Secure Cooperative Sharing of JavaScript, Browser, and Physical Resources

Leo Meyerovich, David Zhu

UC Berkeley

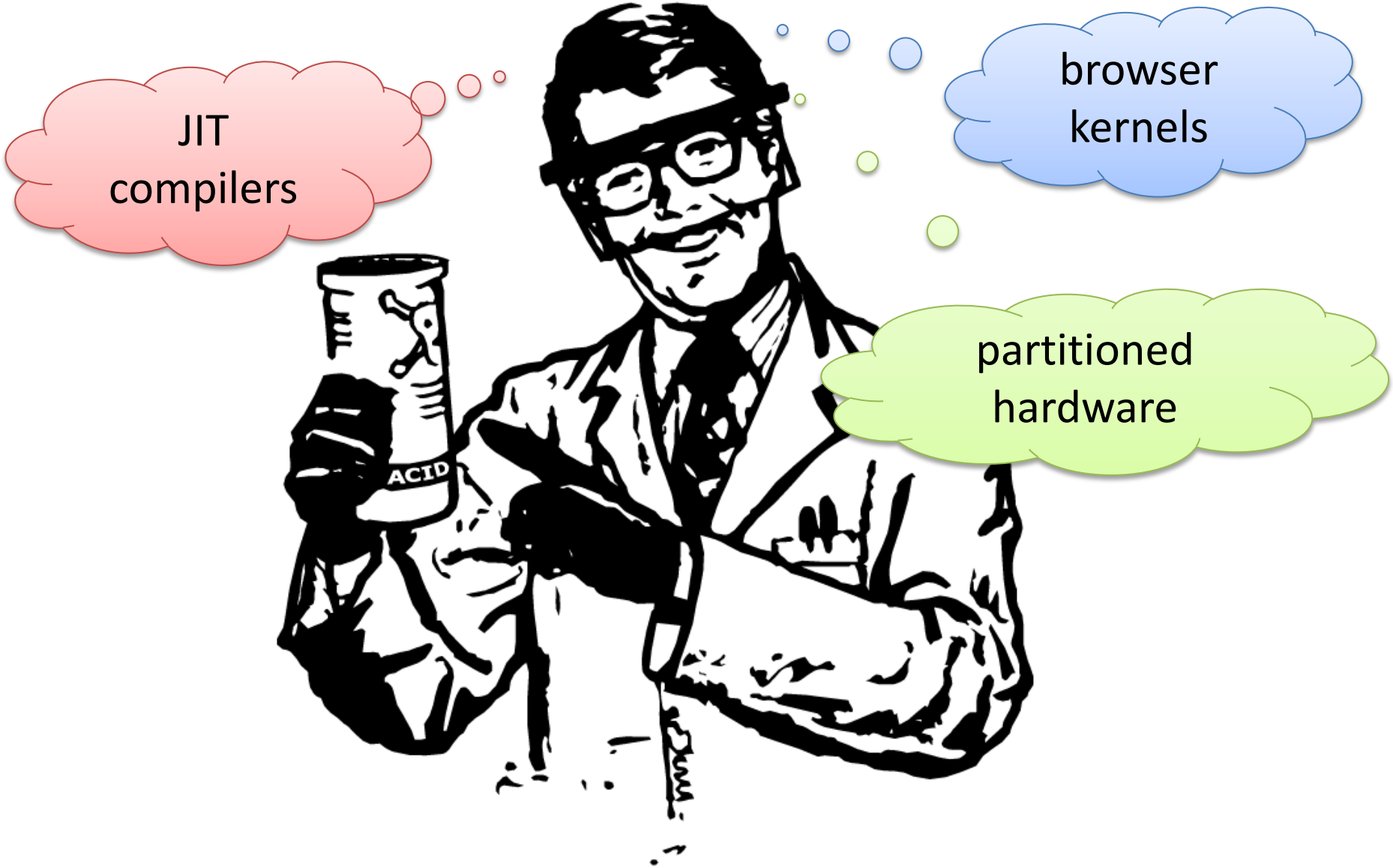
Benjamin Livshits

Microsoft Research

Web Application Security

lipstick on a pig?

Not Your Mother's Browser

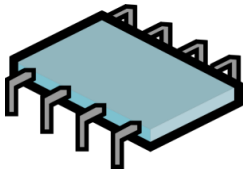
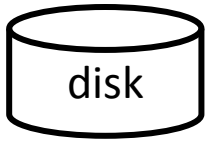


Mashup Manifesto

1. sharing requires control
2. sharing must be natural
3. sharing must be cheap

What to Share?

