CLOUDFLARE®

Massive Multitenancy with V8 Isolates

Kenton Varda - Tech Lead, Cloudflare Workers

The Challenge

CLOUDFLARE®

165 Locations and growing

Scalability can mean…

Traffic (requests)

Easy: More locations = more capacity.

Tenants (apps)

Hard: Every tenant in every location.
Some locations are small!

Needed:

>100x Efficiency

I, **Kenton Varda**, made or led:

- Protobufs v2
- Cap'n Proto
- Sandstorm.io
- Cloudflare Workers

Warning - I am **not**:

- An experienced speaker
- A graphics designer

Efficiency...

App Code Footprint
VM: 10GB
Container: 100MB
Needed: < 1MB

Baseline Memory Usage
VM: 1GB
Container: 100MB
Needed: < 5MB

Context Switching
VM: low
Container: medium
Needed: extreme

Startup Time
VM: 10s
Container: 500ms
Needed: < 5ms

Other use cases

APIs
Run client code directly on API server.

Big Data Processing
Run code where the data lives.

Web Browsers
Run code from visited sites.

WAIT, HOLD UP

We built this already!

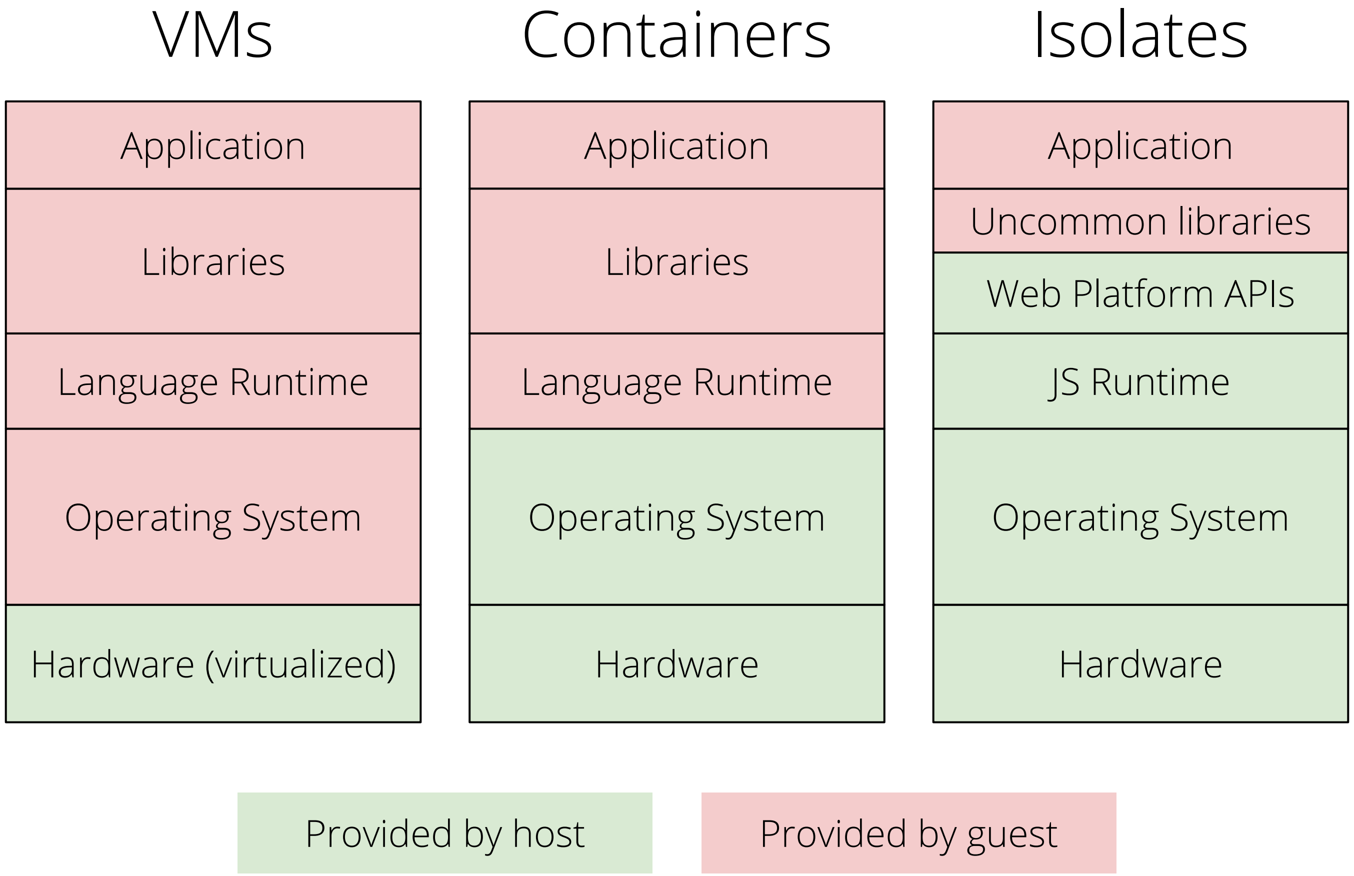
Browsers are optimized for...

