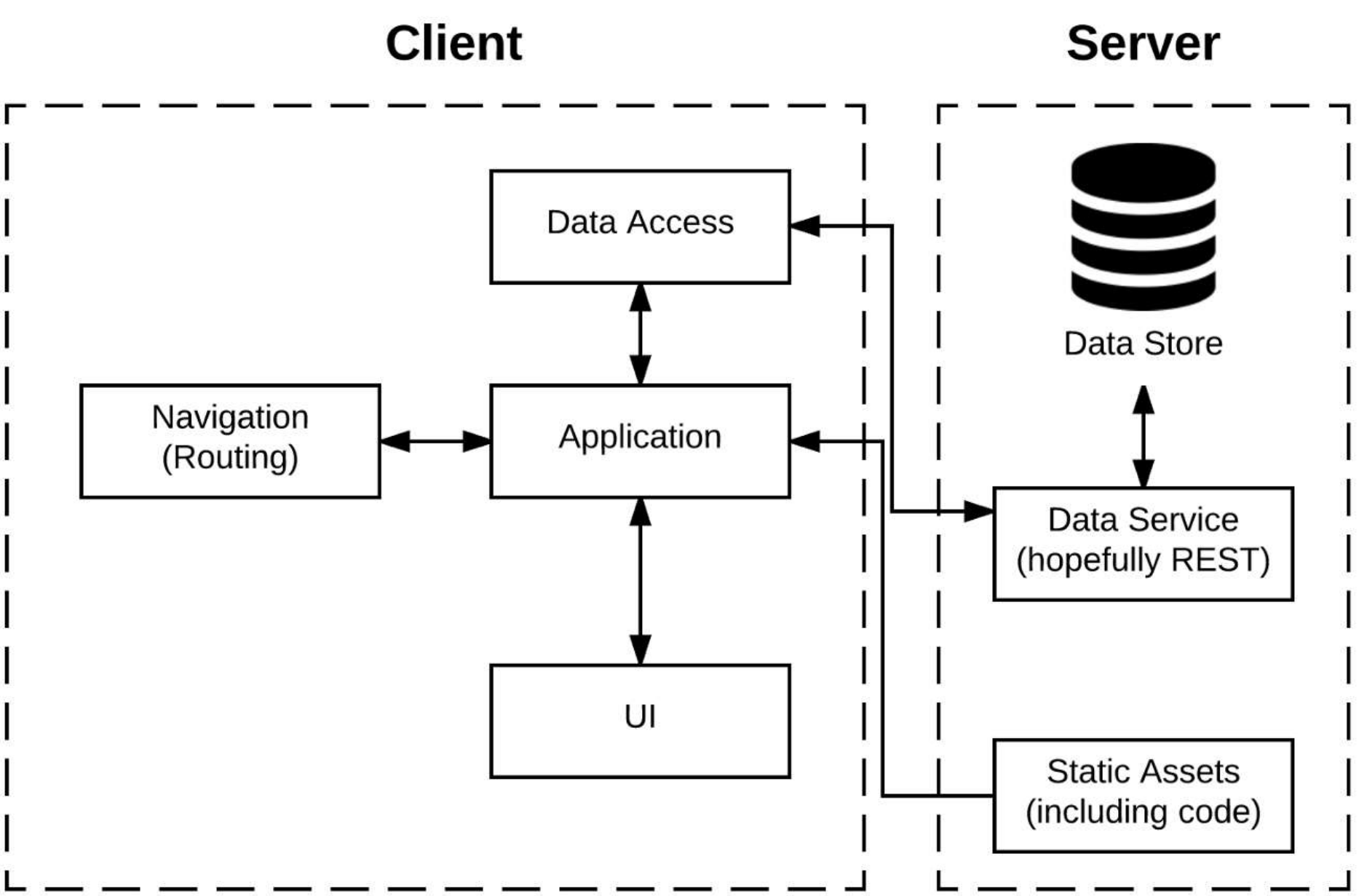
definirea unor componente ce ulterior pot fi inițializate

▼

suport pentru reutilizare

aceste componente pot fi încărcate asincron
(eventual, la cerere) via module

Dr. Sabin Buraga ▲ profs.info.uaic.ro/~busaco



componente majore ale unei SPA (Patrick Ackerman, 2017)

Șabloane de proiectare specifice

aspect de interes:
SPA (*Single Page Applications*)

framework-uri/biblioteci:
Angular, Aurelia, Ember.js
Meteor.js – **meteor.com**
Mithril – **mithril.js.org**
...