Hardware

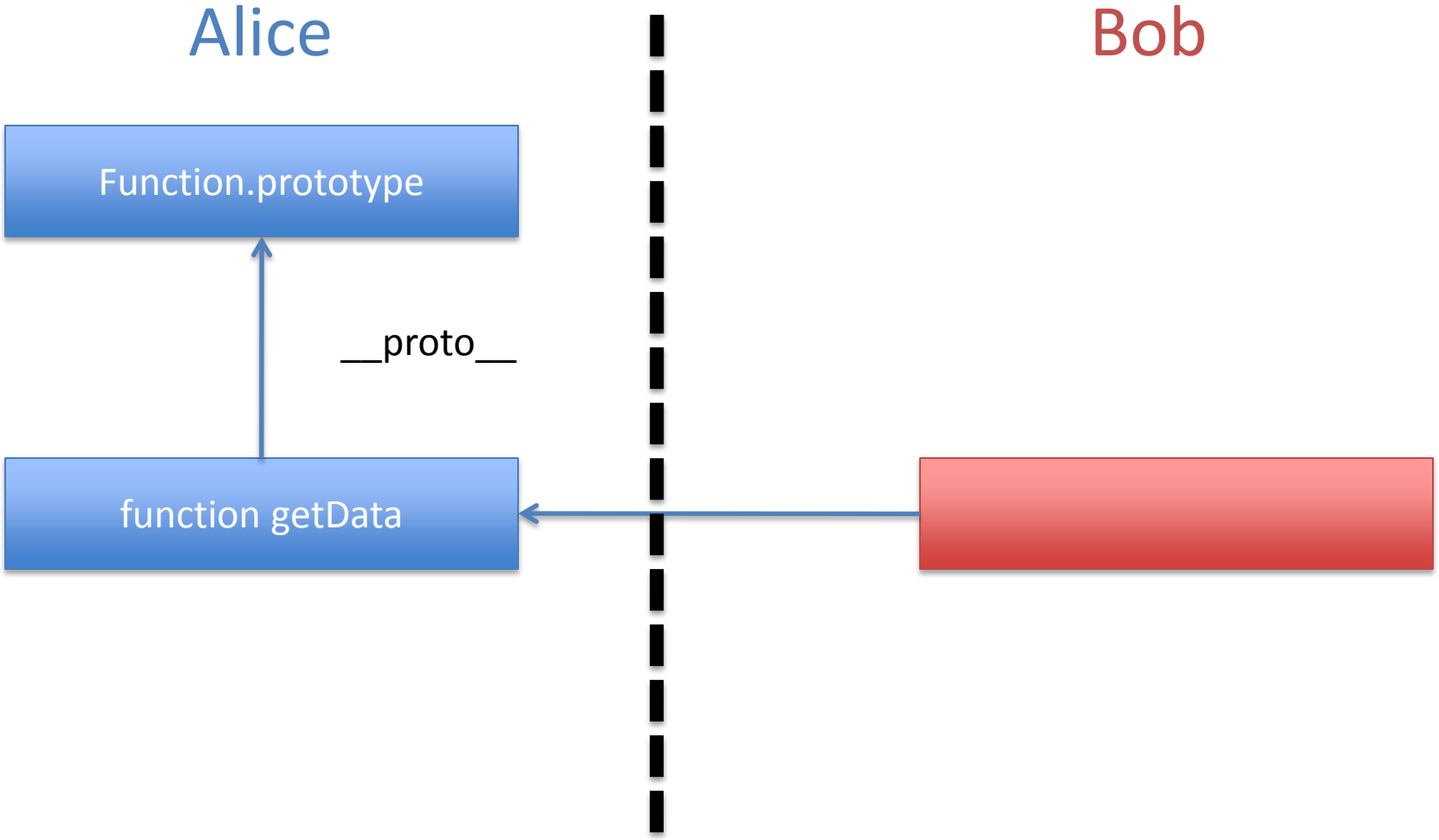

Browser APIs `parser, DOM, network, ...`

JavaScript

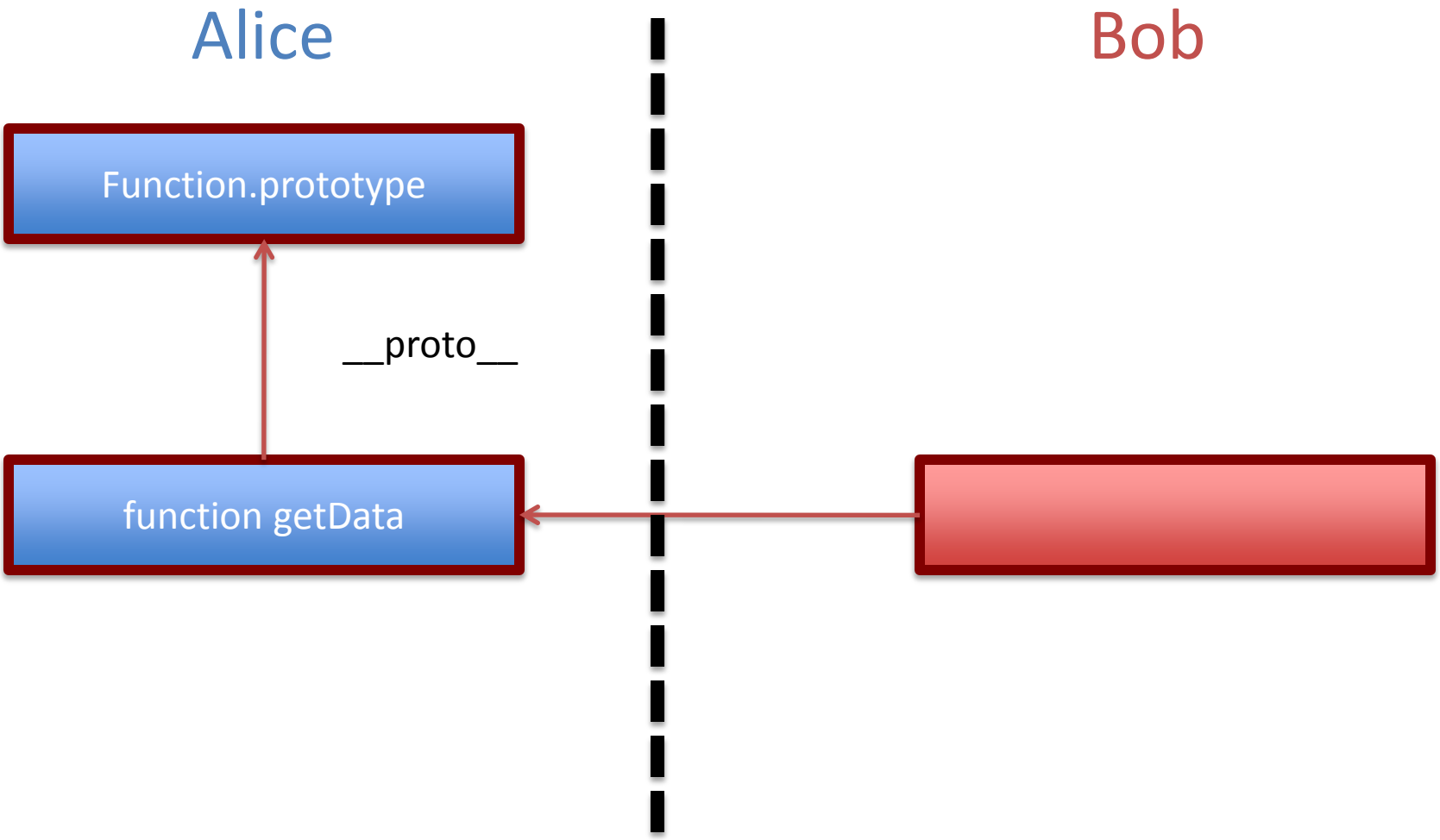


```
1.   <CoFrame src=http://gadget.com/page id=gadget
2.       passthroughBrowser="html css js"
3.      delegatePhysical=".1 cpu"/> ...
4.  var toggle = true;
5.  delegateBrowser(“network”, gadget, "http://gadget.com",
6.      function () { if (toggle) return true; });
7.  function getData() {
8.      toggle = false;
9.      return "profile data"; }
10.  aroundJS(gadget, getData,
11.      function proceed (continue) { return continue(); });
```

