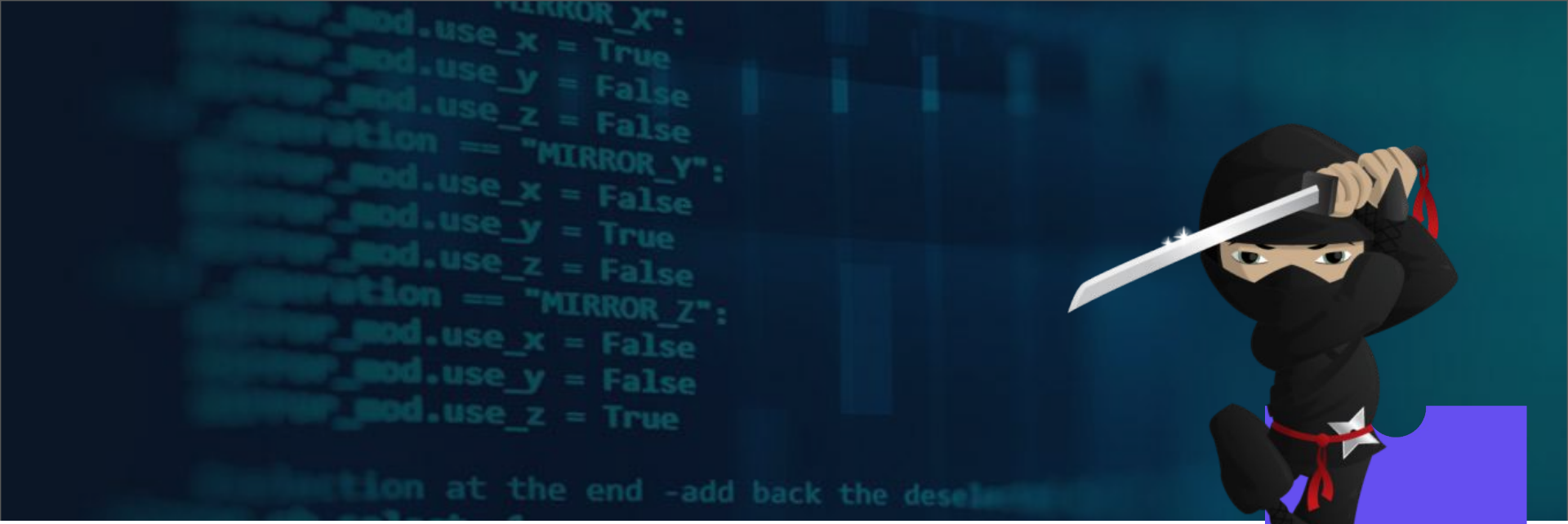
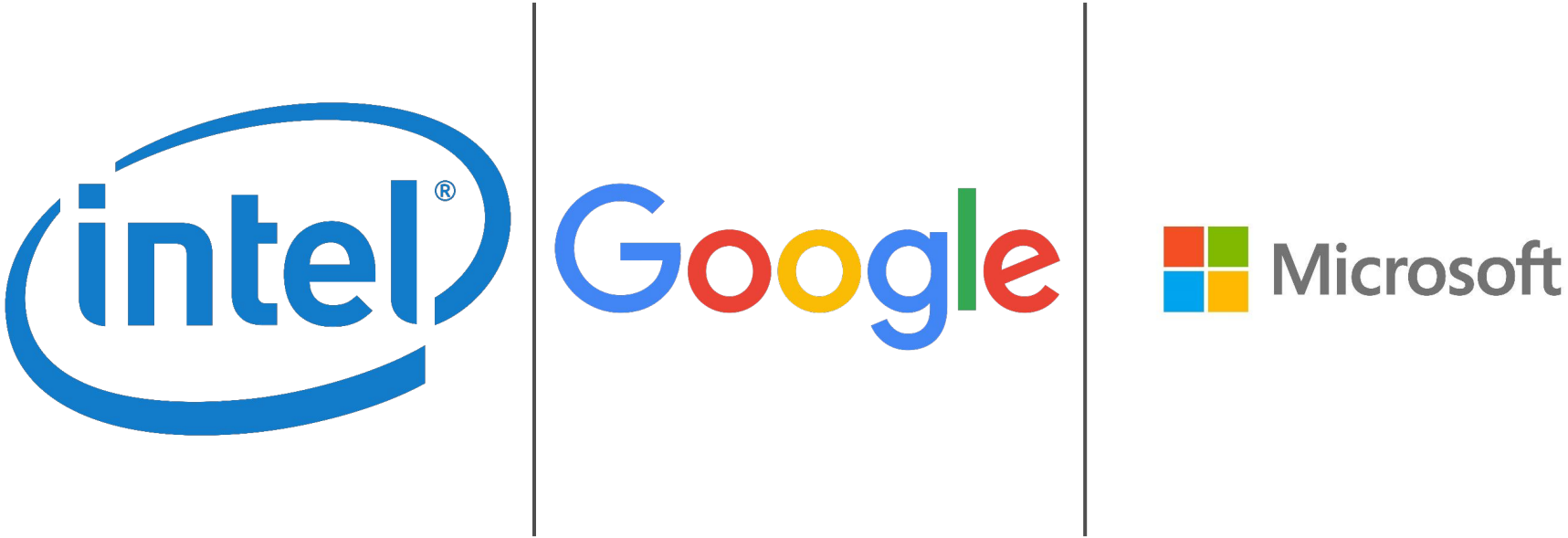



