

ROLLING SCOPES FRONTEND COURSE

Brest, 2016
Aliaksandr Orgish

Rolling Scopes Community



<http://rollingscopes.com>

<http://brest.rollingscopes.com>

Slack chat invites: <http://brest-rolling-scopes.azurewebsites.net>

Plan

- Introduction. How browsers work
- HTTP
- Introduction to HTML5 / CSS3
- Introduction to Javascript 2015
- Introduction to Functional Programming
- Unit testing and Continuous Integration
- Nodejs, Express
- Developers Tools
- Introduction to REST
- Web apps architecture
- Introduction to JS Frameworks (Angular / React)

How to draw an owl

1.



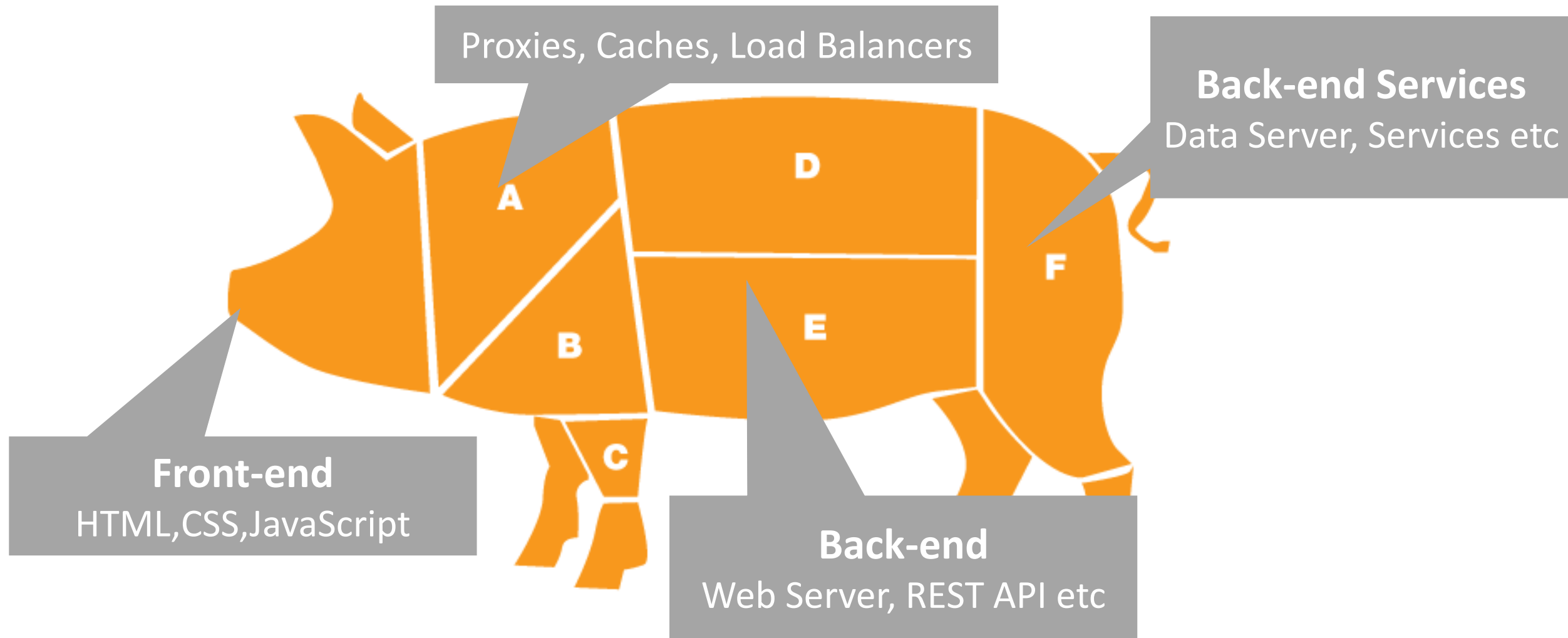
2.



1. Draw some circles

2. Draw the rest of the fucking owl

Web app infrastructure



Frontend Specialization : Web Designer

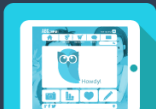
THE 10 COMMANDMENTS OF USER INTERFACE DESIGN

1

CREATE
A STORY



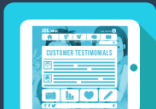
USE REAL IMAGES



MASCOT



CUSTOM MESSAGES



BE AUTHENTIC

2

STREAMLINE
NAVIGATION



USE ICONS



MOBILE-FRIENDLY



DIRECT MANIPULATION



CLEAR CALL TO ACTION

3

MAKE IT
RESPONSIVE



USE COLUMNS



PARALLAX SCROLLING



FLUID GRID



DROP-DOWN MENUS

4

ENSURE
ACCESSIBILITY



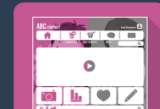
LABEL LINKS



ALT-TEXT FOR IMAGES



AVOID IMAGE AS LINK



MONOCHROME SETTINGS

5

FORM FOLLOWS
FUNCTION



BUTTON



CHECK BOXES



DROP-DOWN LIST



SLIDER

6

USE PLEASANT
COLOR THEMES



COLOR PALETTE



PICK VIBRANT COLORS



CONTRAST IS CLARITY



INDUSTRY RELEVANT

7

DEFINE FONT
FAMILIES



NO SERIFS



PERFECT DUOS



SET HIERARCHIES



USE LEGIBLE FONTS

8

BOOST OPTIMIZED
IMAGES



PREFER VECTOR



MINIFY METADATA



PICK BEST FORMAT



AUTOMATE SCALING

9

MASTER
MINIMALISM



SOLID COLORS



OMIT REDUNDANT



LESS IS MORE



WHITESPACE

10

ELIMINATE
ERRORS



UNDERSTAND THE USER



ALLOW USER ERROR



READABILITY



COMPREHENSIVE DOCUMENTATION

Frontend Specialization : UX Designer



<https://www.youtube.com/watch?v=Ovj4hFsko7c>

Frontend Specialization : Web Developer

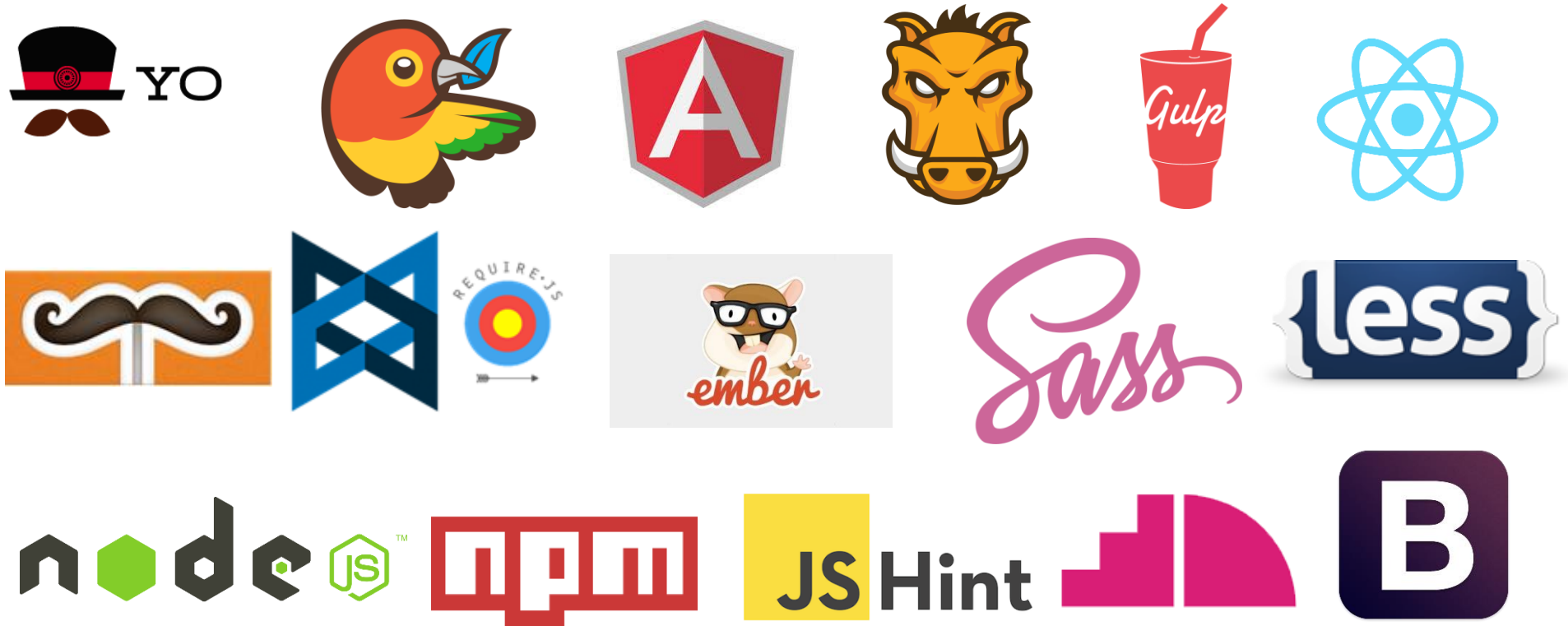


What is Frontend

A word cloud of frontend web technologies. The words are arranged in a roughly circular pattern around the center. The size of each word indicates its relative importance or frequency. 'HTML' is the largest word in the center. Other large words include 'JavaScript', 'CSS', 'DOM', 'WebComponents', 'SVG', 'HTTP', and 'WebSockets'. Smaller words include 'REST', 'AJAX', 'JSON', 'XSS', 'CORS', 'Cookie', 'Animation', 'Typography', 'Media Queries', 'User Experience', 'Local Storage', 'Navigation API', and 'Battery Status API'.