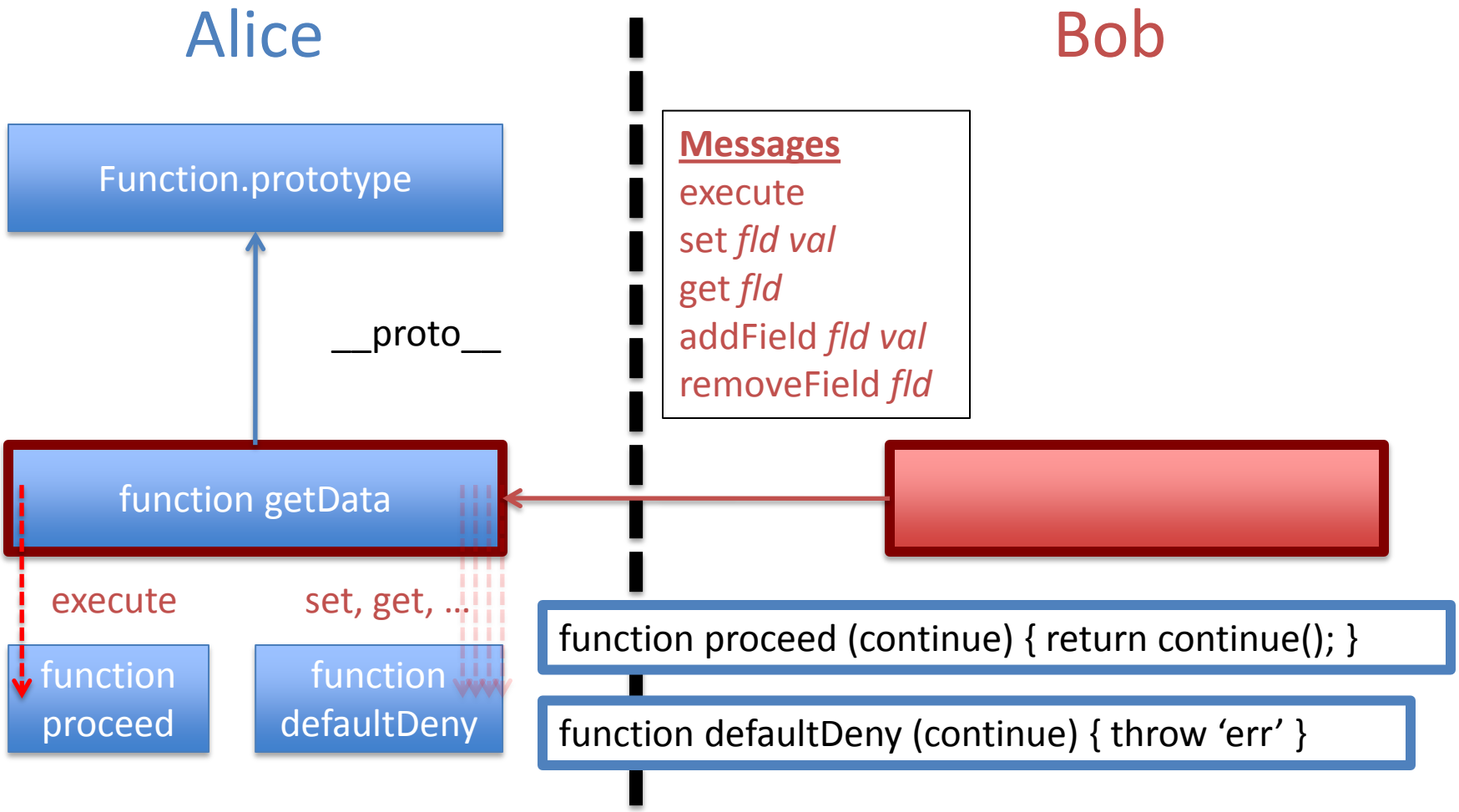
JS Sharing with Cross-Principal Advice



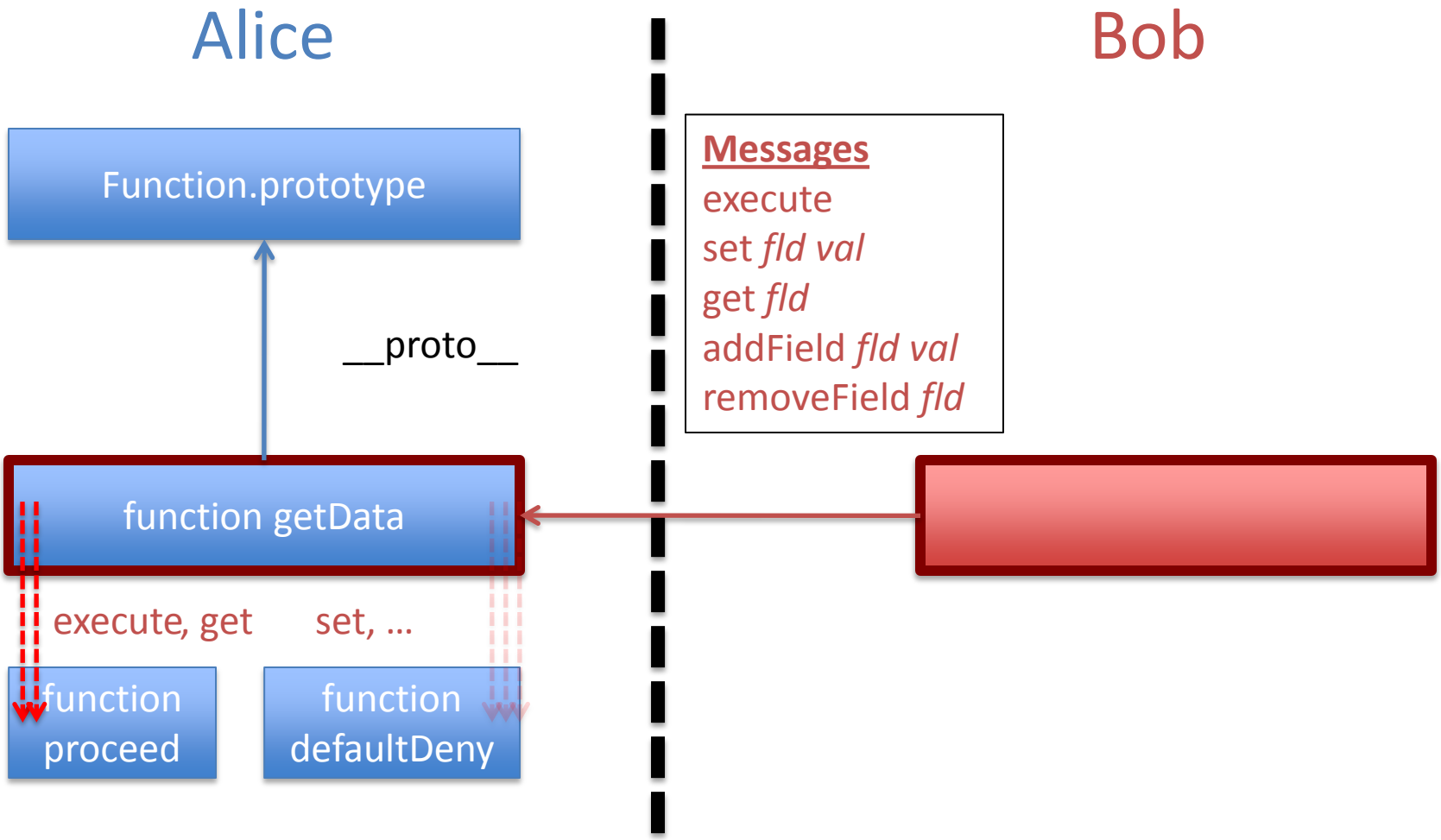
JS Sharing with Cross-Principal Advice



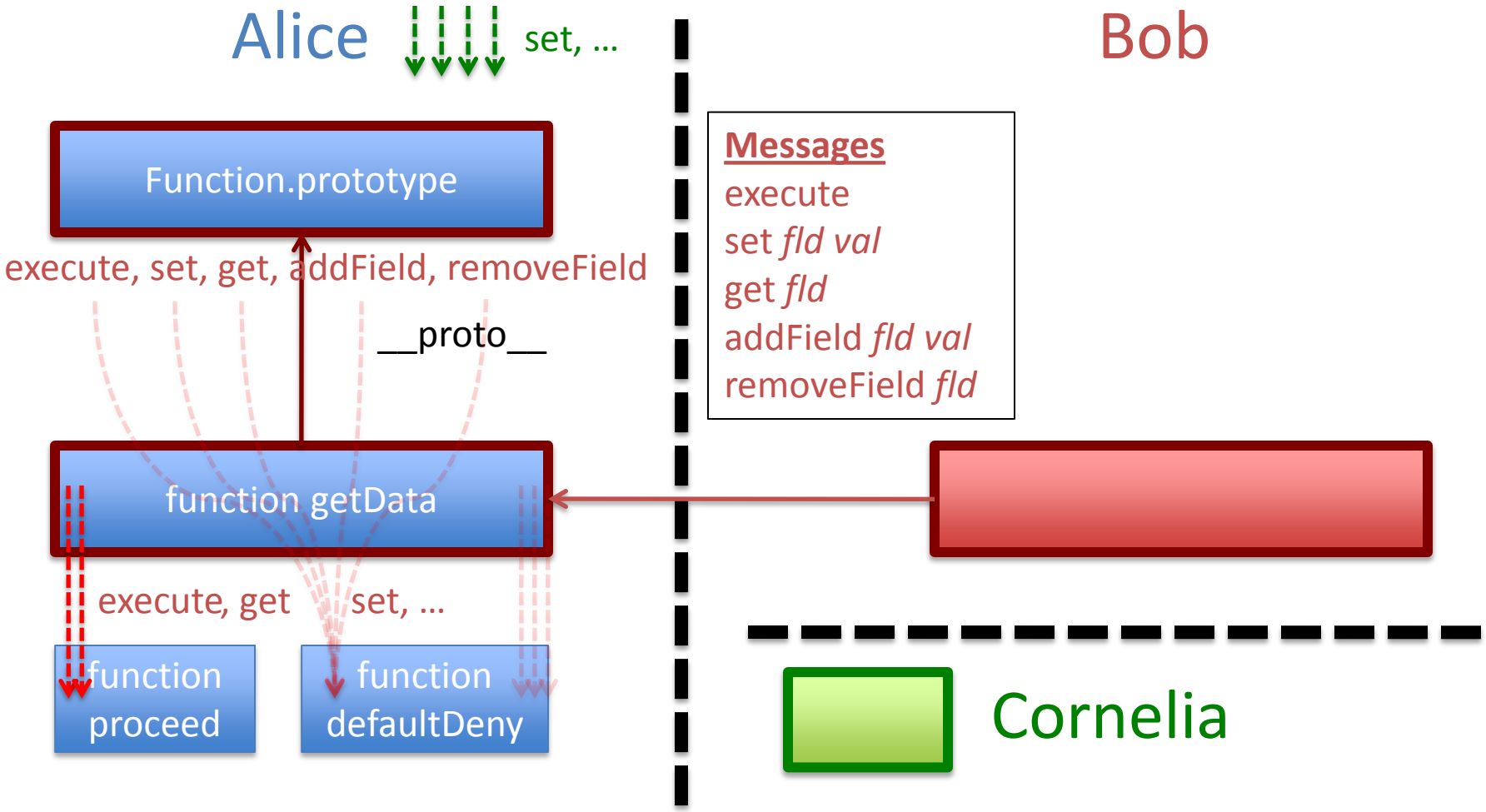
JS Sharing with Cross-Principal Advice



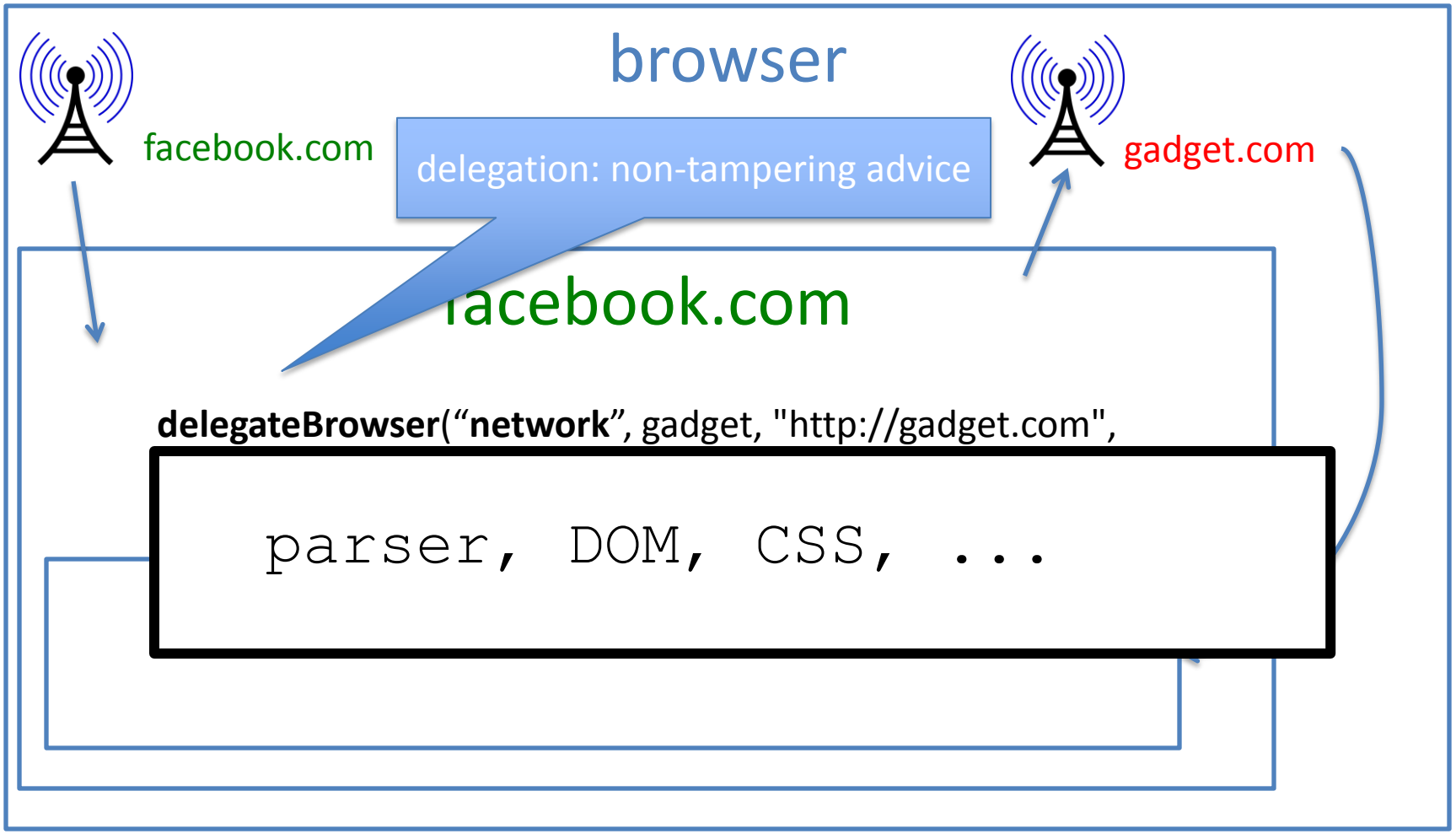
