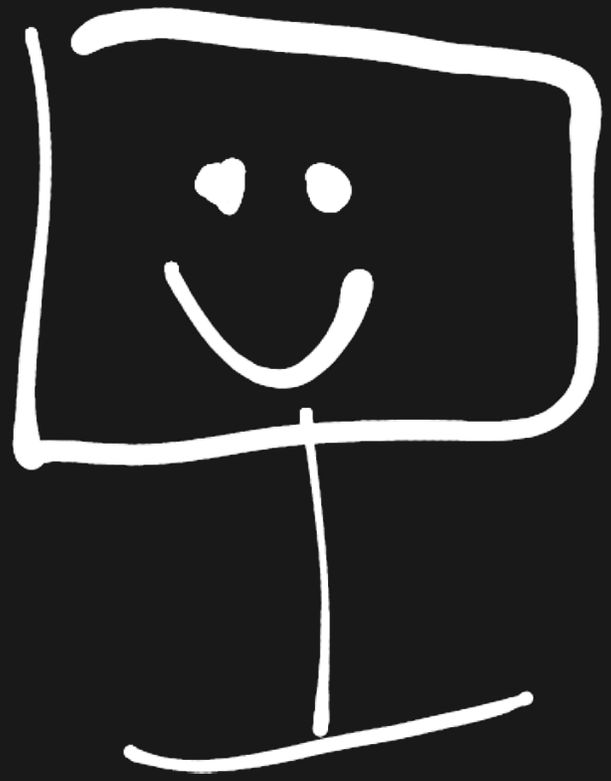
LAVAMOAT

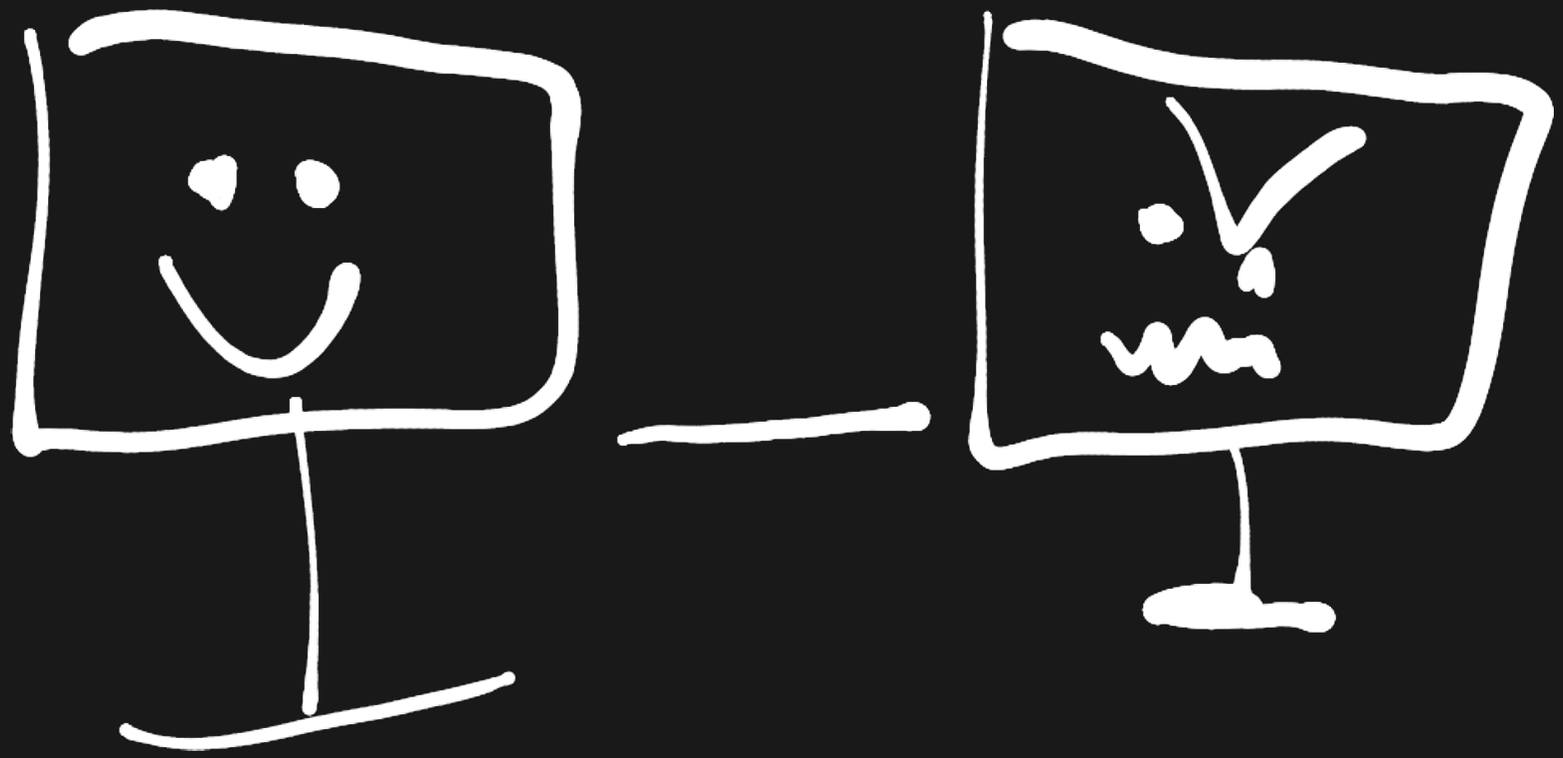
AT SECURE THE WEB FORWARD

@naugtur, 2023

PROBLEM STATEMENT