# Тренды web стандартов

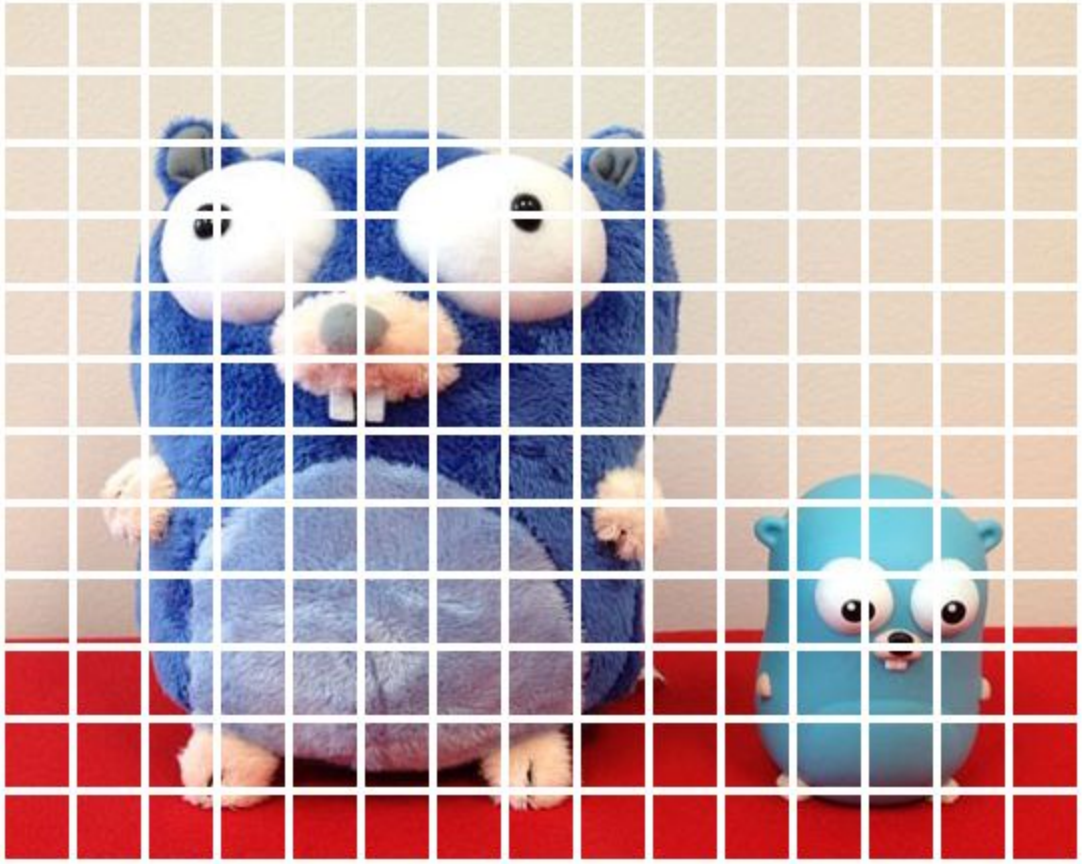
Сергей Пугачёв, Google


intel
Google
Microsoft


Welcome To
The Future

HTTP/2

HTTP 1.1

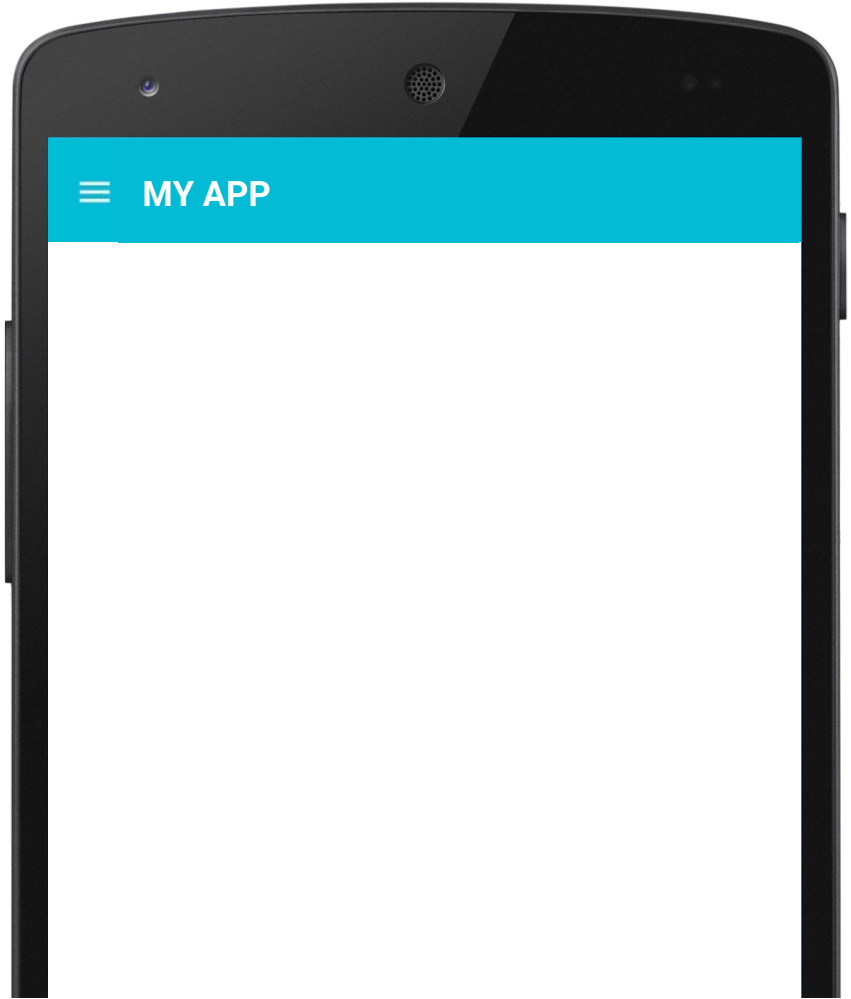


A grid of 180 tiled images is below. Compare:

[HTTP/2, 0 latency] [HTTP/1, 0 latency]
[HTTP/2, 30ms latency] [HTTP/1, 30ms latency]
[HTTP/2, 200ms latency] [HTTP/1, 200ms latency]
[HTTP/2, 1s latency] [HTTP/1, 1s latency]

A grid of 180 tiled images is below. Compare:

[HTTP/2, 0 latency] [HTTP/1, 0 latency]
[HTTP/2, 30ms latency] [HTTP/1, 30ms latency]
[HTTP/2, 200ms latency] [HTTP/1, 200ms latency]
[HTTP/2, 1s latency] [HTTP/1, 1s latency]