- Small downloads
- Fast startup
- Many tabs and frames
- Secure Isolation

V8 JavaScript Runtime:
An Extreme Multitenancy Engine

Isolates and APIs

```
class v8::Isolate
```


VMs
Application
Libraries
Language Runtime
Operating System
Hardware (virtualized)
Containers
Application
Libraries
Language Runtime
Operating System
Hardware
Isolates
Application
Uncommon libraries
Web Platform APIs
JS Runtime
Operating System
Hardware
Provided by host
Provided by guest

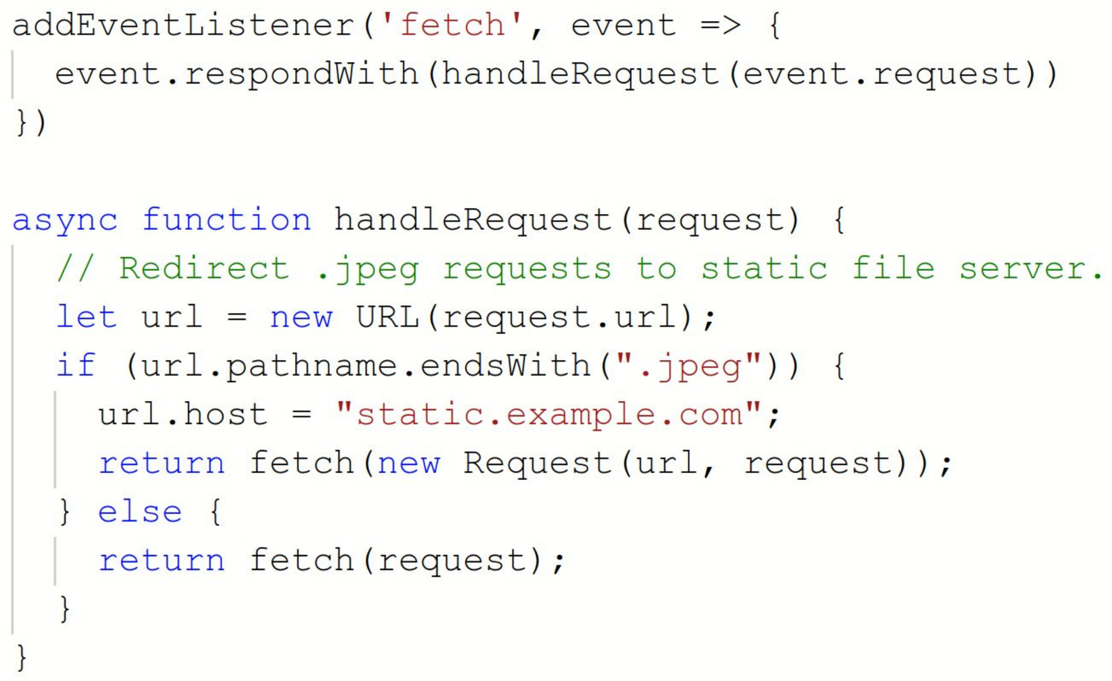
Standard APIs

HTTP client:

Fetch API

HTTP server:

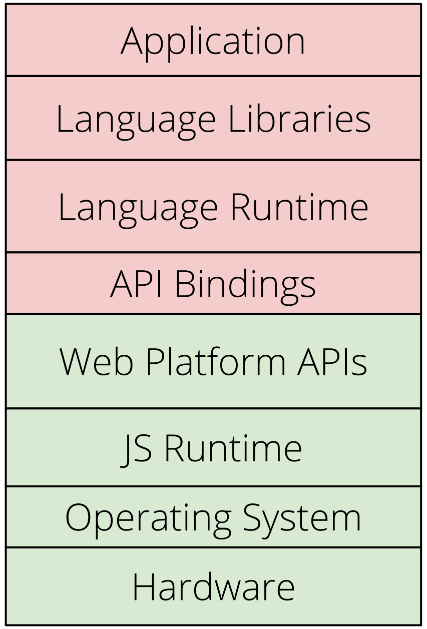
Service Workers



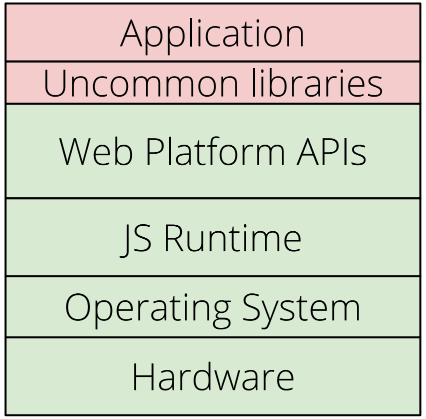
```
addEventListener('fetch', event => {
  event.respondWith(handleRequest(event.request))
})

async function handleRequest(request) {
  // Redirect .jpeg requests to static file server.
  let url = new URL(request.url);
  if (url.pathname.endsWith(".jpeg")) {
    url.host = "static.example.com";
    return fetch(new Request(url, request));
  } else {
    return fetch(request);
  }
}
```

WebAssembly?

WASM



Isolates

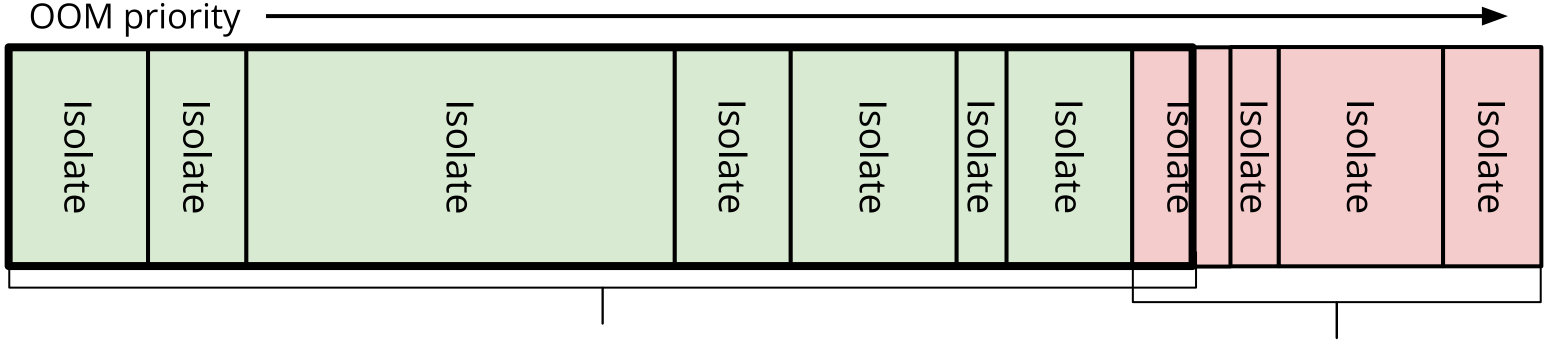


Missing a way to share common runtimes...

Resource Management

OOM Killing

as a First Resort



Desired total memory usage.

Evict these.

Prioritize: LRU, high memory usage

Resource limits

CPU

Isolates run on separate threads.