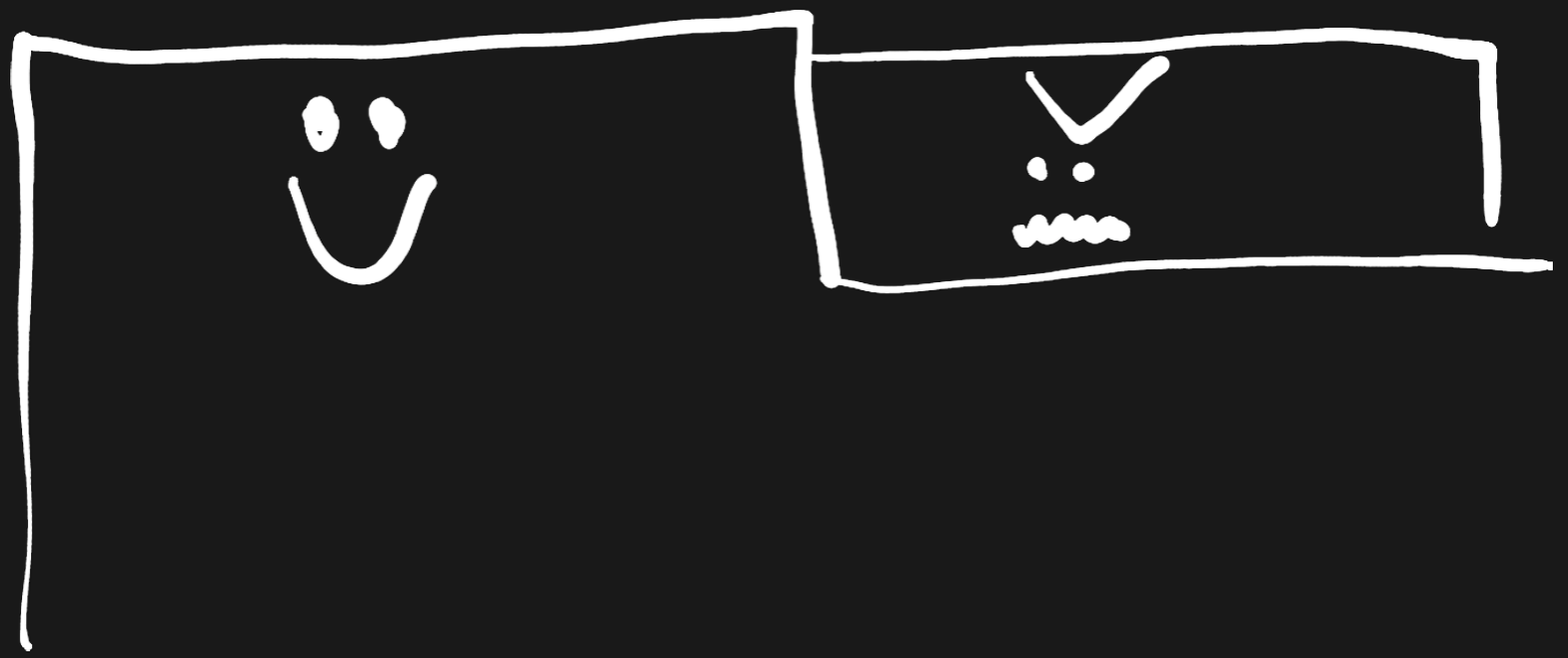
In the beginning, there was software we typed into our computers



Oh no, There's someone in my network and I don't trust them!

