Web Application

Integrity, Consistency & Transparency

Dennis Jackson

moz://a

Motivation

- E2EE Web Apps aren't compatible with the web's threat model today
  - E2EE - Can't trust the server
  - Web Apps - Entirely based on trusting the server

- App Stores provide third party auditability of distributed code. Can we provide similar security properties for the open web?

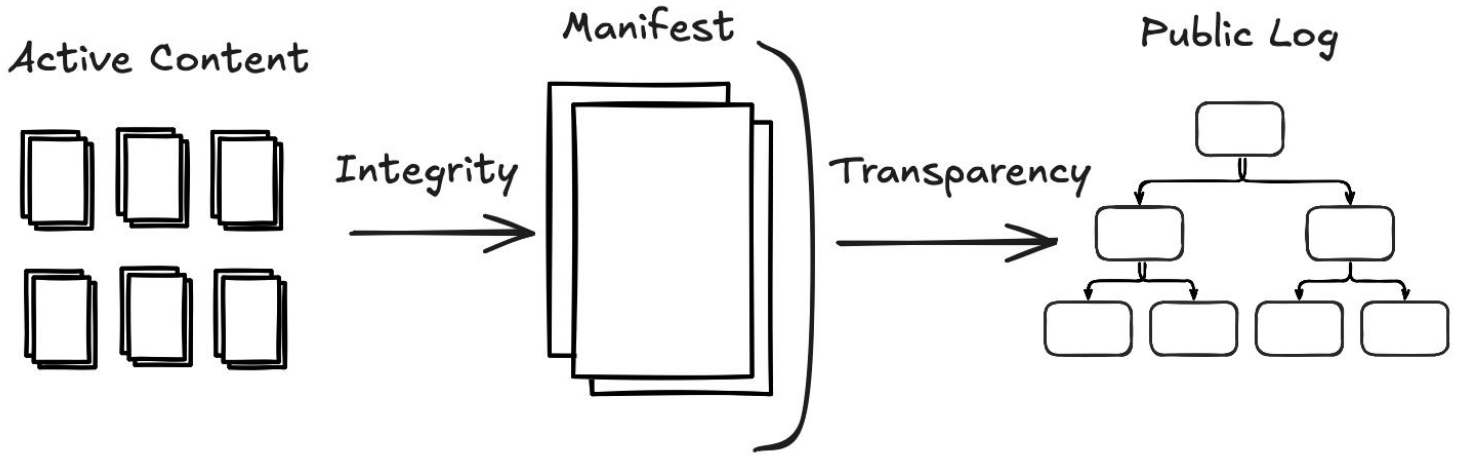
- Explainer

Non-Goals

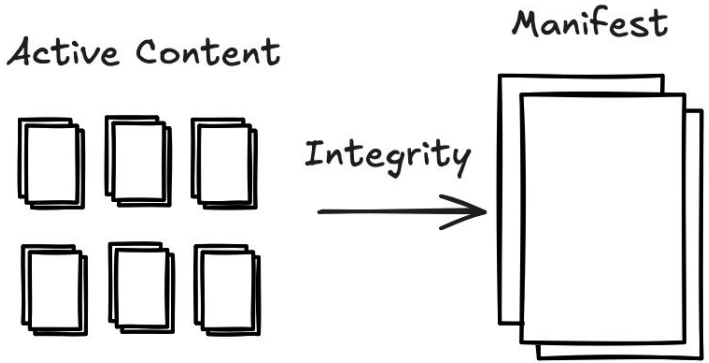
- New packaging model. We want to be ‘webby’.

- Restricting user agency or website autonomy.
  - This is about putting opt-in restrictions on servers, NOT clients.
  - Extensions, ad blockers, alternative user-agents work as they do today.
  - Not to be confused proposals similar in name only (e.g. WEI)