REST AJAX JSON XSS WebComponents
CORS HTTP SVG Animation
WebSockets Cookie HTML
JavaScript CSS DOM
Battery Status API Typography Media Queries
Local Storage Navigation API User Experience

Frontend Tools



User agent (desktop)

57%



25%



12%



5%



1%



User agent (mobile + tablet)

41%



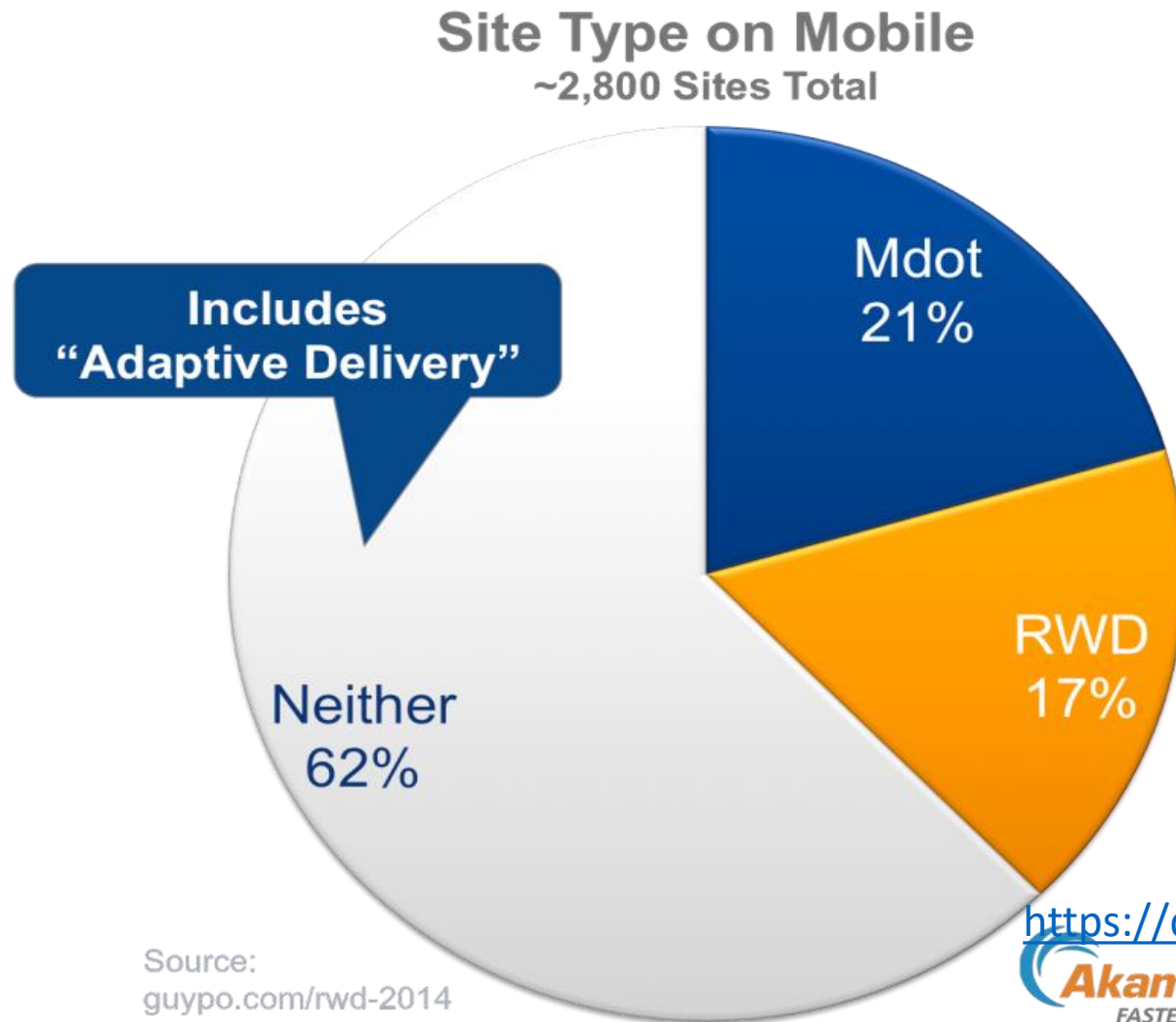
34%



11%



Mdot vs adaptive vs RWD

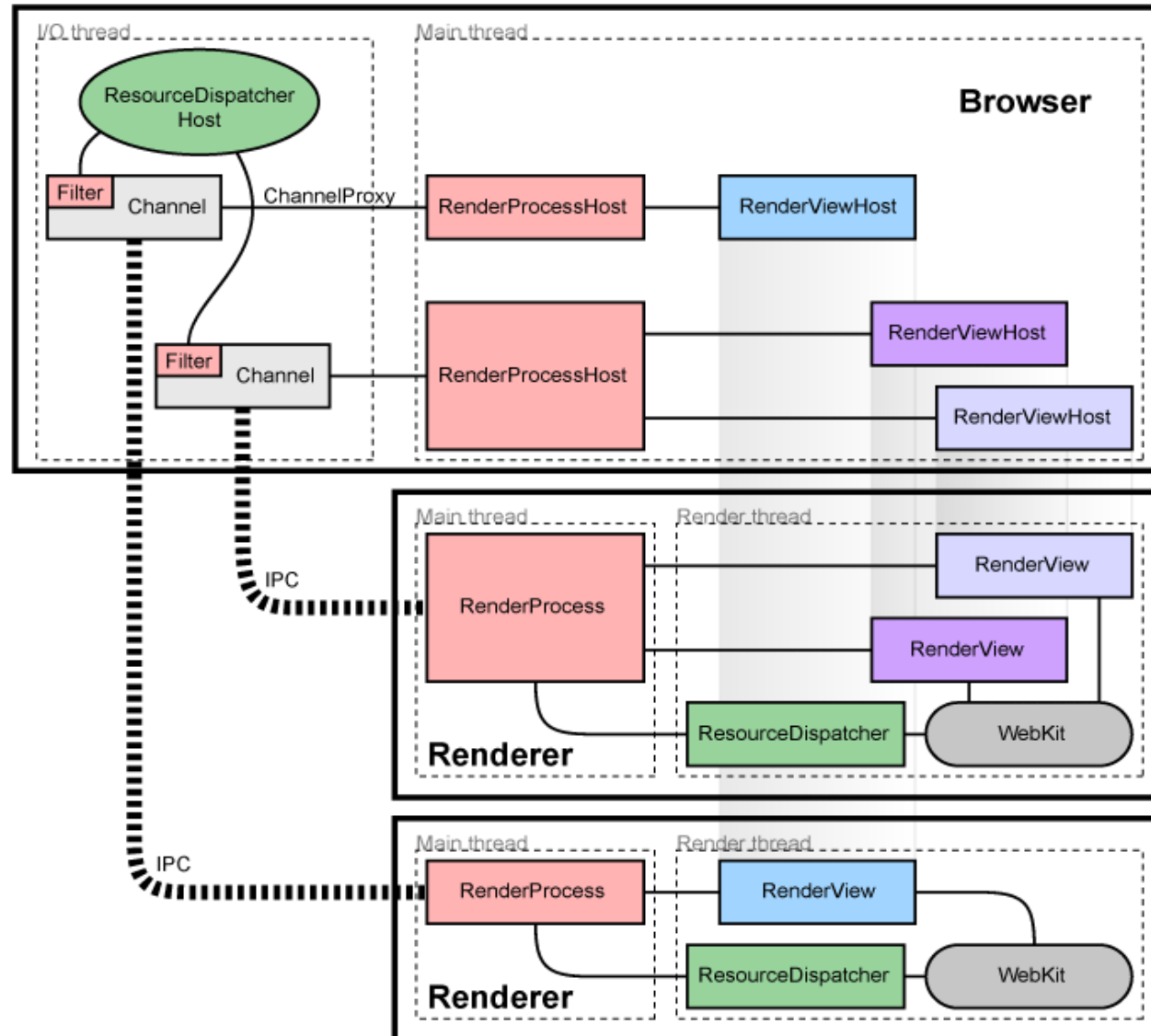


[Responsive Web Design Adoption, 2014](#)

<https://developers.google.com/webmasters/mobile-sites/mobile-seo/>

[Responsive design versus adaptive design](#)

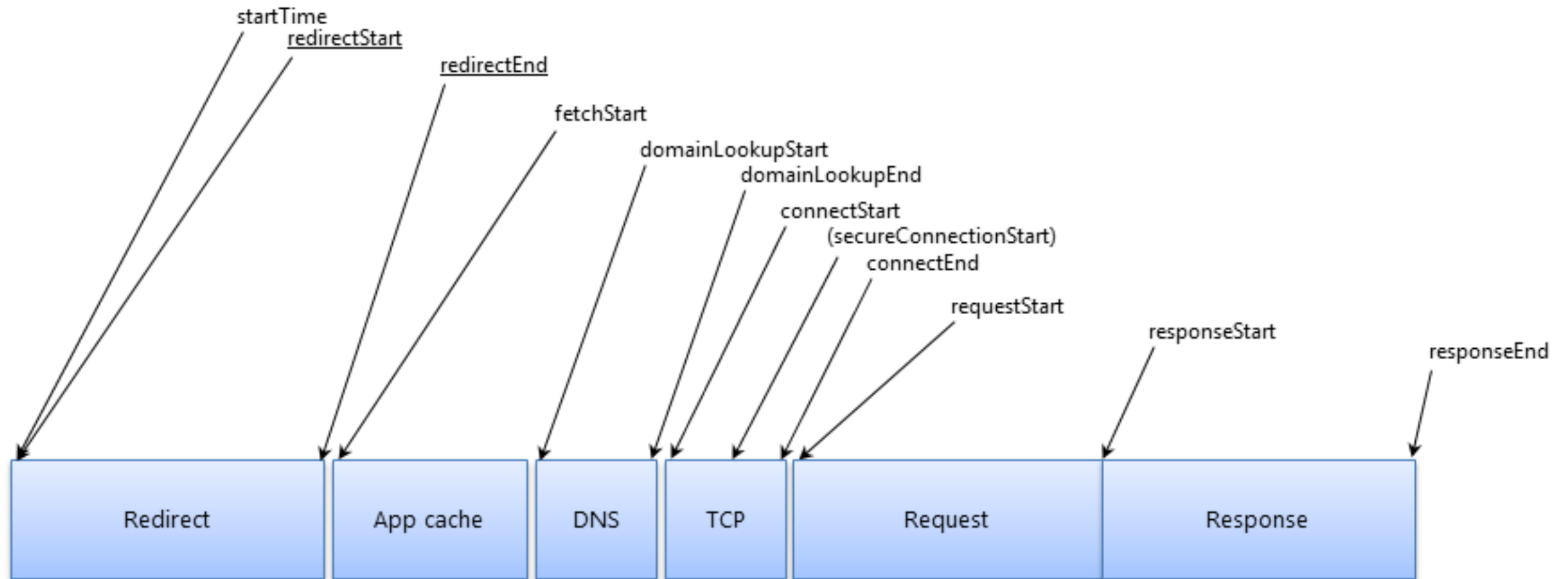
High level browser architecture



[Multi Process Architecture](#)