# Web Components

<paper-toolbar/>

```
<paper-toolbar>
  <paper-icon-button slot="top"
  icon="menu">
  </paper-icon-button>
  <div class="title">MY APP</div>
</paper-toolbar>
```

# CSS

- Grid Layout
- Flexbox
- Native Variables
- Animations


1997
1998
1999
2009
2015
2016
2017
ES1
ES2
ES3
ES5
ES6 / ES2015
ES7 / ES2016
ES8 / ES2017

# ECMAScript 2017

```
import * as utils from './utils';

export class WashHelper {
   async loadWasm(fileName){
       const res = await fetch(`/${fileName}.wasm`);
       const bytes = await res.arrayBuffer();
       const module = await WebAssembly.compile(bytes);
       const instance = await WebAssembly.instantiate(module);
       return {instance, module};
   }
}
```

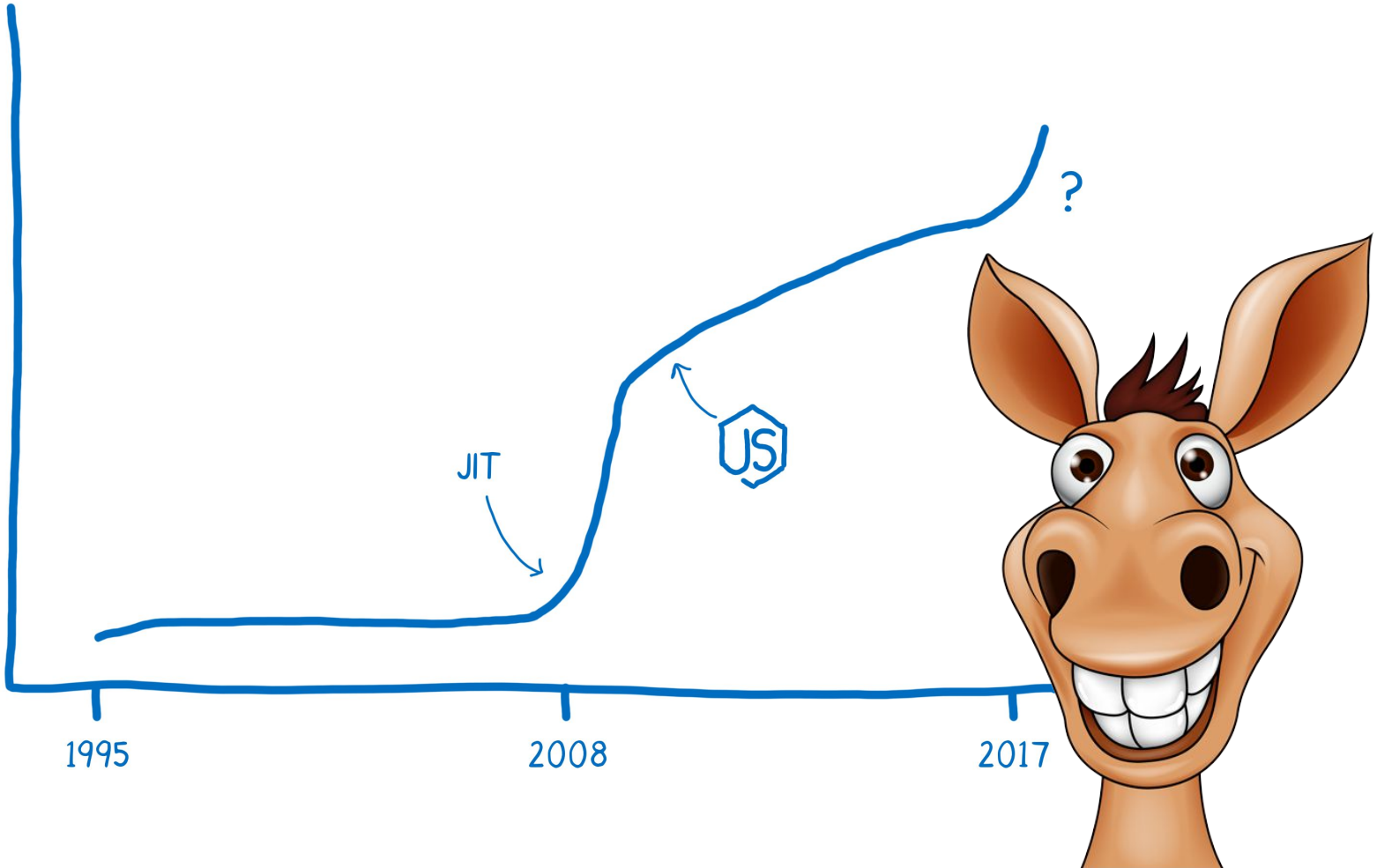
## WebAssembly

Нативная сверхбыстрая языконезависимая виртуальная машина в браузере без плагинов