JS Sharing with Cross-Principal Advice



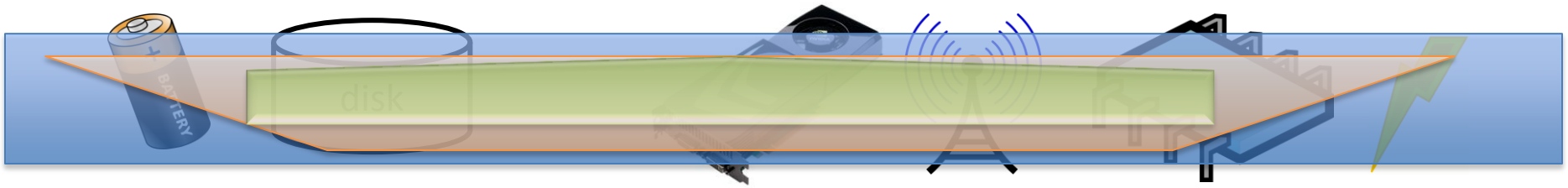
JS Sharing with Cross-Principal Advice



Browser API Sharing with Non-Tampering Advice



Physical Resource Sharing with TessellationOS



render	render	render
layout	layout	layout
...	...	...

facebook

FARMVILLE BETA zynga

double click

Mashup Manifesto

1. sharing requires control
2. sharing must be natural
3. control must be cheap

Related Work

JavaScript Sharing	Caja MashupOS Object Views ConScript
Browser API Sharing	OP Browser ConScript ServiceOS
Physical Resource Sharing	Resource Containers E Gazelle TessellationOS Chrome

backup slides.