[Mozilla Electrolysis](#)

Networking



RESPONSE TIME LIMITS

INSTANTANEOUS  100ms

SEAMLESS  1s

LEAVE A SITE  10sec

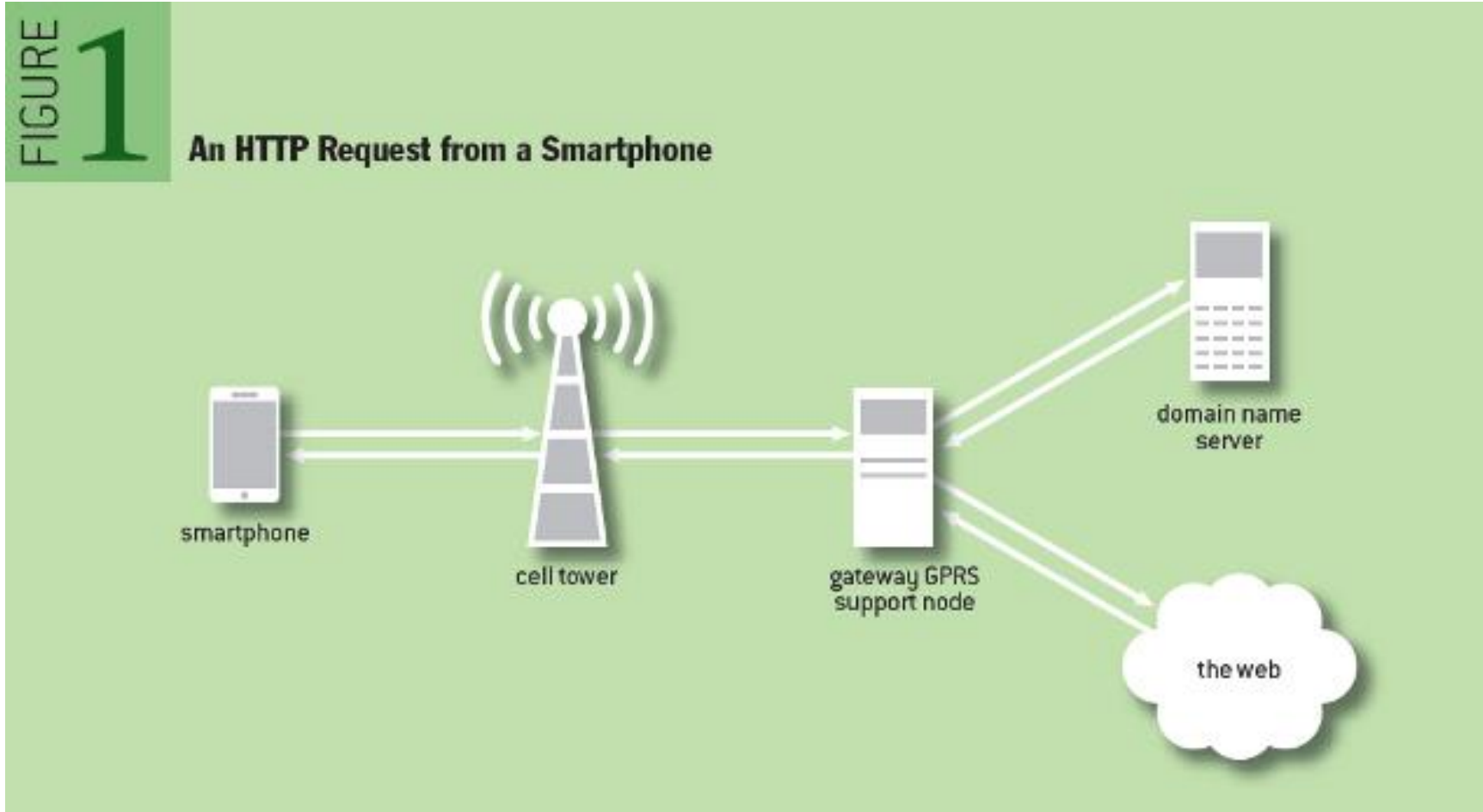
Every 10 seconds, **5500** hot dogs eaten in America.

[Website Response Times](#)

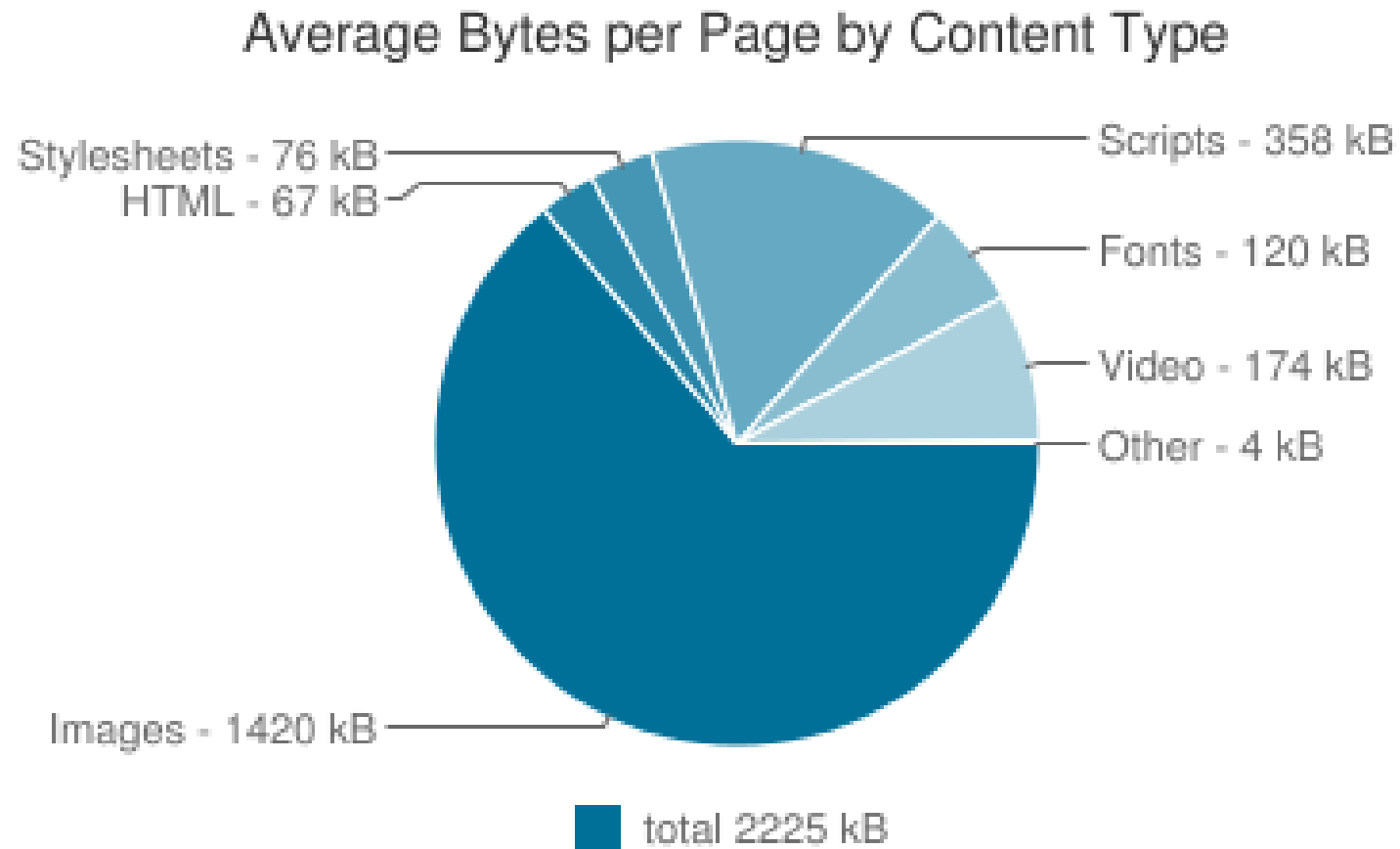
Network latency

Route	Distance, km	Light in Vacuum, ms	Light in Fiber, ms	RTT in Fiber, ms
MINSK – LONDON	2,162	7	11	22
MINSK – NEW YORK	7,118	24	36	72
MINSK – SYDNEY	15,131	51	76	152
EQUATORIAL CIRCUMFERENCE	40,075	133.7	200	200