`timer_create(CLOCK_THREAD_CPUTIME_ID)`

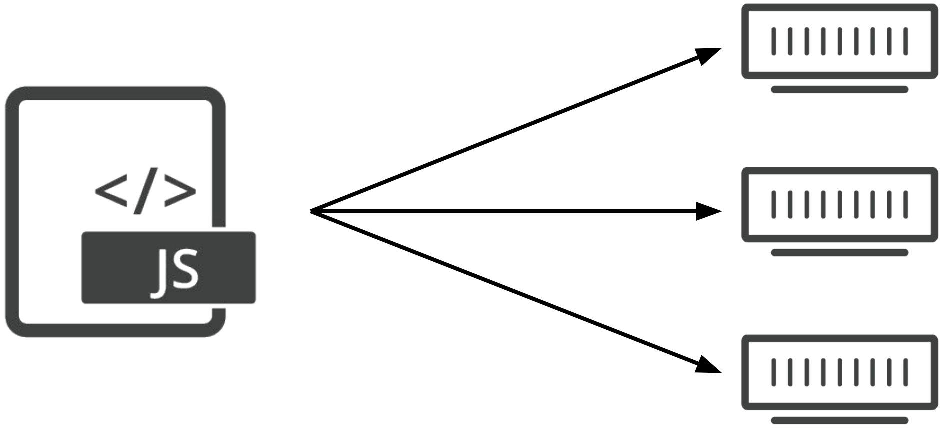
`isolate.TerminateExecution()`

RAM

Monitor with `isolate.GetHeapStatistics()`

Evict isolates that go over limit.

Code Distribution

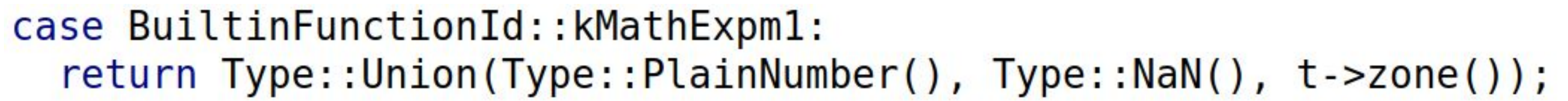


Security

Is V8 secure enough for servers?

V8 bugs...

Deep in `v8/src/compiler/typer.cc...`



```
case BuiltinFunctionId::kMathExpm1:
  return Type::Union(Type::PlainNumber(), Type::NaN(), t->zone());
```

Optimizer: "`Math.expm1()` can return real number or NaN."

Forgot: -0 (negative zero)

Full sandbox breakout!

Awesome writeup: Google "Andrea Biondo V8 bug"

Link: https://abiondo.me/2019/01/02/exploiting-math-expm1-v8/

NOTHING IS "SECURE"

Security is **Risk Management**

Relatively more bugs than VMs.

Reasons:

- Larger attack surface (Bad)
- More research (Good)
  - Bug Bounty
  - Fuzzing
  - Important target

Risk Management

Browser

VS

Server

Risk Management

VS

Browser

Server

Install updates fast.

Risk Management

Browser

VS

Server

Install updates fast.

Install updates faster.

Risk Management

Browser

Install updates fast.

Use separate profiles for
trusted vs "suspicious" sites.

VS

Server

Install updates faster.

Risk Management

VS

Browser

Install updates fast.

Use separate profiles for trusted vs "suspicious" sites.

Server

Install updates faster.

Use separate processes for trusted vs. "suspicious" tenants.

Risk Management

VS

Server

Browser

Risk Management

VS

Server

Browser

Store all scripts ever uploaded
for forensic purposes. No eval().

Risk Management

VS

Server

Browser

Store all scripts ever uploaded
for forensic purposes. No eval().

Watch for segfaults, inspect
scripts that cause them.