WE INVENTED

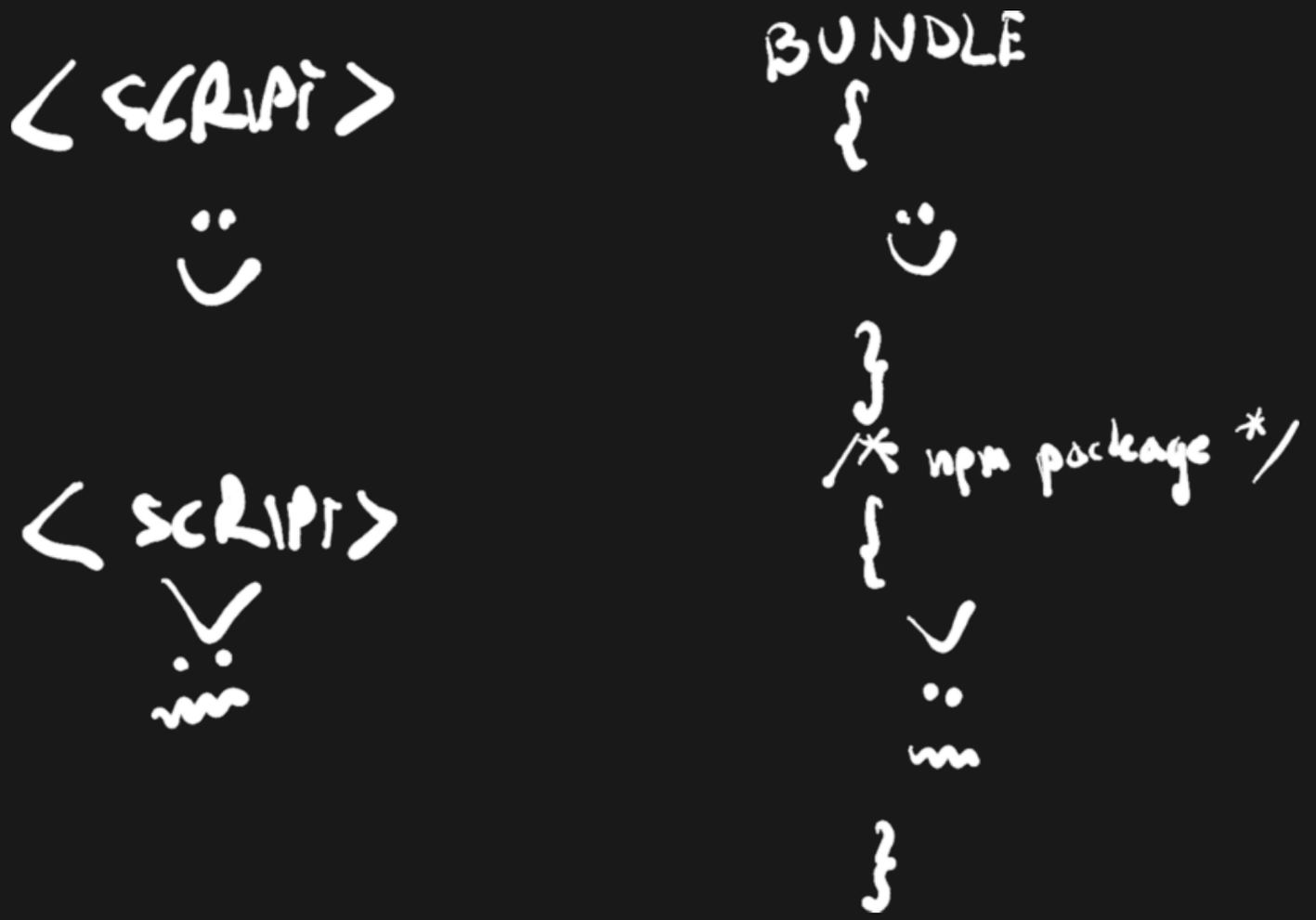
FIREWALLS



Oh no, There's someone in my browser and I don't trust them!

WE INVENTED

SAME ORIGIN POLICY



Oh no, There's someone in my codebase and I don't trust them!