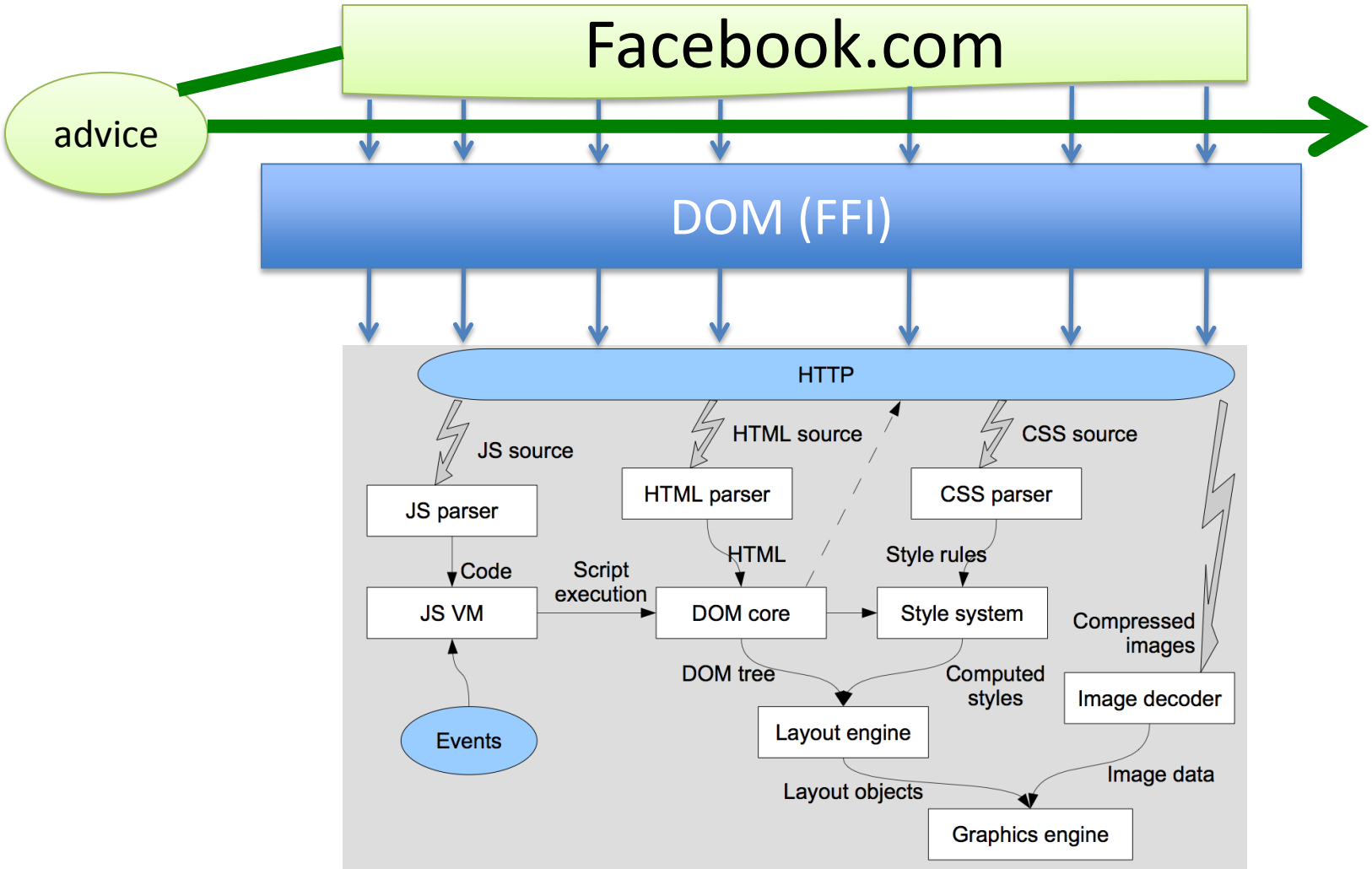
	Gadget Access of Container-Origin Resources						Gadget Access of Gadget-Origin Resources				
	JavaScript		Browser		Physical		JavaScript	Browser		Physical	
Mechanism	d. deny	control	d. deny	control	d. deny	control	untampered	d. deny	control	d. deny	control
frame	✓	string	✓	string	✗	✗	✓	✗	✗	✗	✗
serviceinstance	✓	string	✓	string	✗[f]	✗	✓	✗	✗	✗[f]	✗
nullinstance	✓	string	✓	string	✗[f]	✗	✓	✓	✗	✗[f]	✗
omash	✓	ref	✓	ind. ref	✗	✗	✓	✗	✗	✗	✗
caja$_{\text{same-frame}}$	✓	ref	✓	ind. ref	✗[c]	✗[c]	✗	✓	✗	✗[c]	✗[c]
caja$_{\text{diff-frame}}$	✓	ref	✓	ind. ref	✗	✗	frame	✓	all/none	✗	✗
object views	✓	value	✓	ind. val	✗	✗	✓	✗	✗	✗	✗
conscript	✗[b]	value	✗[b]	value	✗	✗	✗	✗	all/none	✓	✗
coframe$_{\textbf{(ideal)}}$	✓	**value**	✓	**value**	✓	**value**	✓	✓	**value**	✓	**value**

