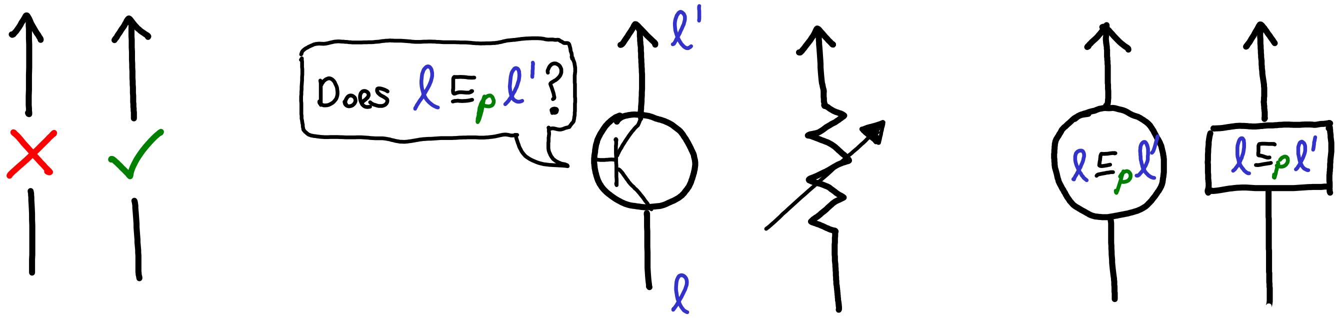
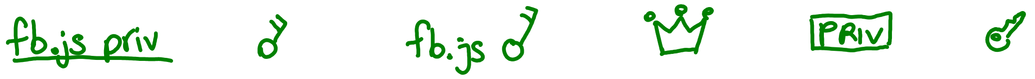

Does $\ell \sqsubseteq_p \ell'$?
ℓ'
ℓ
$\ell \sqsubseteq_p \ell'$
$\ell \sqsubseteq_p \ell'$

Here are some possible symbol choices for arrows to indicate if a flow is allowed or not. Left-most set is the simplest possible iconography, simply indicating if the flow is allowed or not (description must be referred to to figure out why the flow is allowed/not allowed. The other four arrows try to indicate that the arrows are "smart" in some sense; whether or not the flow is allowed is computed automatically based on the labels of the endpoints.



Here are some icons for showing that a compartment has the privilege for some principal. Text-less icons rely on color-coding to demonstrate what privilege they have.

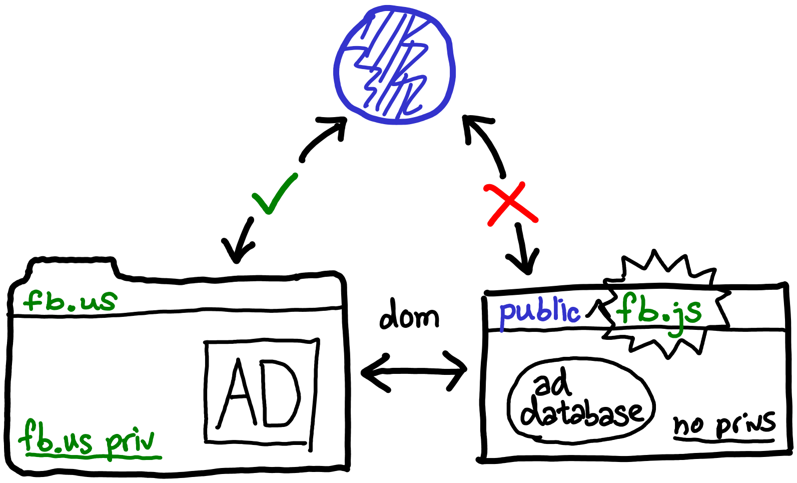


Figure 2. Ad-block extension (steady state)

This diagram shows an ad-block extension operating in steady-state (i.e. blocking ads). Similar to the password checker, prior to performing any ad-blocking, the ad-blocker downloaded a database of ads. Once it is done, it raises its label to the website and then directly manipulates the page's DOM to detect and block ads. When it has access to page data, it does not have access to the Internet (red X). However, the original context fb.us still has access to the Internet. Why? Because this context has the fb.us privilege, giving it the ability to declassify data and send it to the wider world, whereas the ad-blocker has no such privileges. Note that any code the ad-blocker places in the DOM or any images/etc it may try to load execute in the context of the ad-blocker compartment, so by default, there is no way for the ad blocker to trick fb.us into accidentally exercising its privilege. However, a poorly coded fb.us, or a phishing extension, could still exfiltrate data. Nevertheless, this is still an improvement over the carte blanche extensions have today.