Components

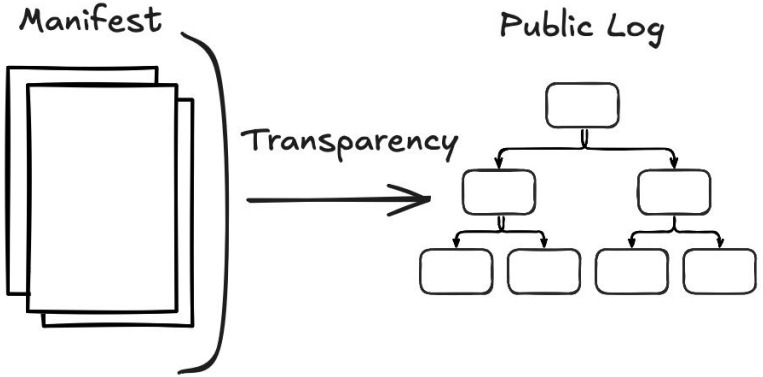

Active Content
Integrity
Manifest
Transparency
Public Log

Integrity



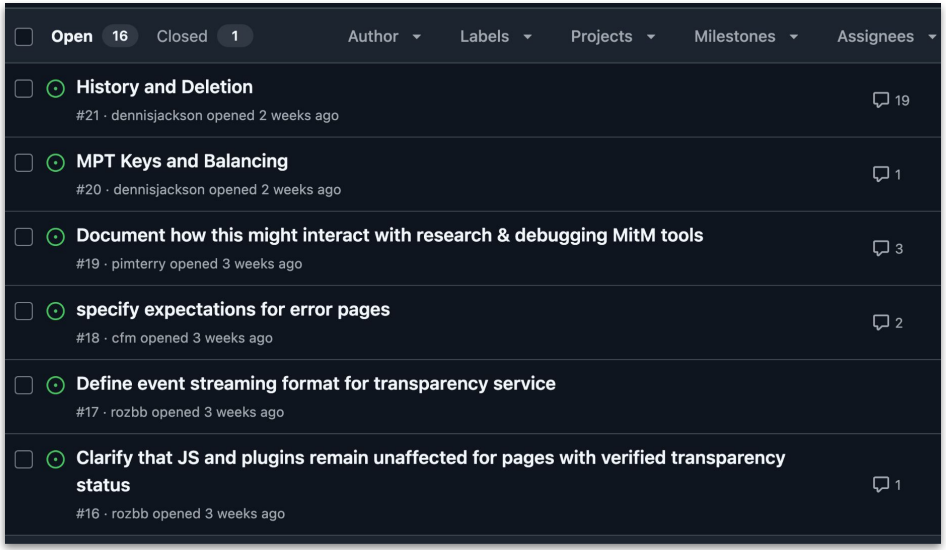
- Building on the Subresource Integrity Spec
  - Introduced Integrity-Policy which can require integrity policies for all resources of a particular type
  - Extending coverage of different resources
  - Moving integrity policy into a public manifest file
- Open Issues

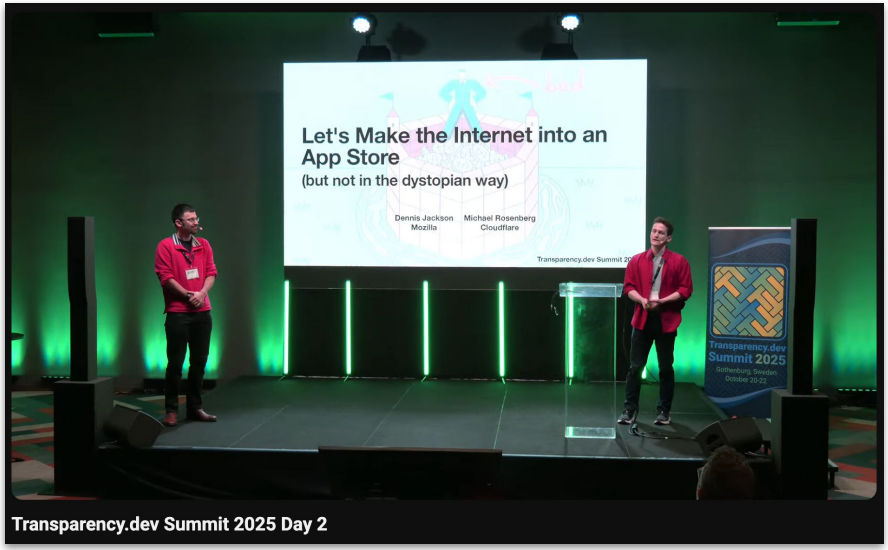
Transparency



- Delivering a proof to clients that third parties have publicly logged the manifests
- Ensuring that clients, websites and security researchers agree on the set of valid manifests

Transparency





Spec

Talk

Enforcement

- Anticipate websites enrolling via a header or preload list (similar to HSTS).
- Failure to deliver integrity manifest and transparency proof on future visit will lead to errors
- Ability to unenroll quickly via special signal delivered with transparency proof.

Next Steps

- Continuing to iterate
- Many folks from different orgs involved and contributing
- Thinking about a home for the non-SRI spec work