WE INVENTED

HOPING FOR THE BEST

WHAT WE NEED IS

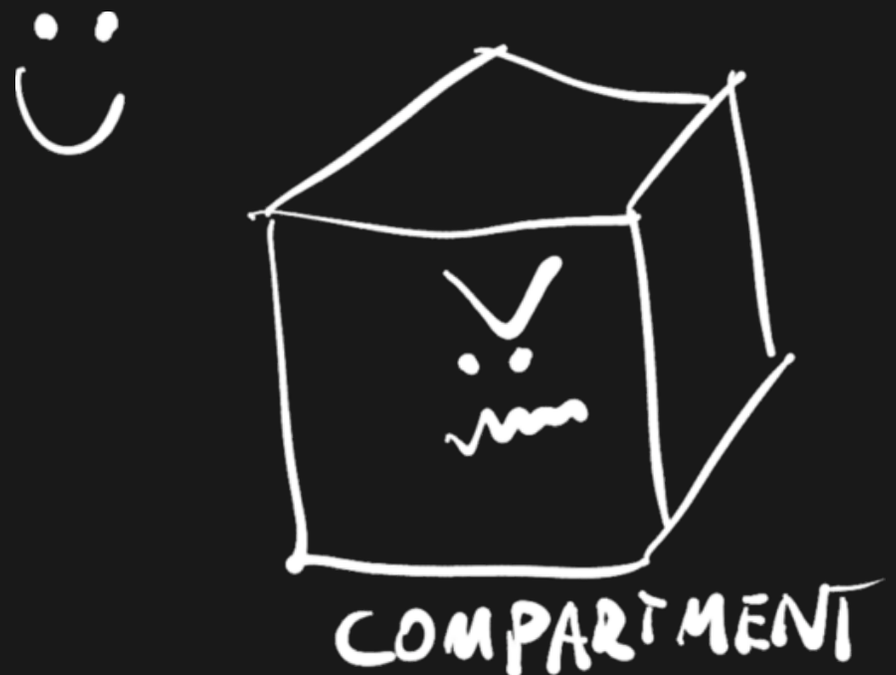
FEARLESS COOPERATION

PROGRESS

- Subresource Integrity
- Content Security Policy
- Trusted Types
- Hardened Javascript