Șabloane de proiectare specifice

aspect de interes:
comunicare prin paradigma *publish/subscribe*

WebSub (*W3C Recommendation*, ianuarie 2018):
www.w3.org/TR/websub/

biblioteci:
AmplifyJS – **amplifyjs.com**
PubSubJS – **github.com/mroderick/PubSubJS**

JavaScript în contextul Web-ului mobil

instrumente multi-platformă pentru crearea așa-numitor aplicații hibride (Web + native)

Apache Cordova (ex-PhoneGap) – **cordova.apache.org**
Ionic – **ionicframework.com**
React Native – **facebook.github.io/react-native/**
Sencha Ext JS – **www.sencha.com/products/extjs/**
Tabris.js – **tabrisjs.com**

Tabris.js Playground

Snippet:

navigationview-searchaction.js

Save

```
1  import {Button, Composite, NavigationView, Page, SearchAction, TextView, ui} from
2
3  // Create an action on NavigationView to perform a search with dynamic proposals
4
5  const PROPOSALS = ['baseball', 'batman', 'battleship', 'bangkok', 'banana'];
6
7  const navigationView = new NavigationVie
8    left: 0, top: 0, right: 0, bottom: 0
9  }).appendTo(ui.contentView);
10
11 const page = new Page({
12   title: 'Search action'
13 }).appendTo(navigationView);
14
15 const searchBox = new Composite({
16   centerX: 0, centerY: 0
17 }).appendTo(page);
18
19 const textView = new TextView().appendTo
20
21 const action = new SearchAction({
22   title: 'Search',
23   image: {
24     src: device.platform === 'iOS' ? 're                    .png' :
25                  'resources/search-white-
26     scale: 3
```

Go to Definition Ctrl+F12
Peek Definition Alt+F12
Find All References Shift+F12
Go to Symbol... Ctrl+Shift+O
Change All Occurrences Ctrl+F2
Format Document Shift+Alt+F
Format Selection Ctrl+K Ctrl+F
Cut
Copy
Command Palette F1

Ctrl+Space: Auto Complete | F1: Command Palette | Right Click: Context Menu |

Compiled Code

The playground is bound to the latest release of Tabris.js. TypeScript and JSX are supported.

How to Run:

1 - Get the Developer App

Download on the App Store

GET IT ON Google play

2 - Scan QR Code

Press the barcode button in the app's URL tab and scan this:

JavaScript în contextul Web-ului mobil

framework-uri și biblioteci JavaScript specializate

componente de interfață (eventual, pentru prototipizare):

jQuery UI – **jqueryui.com**

jQuery Mobile – **jquerymobile.com**

Mobile Angular UI – **mobileangularui.com**

Onsen UI – **onsen.io**

JavaScript în contextul Web-ului mobil

framework-uri și biblioteci JavaScript specializate

e.g., interacțiune prin gesturi:
Hammer.js – **hammerjs.github.io**
iScroll – **github.com/cubiq/iscroll**
Slideout.js – **slideout.js.org**