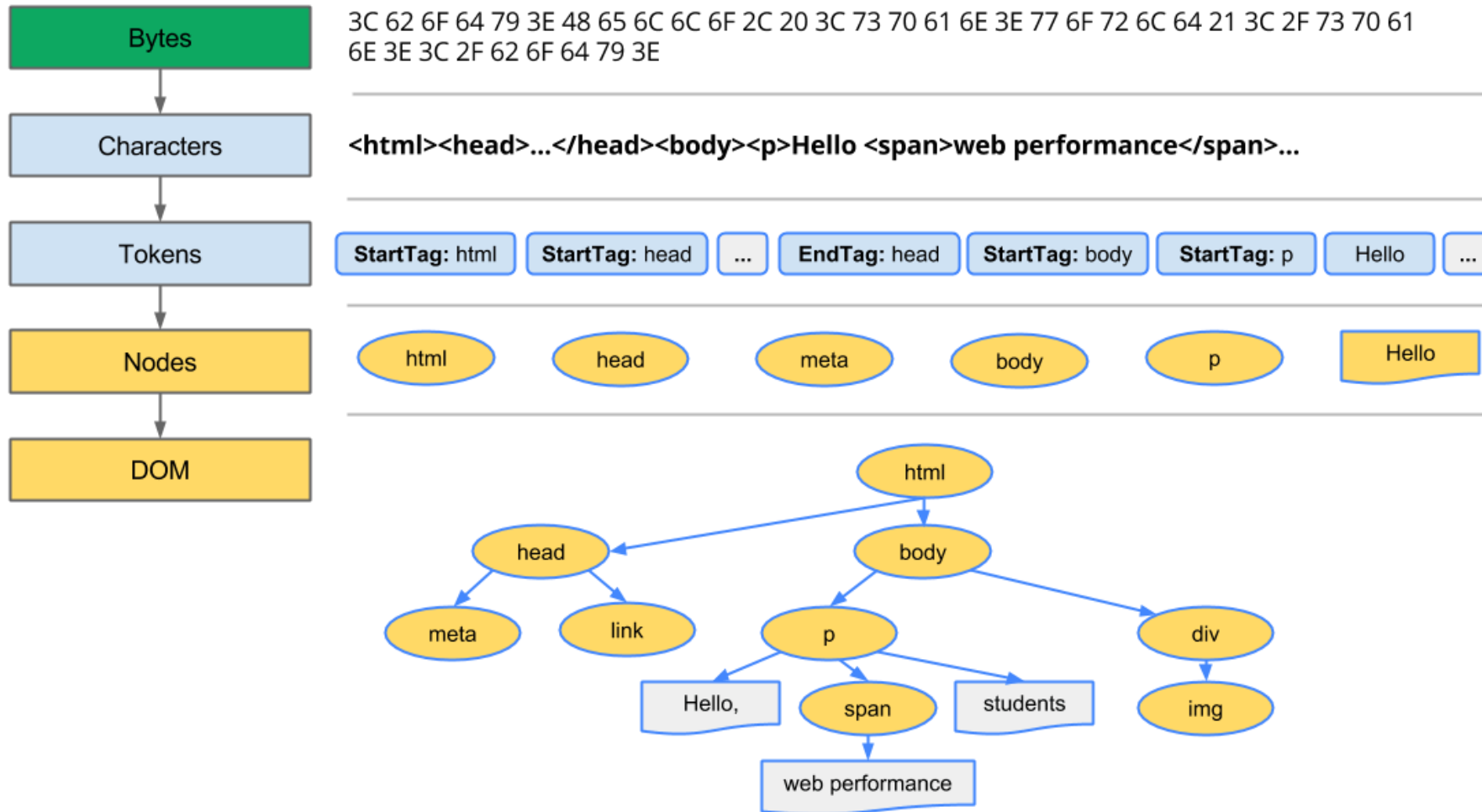
3G Mobile Network



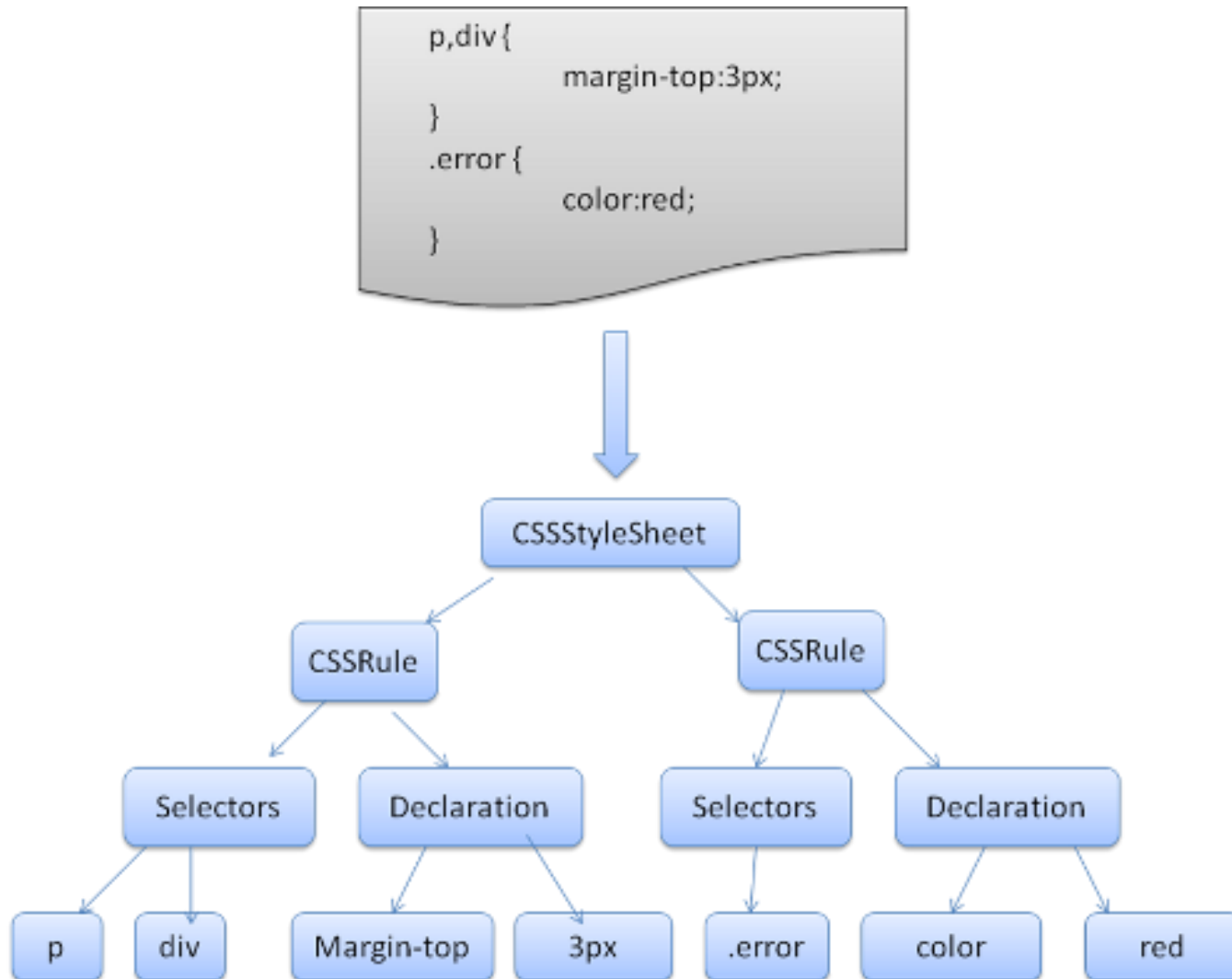
Http archive : modern trend & stats



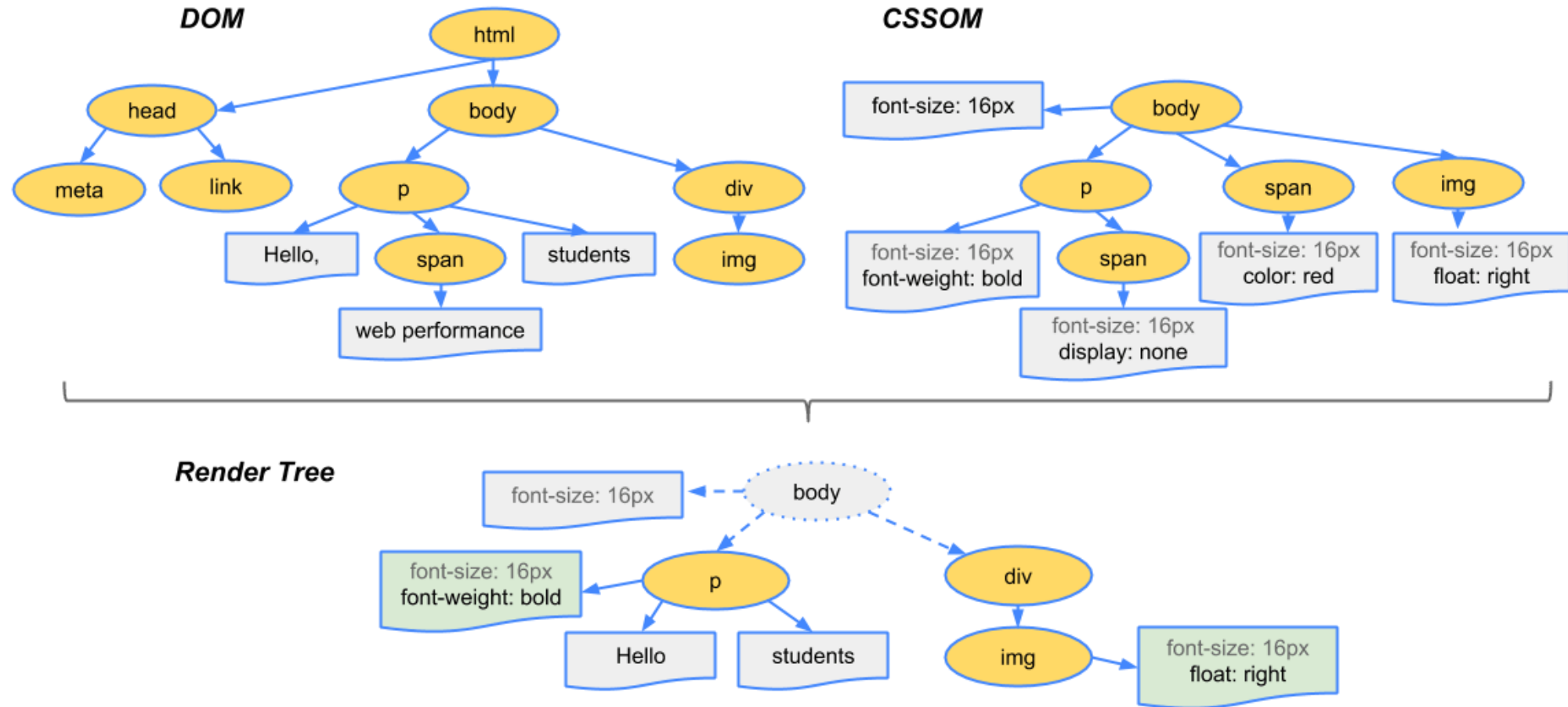
HTML parsing - DOM



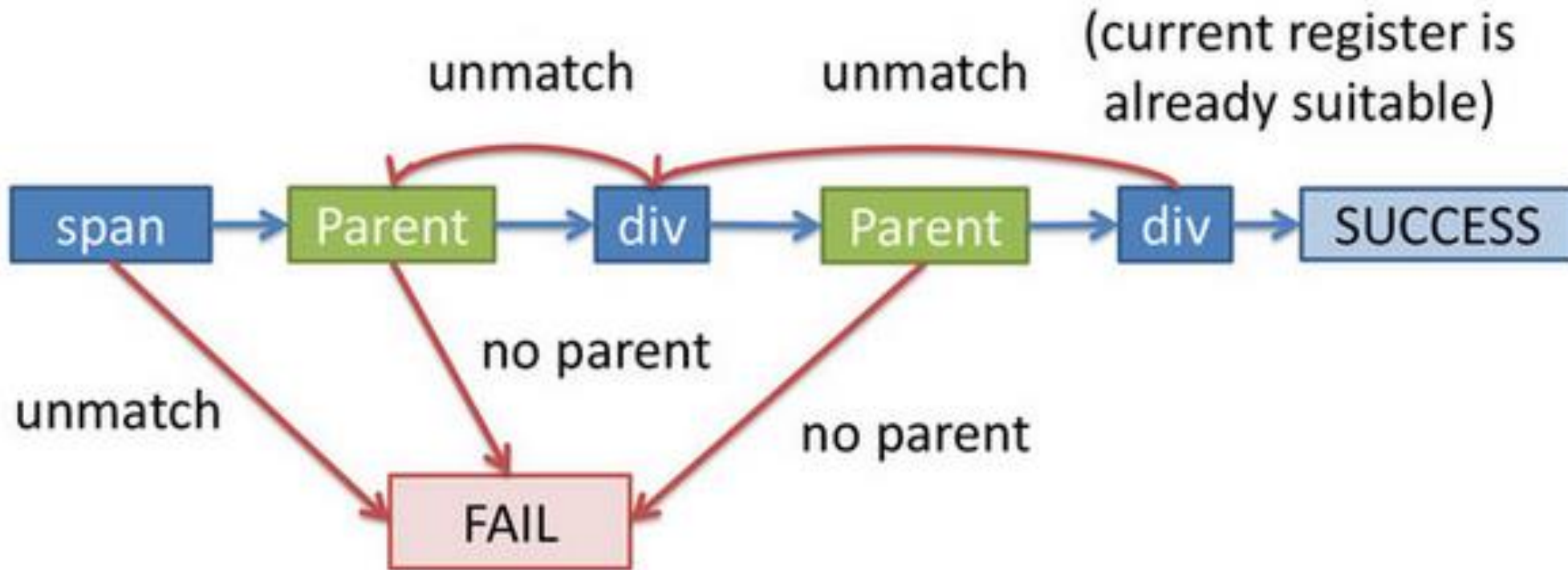
CSS Parsing - CSSOM



Render tree



CSS Selector : div > div span



JIT Code : div > div span

Generated JIT code for CSS Selector JIT for "div > div span":

Code at [0x7fd031187000, 0x7fd0311870a0):

```
0x7fd031187000: push %rbp
0x7fd031187001: mov 0x58(%rdi), %rax
0x7fd031187005: mov $0x7fd08c30ec50, %rcx
0x7fd03118700f: cmp %rcx, 0x18(%rax)
0x7fd031187013: jnz 0x7fd031187081
0x7fd031187019: mov 0x20(%rdi), %rdi
0x7fd03118701d: test %rdi, %rdi
0x7fd031187020: jz 0x7fd031187081
0x7fd031187026: test $0x4, 0x1c(%rdi)
0x7fd03118702a: jz 0x7fd031187081
0x7fd031187030: mov 0x58(%rdi), %rdx
0x7fd031187034: mov $0x7fd08c30f670, %rsi
0x7fd03118703e: cmp %rsi, 0x18(%rdx)
0x7fd031187042: jnz 0x7fd031187019