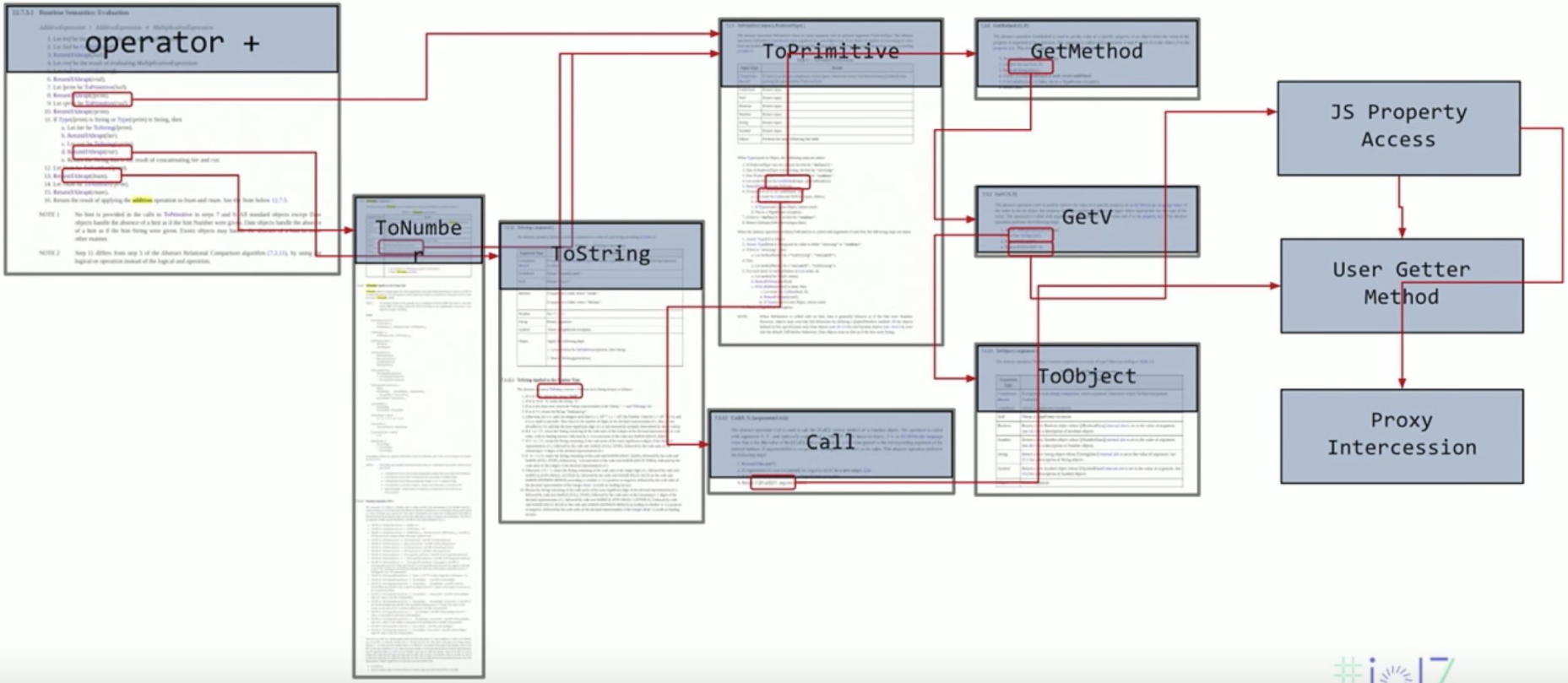
## «Нативные» возможности

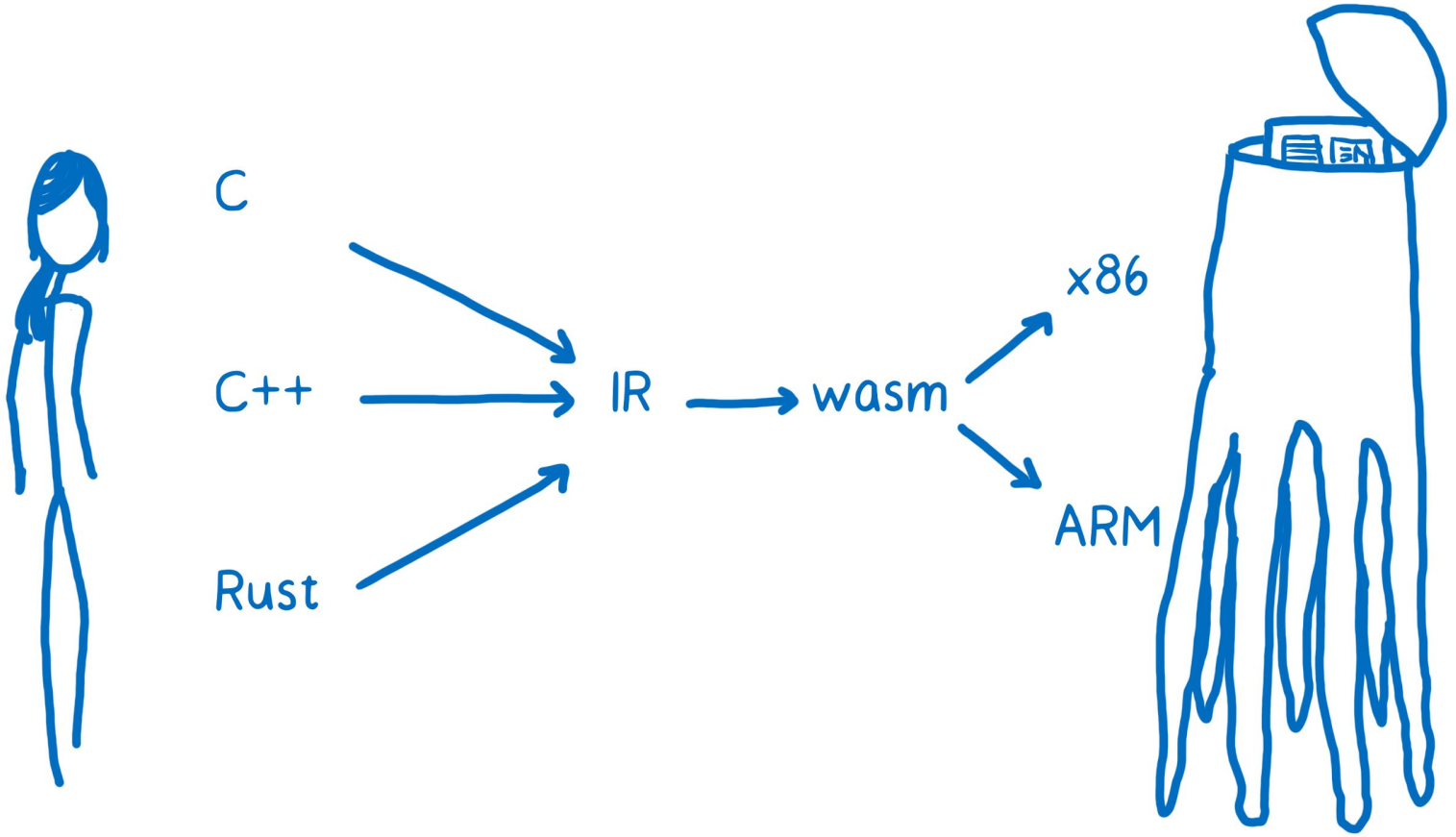
Progressive Web Apps (PWA) & Service Workers, Notifications, Offline, Payment Request, Web Bluetooth, Web USB, Web Share, Sensors, Battery, WebVR и т.д.


