Messaging, Origins, etc.

Mike West - TPAC 2025

`postMessage()` is somewhat difficult to hold correctly.

```
window.addEventListener("message", e => {
  if (e.origin.indexOf("trusted.example")) {
    doSensitiveThing(e.data);
  }
});

postMessage({ ... }, "*");
```


window.addEventListen ssage", e => {
if (e.origin.indexOf(d.example")) {
doSensitiveThing(e.data);
}
});
postMessage({ . "*");
trusted.example.evil.com
<iframe sandbox>

https://github.com/mikewest/origin-api

```
window.addEventListener("message", e => {
  const trusted = Origin("https://trusted.example/");
  if (Origin.from(e).isSameSite(trusted)) {
    doSensitiveThing(e.data);
  }
});

postMessage({ ... }, originThatMessagedMeBefore);
```

Forcing point-of-use checks is inherently error prone. We should consider alternatives:

```
window.addEventListener('message-same-site', e => {
  // Safely proceed without doing any
  // `e.origin` checks!
});
```

```
const handler = e => {
  // Safely proceed without doing
  // any `origin` checks!
};
const originAllowList = [
  new URLPattern("https://{*.}?site.example"),
  new URLPattern("https://subdomain.other.site"),
];
window.addFilteredMessageEventListener(
  handler,
  originAllowList);
```