0x7fd031187048: mov 0x20(%rdi), %rdi
0x7fd03118704c: test %rdi, %rdi
0x7fd03118704f: jz 0x7fd031187081
```

```
0x7fd031187055: test $0x4, 0x1c(%rdi)
0x7fd031187059: jz 0x7fd031187081
0x7fd03118705f: mov 0x58(%rdi), %r8
0x7fd031187063: mov $0x7fd08c30f670, %r9
0x7fd03118706d: cmp %r9, 0x18(%r8)
0x7fd031187071: jnz 0x7fd031187019
0x7fd031187075: mov $0x1, %eax
0x7fd031187079: jmp 0x7fd031187083
0x7fd031187081: xor %eax, %eax
0x7fd031187083: pop %rbp
0x7fd031187084: ret
```

SUCCESS

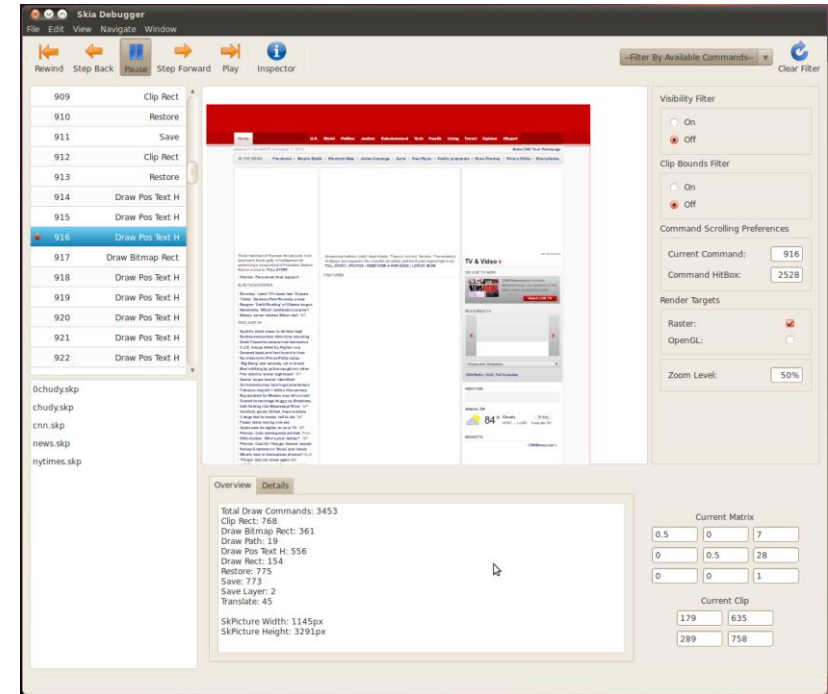
FAIL

- No call jumps, no recursions
- Storing to registers only

Browser Graphics



An open source 2D graphics library which provides common APIs that work across a variety of hardware and software platforms. It serves as the graphics engine for Google Chrome and Chrome OS, Android, Mozilla Firefox and Firefox OS, and many other products.

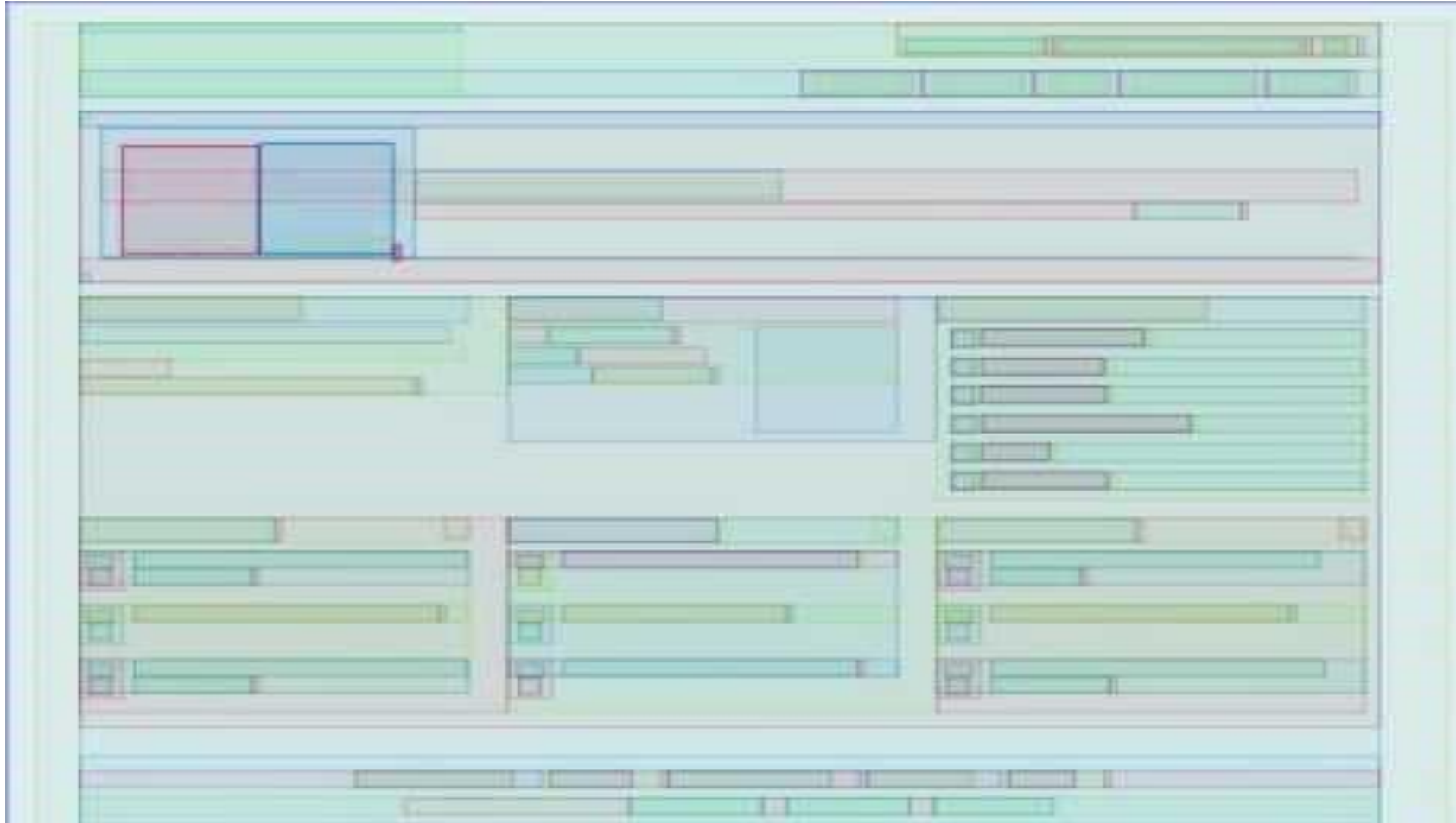


[SKIA](#)