WA
WEBASSEMBLY

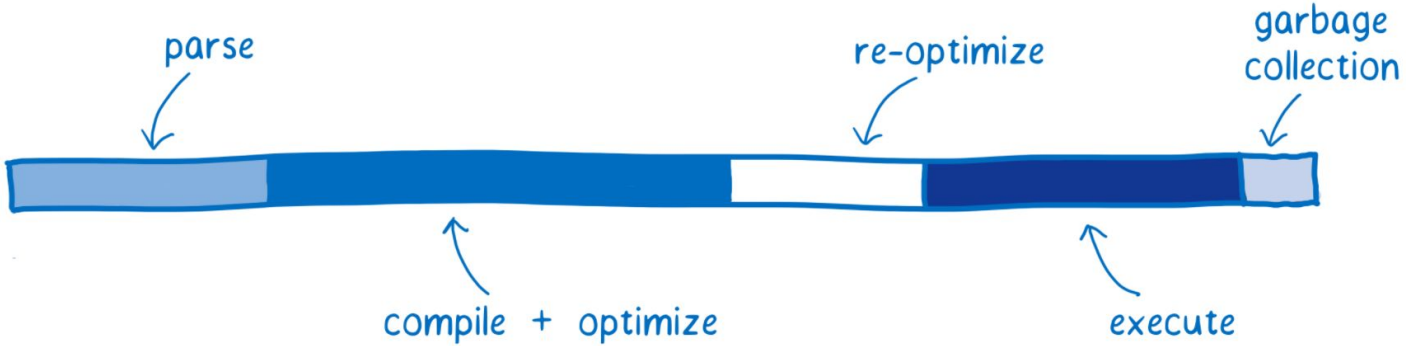

?
JIT
1995
2008
2017

# JS Semantics for '+'

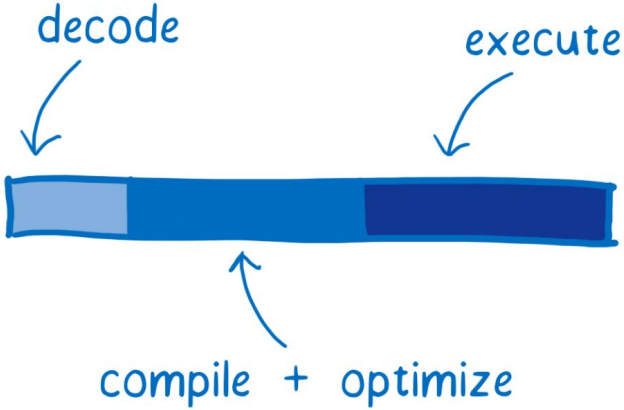



C
C++
Rust
IR
wasm
x86
ARM

# JavaScript




# WASM



# WebAssembly

- Data Types
  - `void i32 i64 f32 f64`
- Functions
  - Flat, single global table
  - Static binding
  - Indirect calls through table
- State: linear memory
  - large, bounds-checked array
- Trusted execution stack