Risk Management

VS

Server

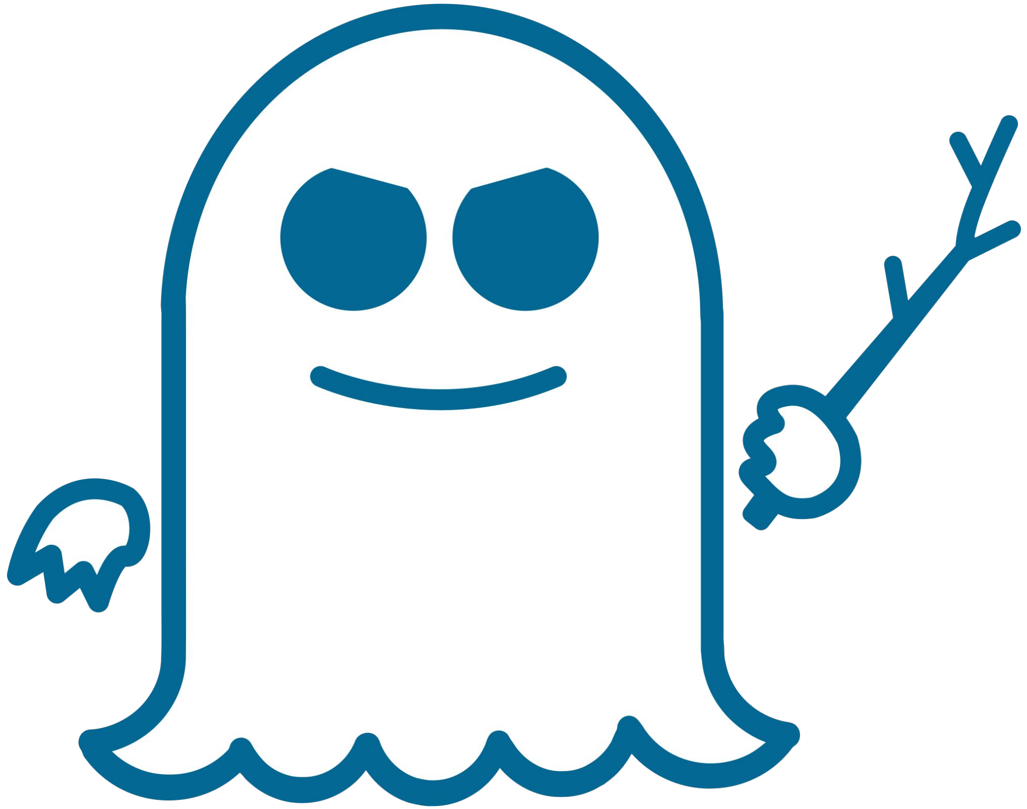
Store all scripts ever uploaded for forensic purposes. No eval().

Watch for segfaults, inspect scripts that cause them.

Browser

... can't, privacy violation.

What about Spectre?



Spectre is here to stay
An analysis of side-channels and speculative execution

Ross Mcilroy
Google
rcmilroy@google.com

Jaroslav Sevcik
Google
jarin@google.com

Tobias Tebbi
Google
tebbi@google.com

Ben L. Titzer
Google
titzer@google.com

Toon Verwaest
Google
verwaest@google.com

February 15, 2019

discovered that untrusted code can construct a universal read gadget to read all memory in the same address space through side-channels. In the face of this reality, we have shifted the security model of the Chrome web browser and V8 to process isolation.

be separated from the state that triggers the optimization, forcing the optimization to repeatedly occur.

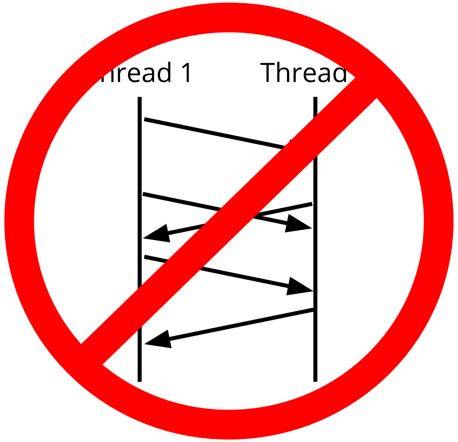
Impossibility of complete mitigation with timers. Based on the generality argument, we argue that mitigating timing channels by manipulating timers is impossible, nonsensical, and in any case ultimately self-defeating. For example, a common thought is that perhaps the μ-architecture can track all time that has been saved due to optimizations and somehow charge the program back. To see why this

We have no solution except process isolation.

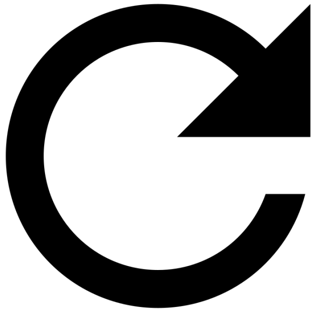
We can neither confirm nor deny that process isolation is enough.

We have tools nobody else has…

No (local) timers (at all!)



No (local) concurrency



Freedom to reschedule

Big Picture

Units of Compute

Granularity

Mainframe

Commodity Server

Virtual Machine

Container

Isolate

Questions?