[Skia Debugger](#)

[CSS Paint Times and Page Render Weight](#)

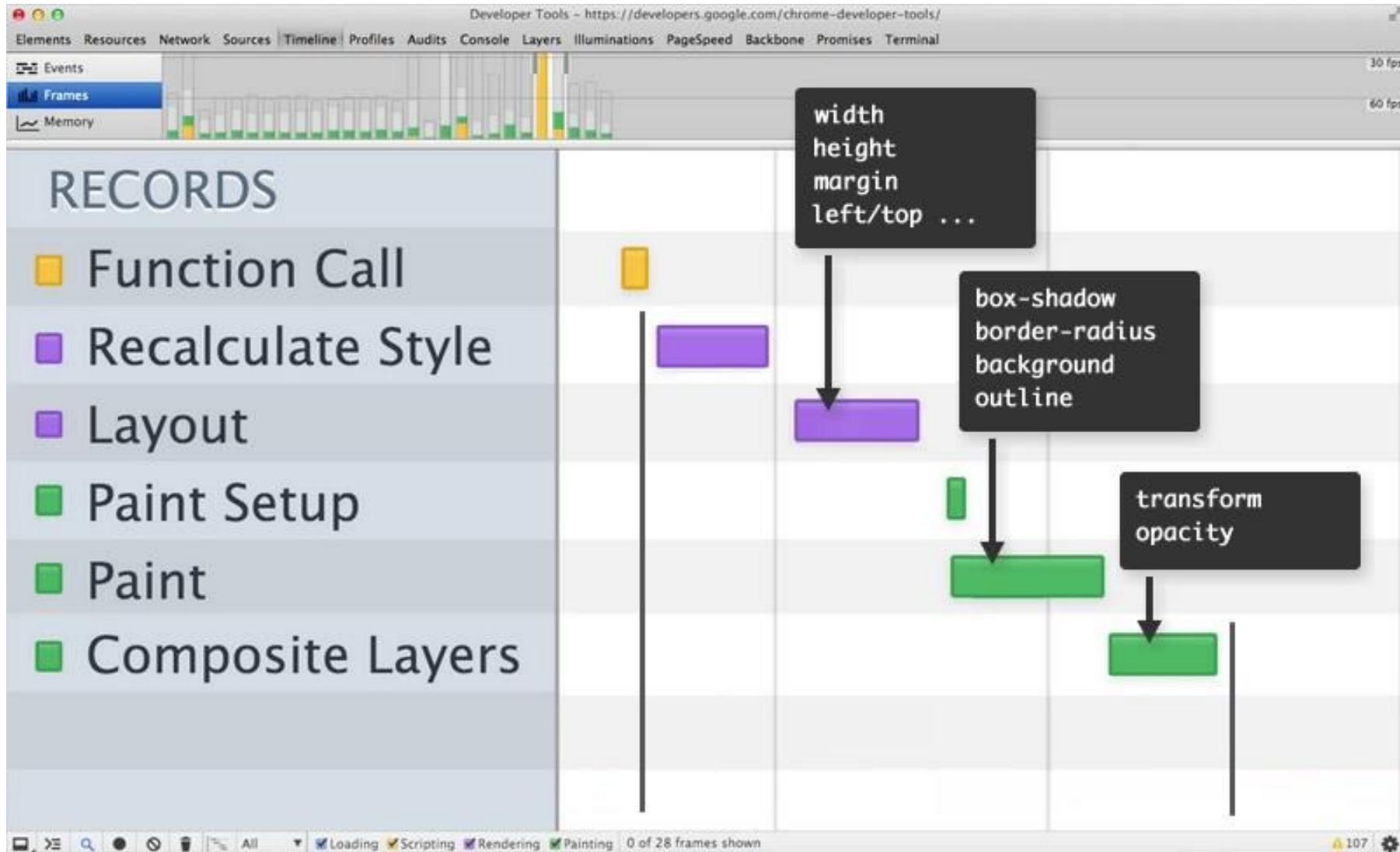
Page layout (video)



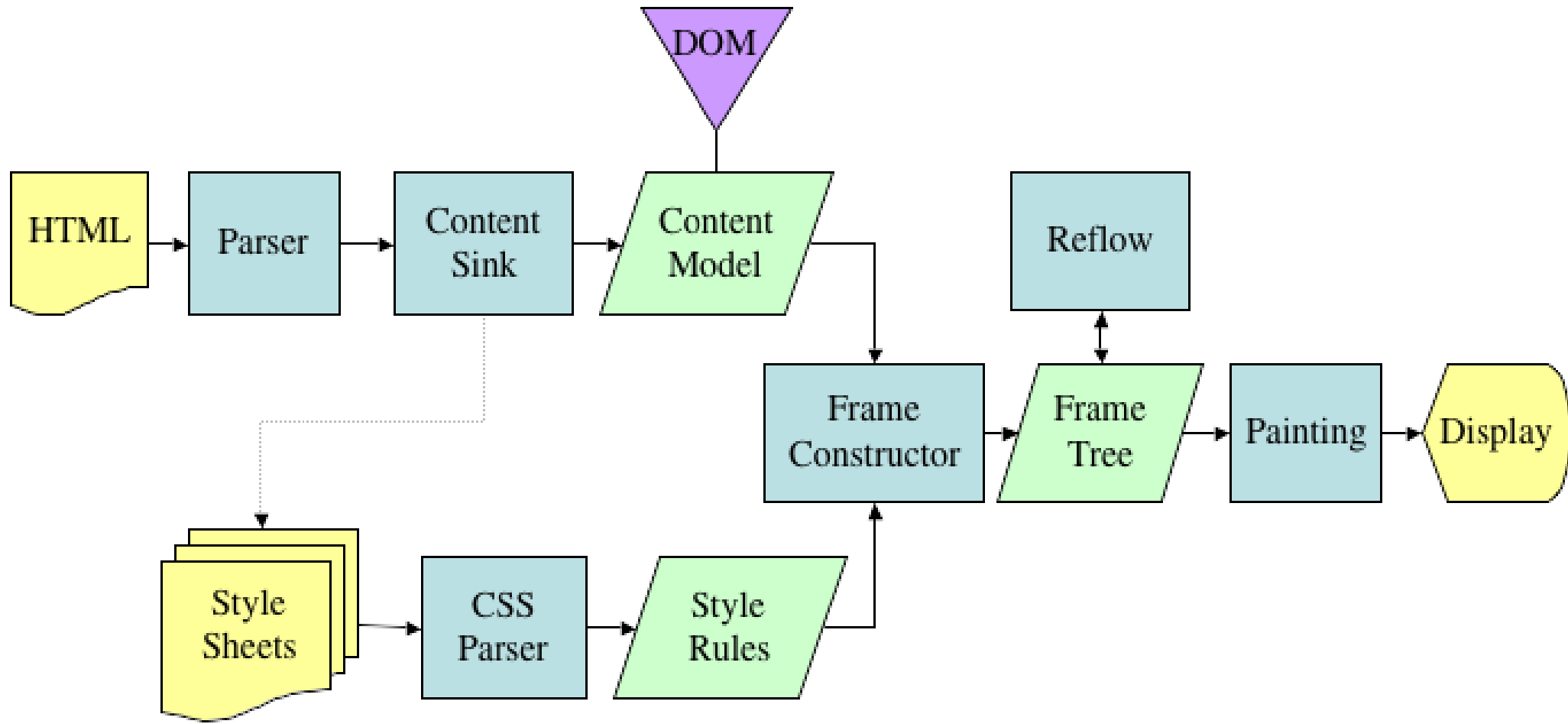
Composing layers



Reflow & Repaint



Render Page Summary

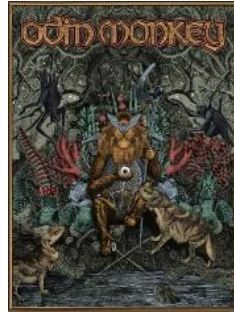


JS Engines



V8 Crankshaft

- [V8 JavaScript Engine](#)
- [Crazy Russian compiler engineer that talks.](#)



Spider Monkey

- [Mozilla SpiderMonkey](#)



Chakra

- [Architecture Overview](#)