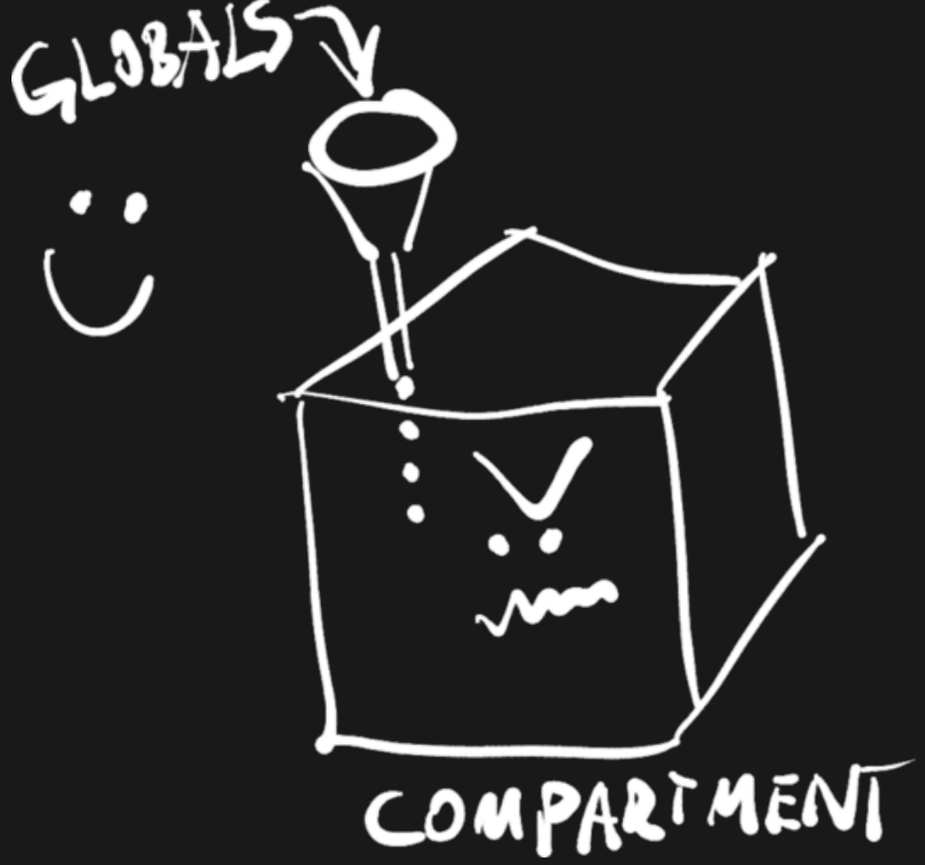
Hardened JavaScript

- Compartment proposal(s) in TC39
- SES Shim
- LavaMoat

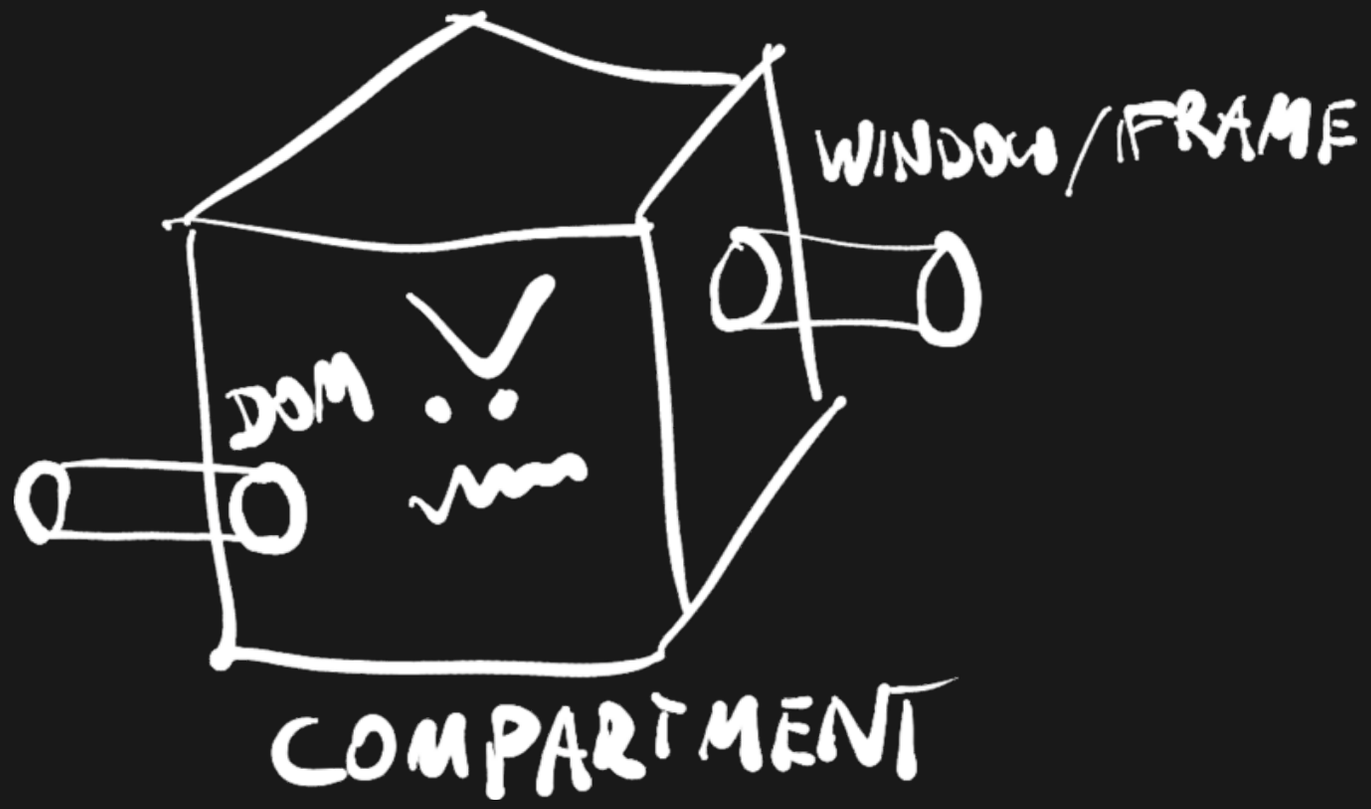

COMPARTMENT

JS design is good for security?

- Take ECMA + W3C
- Add Conway's Law
- Separation between language and APIs
- Power only reachable through scope
- Compartment controls scope



You decide which powers to pass in



So are we done?

Problem statement ++

- Any access to DOM leaks globalThis
- Compartment depends on evaluators or bundling
- `strict-dynamic` but for `eval`?
- same origin realms (tomorrow)

Call to action

How can we support the users of Hardened Javascript in the browser?