[b] Opt-in (e.g., blacklist). [c] Same-frame JavaScript CPU control in Web Sandbox [f] Gadgets are fairly scheduled with the container, giving excess privilege

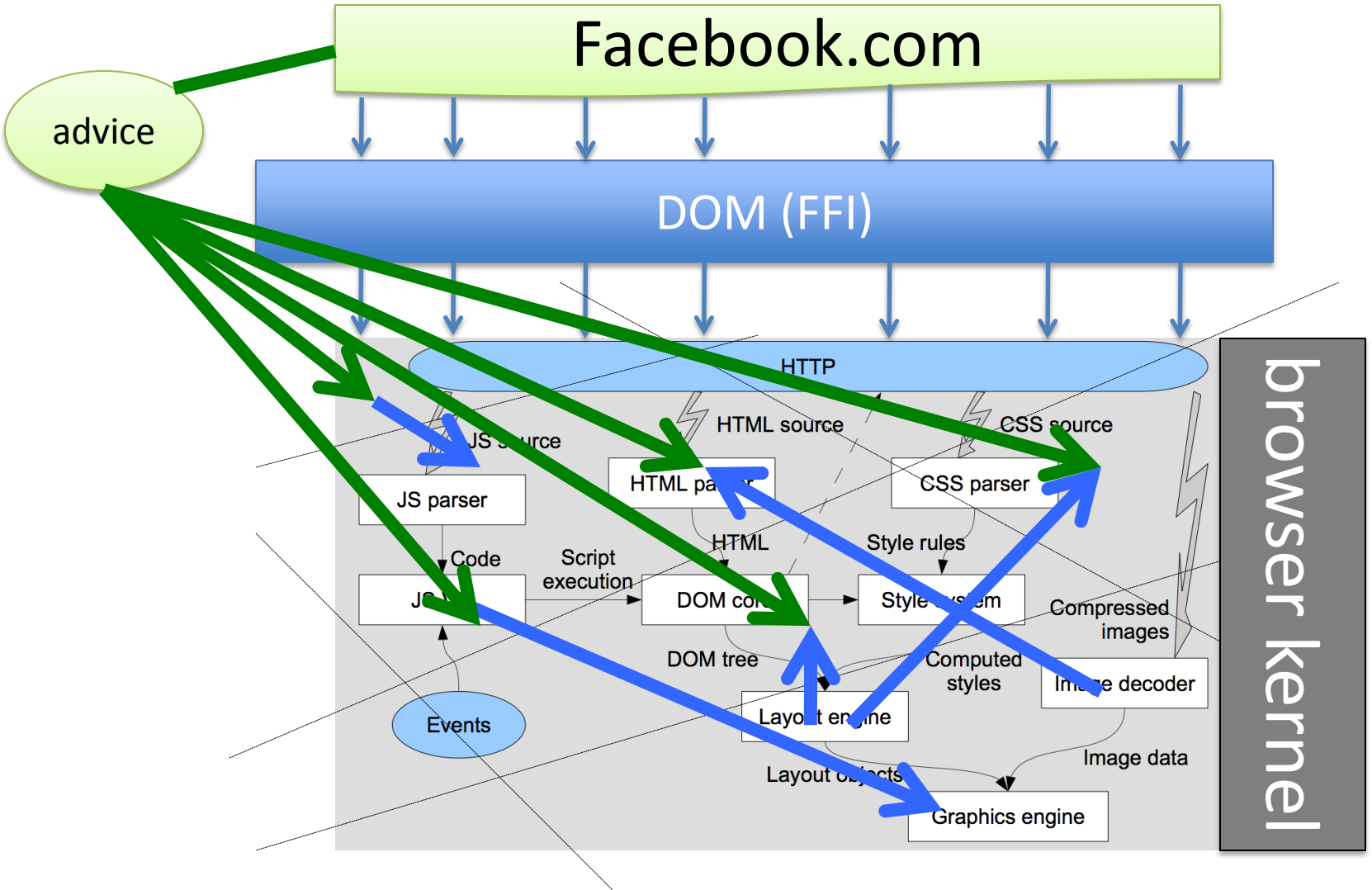
[ref] Sharing a JavaScript value passes a reference graph [val] Sharing a JavaScript value only enables direct access to just that value

[ind] Security-critical functions are not exposed to direct JavaScript control

Sharing Browser APIs: Today

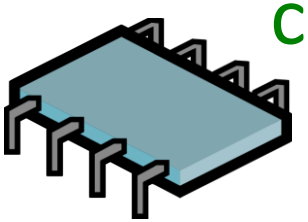


Sharing Browser APIs: Tomorrow

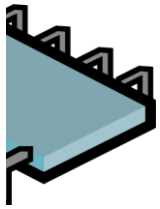


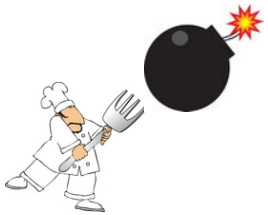
BROWSER

container.com



gadget.com





gadget
fork
bomb!!!

YouTube

YouTube
policy?

BROWSER

container.com

gadget.com

gadget.com

A New Hope