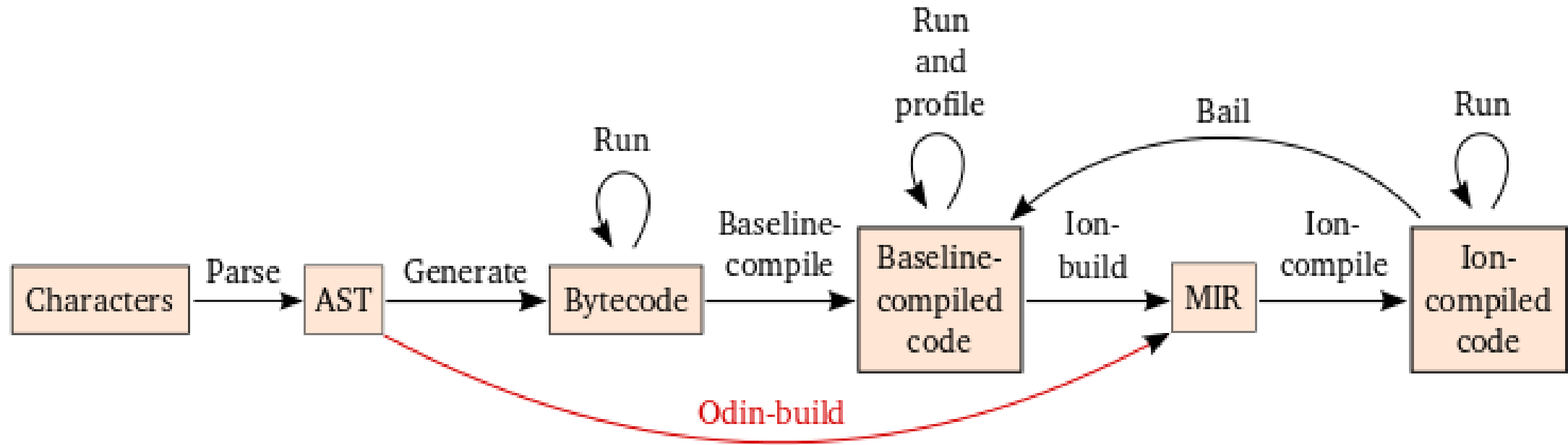


Nitro

- [JavaScriptCore](#)

[Performance Comparison](#)

Javascript JIT compilation



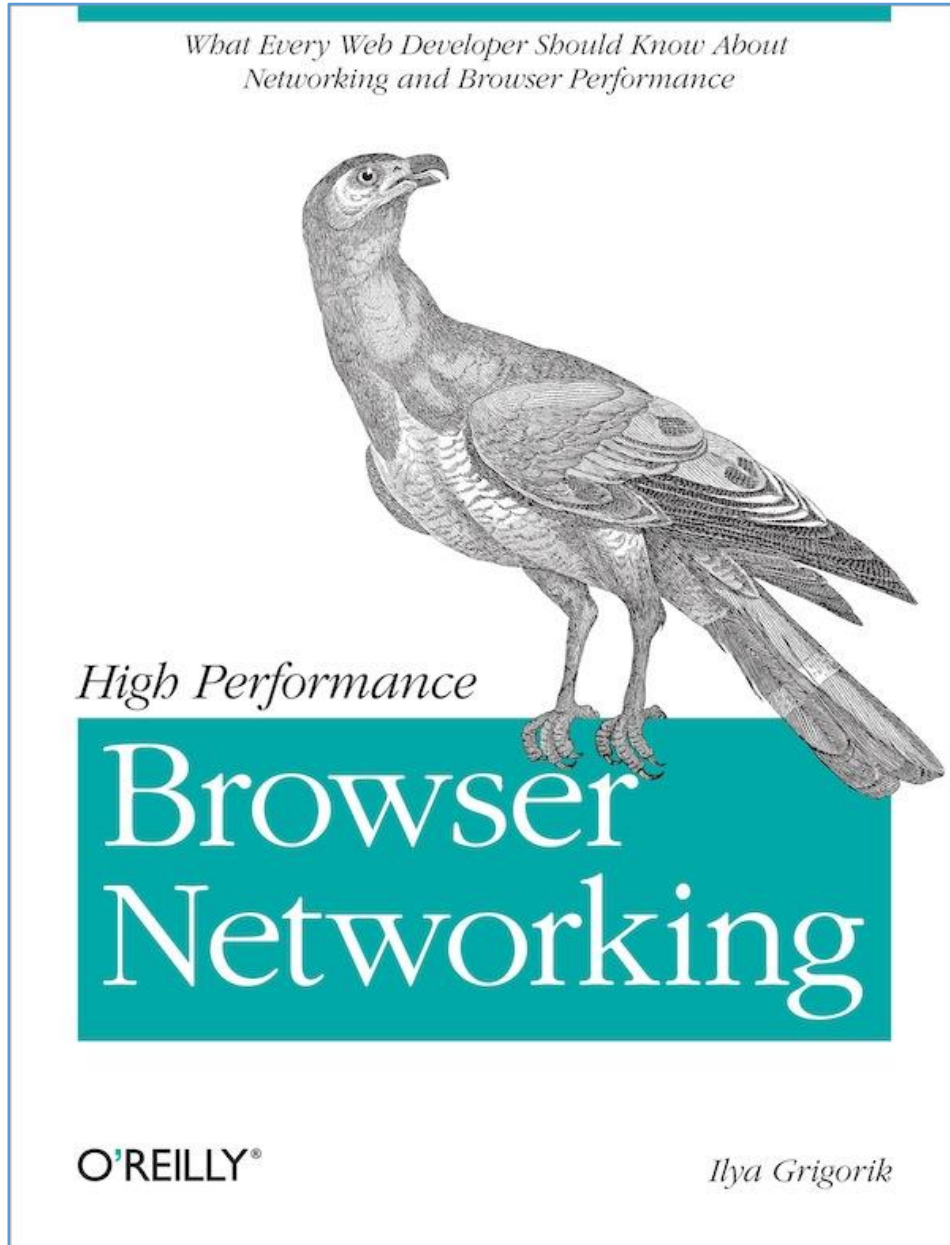
[v8: a tale of two compilers](#)

[asm.js AOT compilation and startup performance](#)

[Introducing the WebKit FTL JIT](#)

How to start learning frontend:

- <https://developer.mozilla.org/en-US/Learn>
- <https://developers.google.com/web/>
- [http://docs.webplatform.org/wiki/Main Page](http://docs.webplatform.org/wiki/Main_Page)
- How browsers work:
<http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>