altele la **www.javascripting.com/mobile-and-touch/**

JavaScript în contextul Web-ului mobil

recurgerea la API-uri JavaScript oferite de platformă

Amazon Fire

developer.amazon.com/webapps

Microsoft Windows Universal Applications

docs.microsoft.com/en-us/windows/uwp/

Tizen

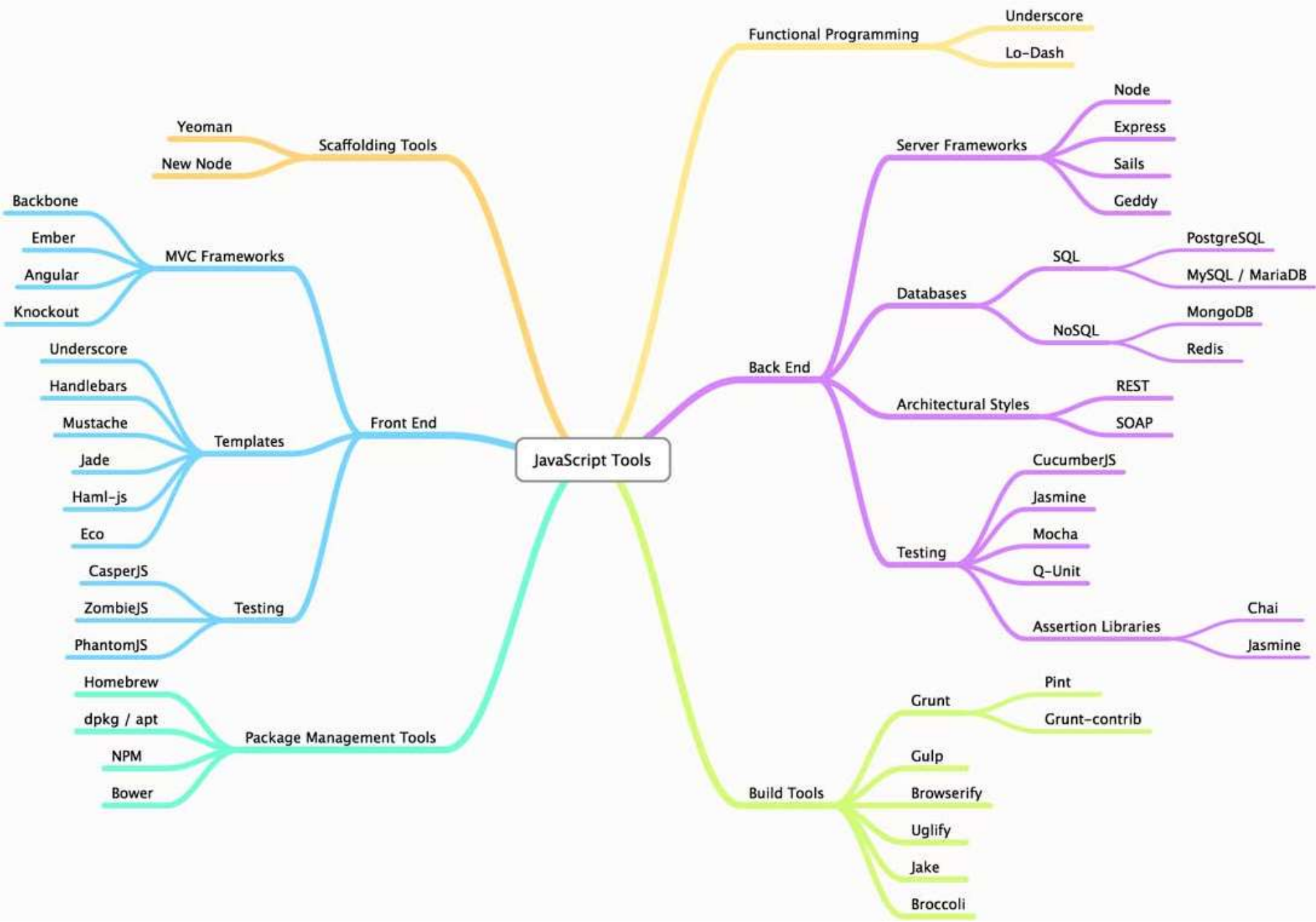
developer.tizen.org/category/tags/javascript

Extinderi

JS++ (garantează tipul variabilelor – *type guarantees*)
www.onux.com/jspp/

Objective-J (modelat după Objective-C,
integrat în *framework*-ul Cappuccino)
www.cappuccino-project.org

PLV8 (extensie JavaScript pentru PostgreSQL)
plv8.github.io



www.codefellows.org/blog/a-list-of-foundational-javascript-tools

episodul viitor:

suita de tehnologii HTML5