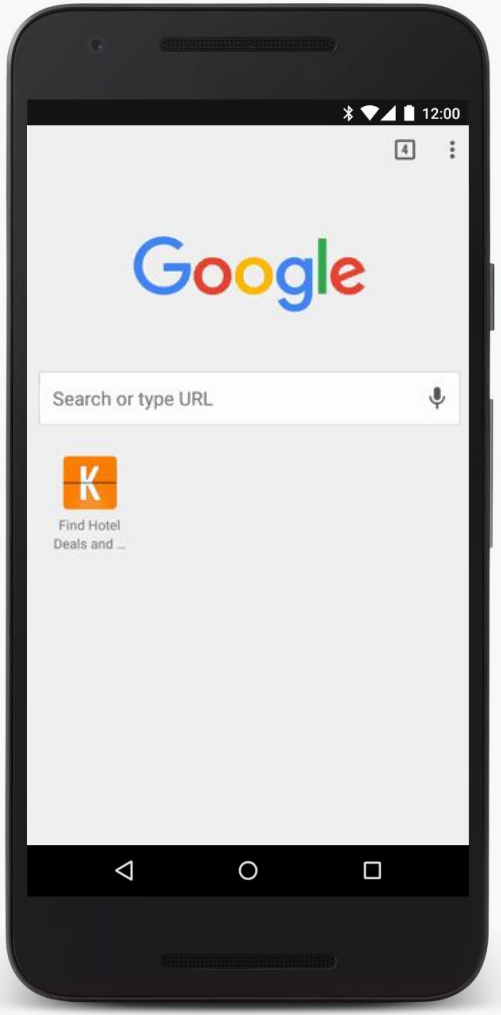
- Data Operations
  - `i32: + - * / % << >> >>> etc`
  - `i64: + - * / % << >> >>> etc`
  - `f32: + - * / sqrt ceil floor`
  - `f64: + - * / sqrt ceil floor`
  - `conversions`
  - `load store`
  - `call_direct call_indirect`
- Structured Control Flow
  - `if loop block br switch`

# ДЕМО
WebAssembly

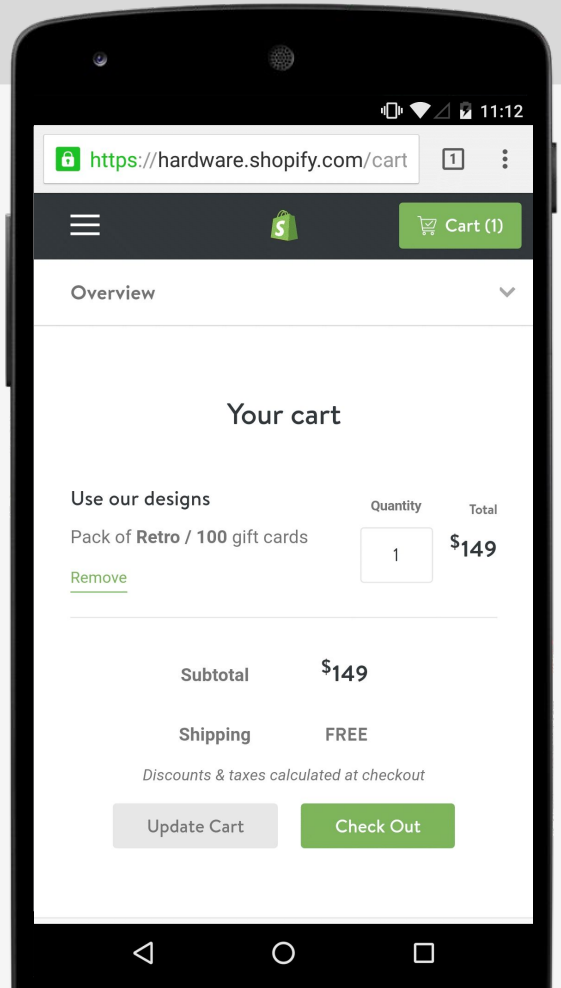
# Progressive Web Apps (PWA)


- **Reliable:** Быстрая загрузка, работа оффлайн и с плохими сетями
- **Fast:** Быстрые анимации, прокрутка и навигация
- **Engaging**
  - Иконка на Home Screen, нет окна браузера, поддержка заставок (splash screen)
  - Вовлечение с помощью Push-нотификаций

# Credentials API
# Автоматический вход



# Web Payments API




```
let supportedInstruments = [{
   supportedMethods: ['visa', 'mastercard', 'amex']
}];

let details = {
  displayItems: [{
    label: 'Original donation amount',
    amount: { currency: 'USD', value: '65.00' }
  }
};

let options = { requestShipping: true,
  requestPayerEmail: true, requestPayerPhone: true };

let request = new PaymentRequest(
supportedInstruments, details, options);

request.show().then(result => {});
```


ДЕМО
Progressive Web Apps

# Вопросы?

Сергей Пугачёв
@spugachev