Ilya Grigorik

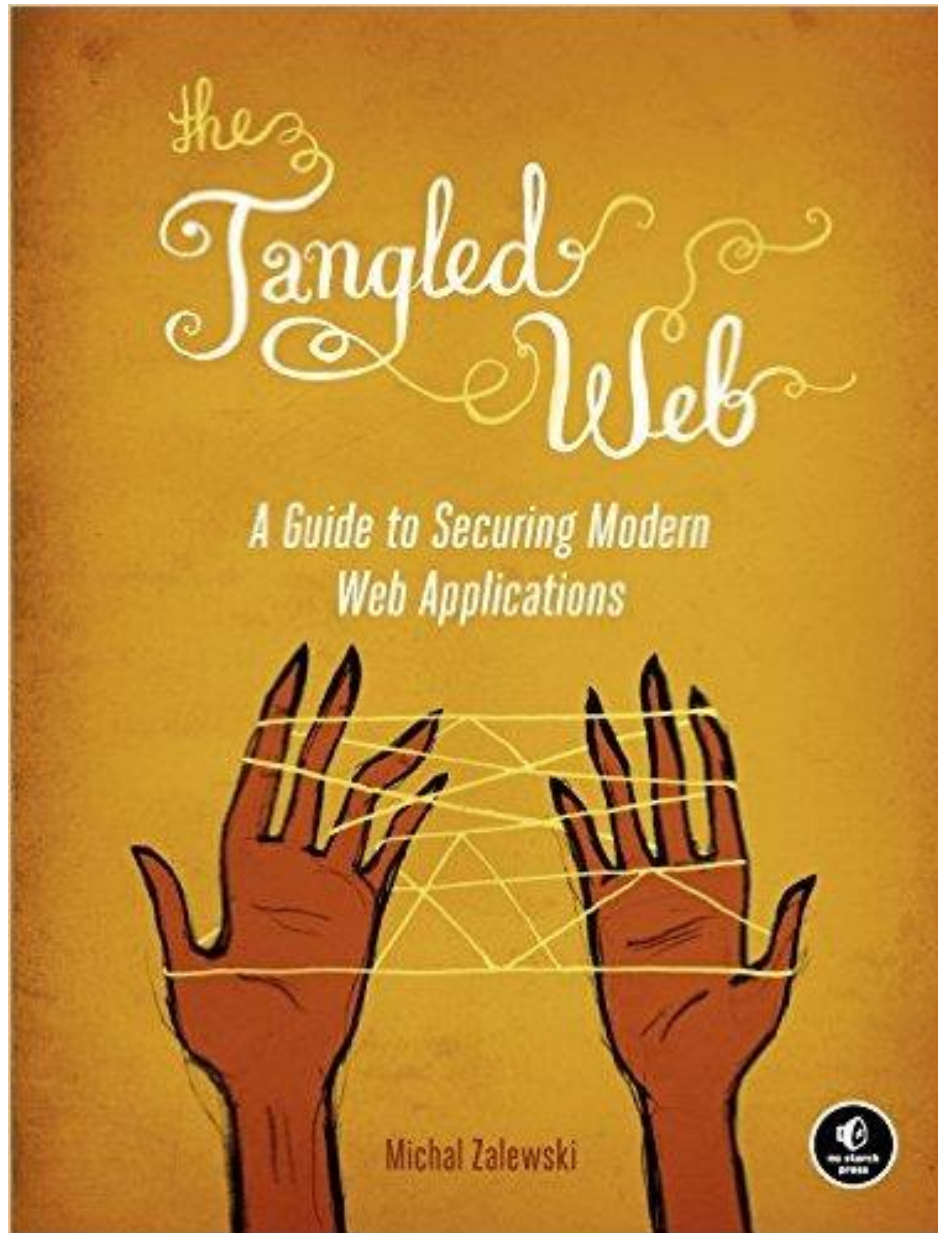
High Performance

Browser Networking

<http://chimera.labs.oreilly.com/books/12300000000545/index.html>



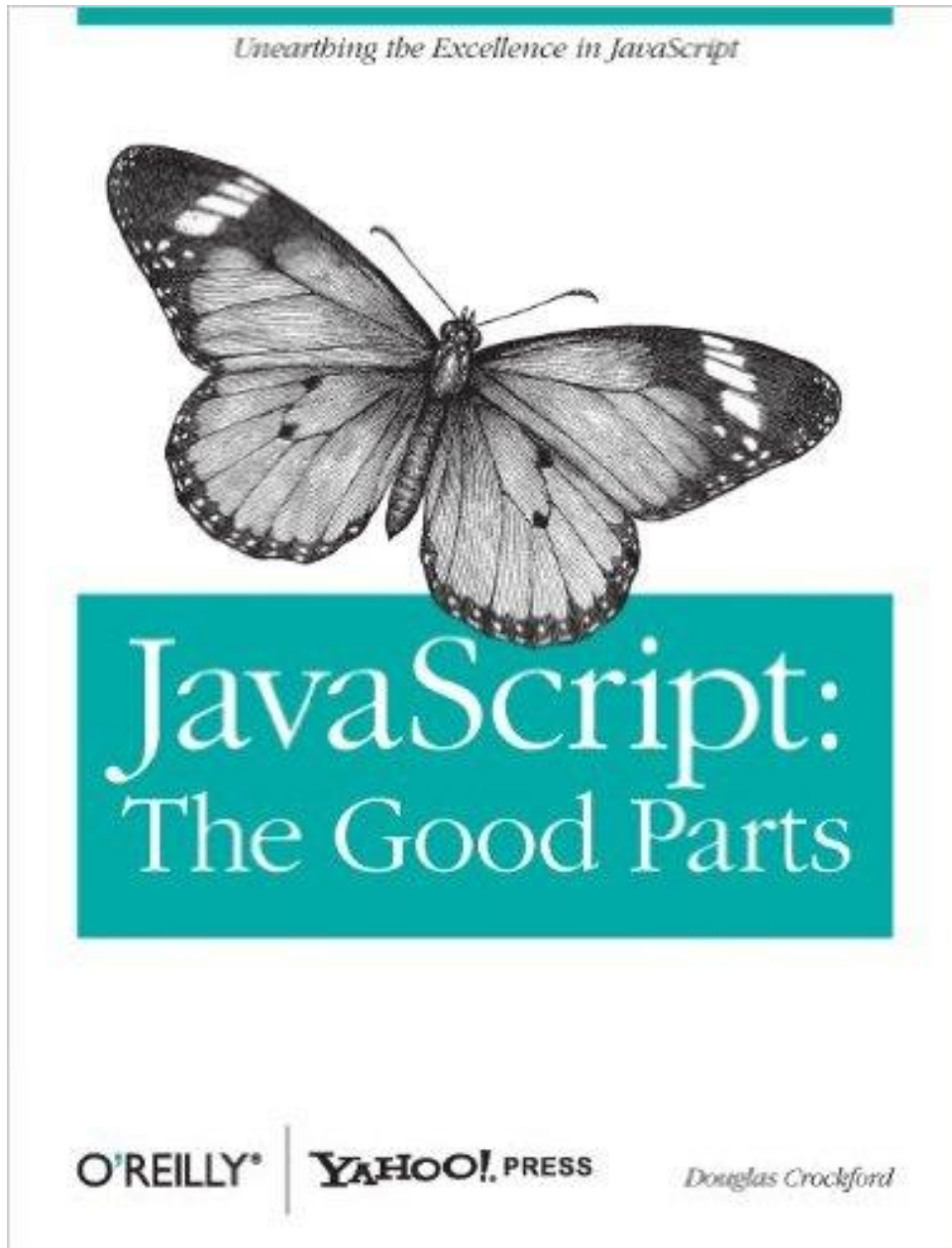
<https://plus.google.com/+IlyaGrigorik>



Michal Zalewski

The Tangled Web

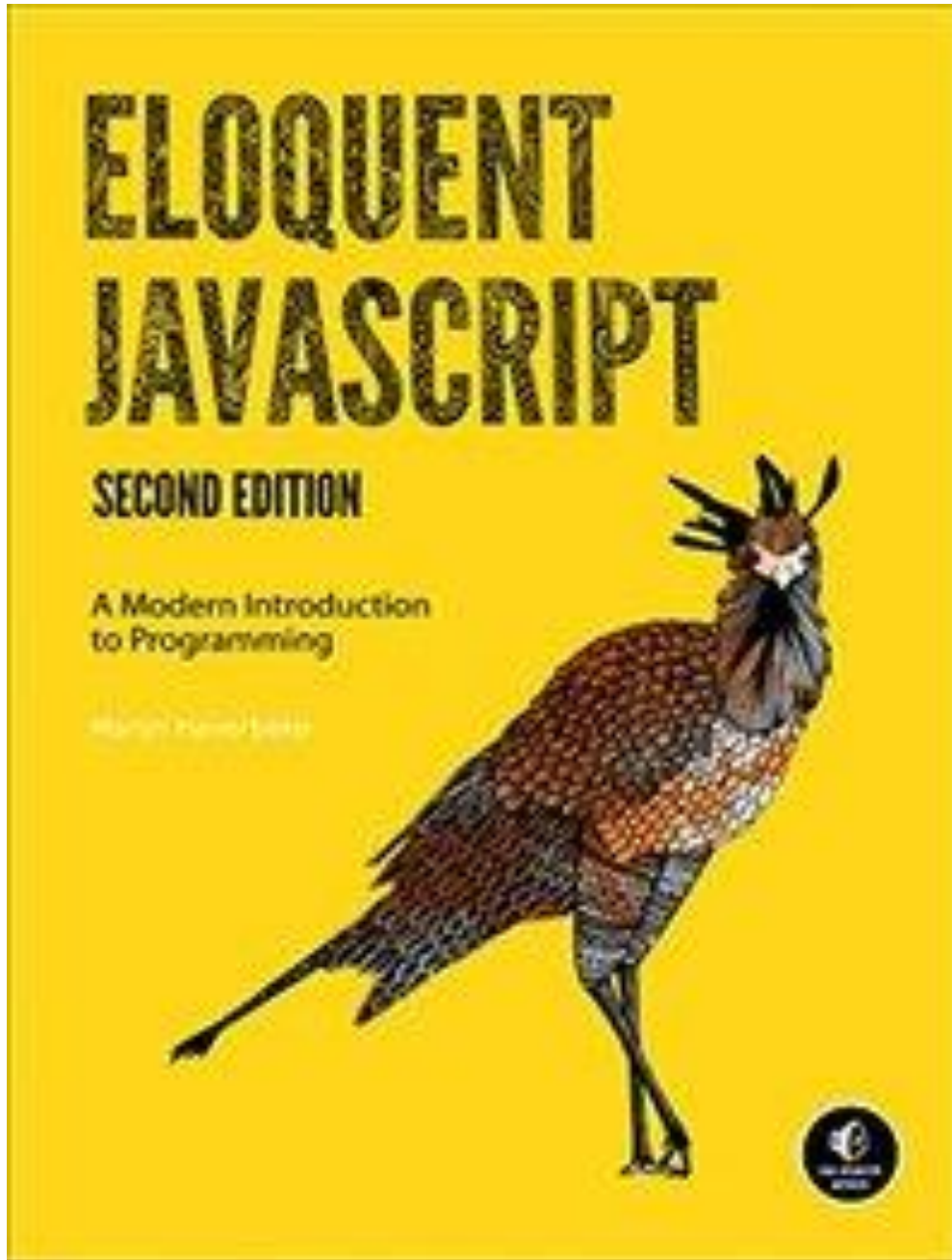
<http://www.amazon.com/The-Tangled-Web-Securing-Applications/dp/1593273886>



Douglas Crockford

JavaScript: The Good Parts

<http://www.amazon.com/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742/>



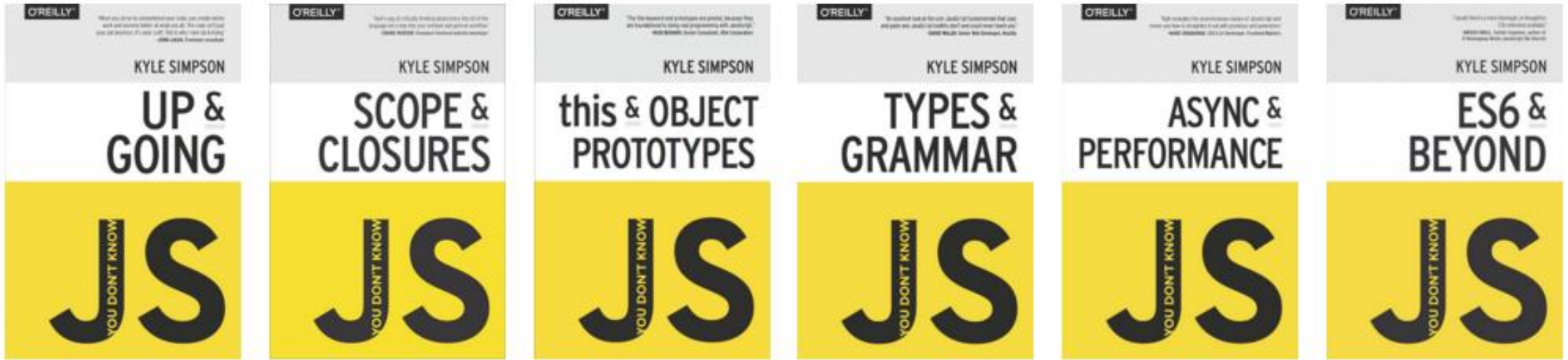
Marijn Haverbeke

Eloquent

Javascript

<http://www.amazon.com/Eloquent-JavaScript-Modern-Introduction-Programming/dp/1593275846/>

https://karmazzin.gitbooks.io/eloquentjavascript_ru/



Kyle Simpson

You don't know JS

<https://github.com/getify/You-Dont-Know-JS>

Frontend video courses

- <https://academy.yandex.ru/events/shri/ekb-2013/>
- <https://academy.yandex.ru/events/shri>
- <https://park.mail.ru/materials/video/#9>
- <https://frontendmasters.com/> (commercial)
- <https://www.youtube.com/playlist?list=PLEzQf147-uEpvTa1bHDNIxUL2klHUMHJu> Douglas Crockford channel

Questions?

