

IETF

Embedded Web Package Use Case

Embedded

Web technologies are increasingly used
for embedded systems.

Already hundreds of millions of devices.

It's a big commons.

**It's currently a little
'other'.**

file://

The most straightforward use cases just
use the file protocol

https://localhost

Some use cases also use a server,
basically loopback

But the majority of devices are severely resource constrained by comparison...

Ultimately your embedded device comes with an application (at least one).

That application *can't* depend on the network on first load

But it almost always will want to update
when the network is available.

I'd like to ship you an e-reader with some
books, or a cookbook with some recipes -
etc. These exist as web resources, and
over time content, design, etc will change.

The Web has always assumed your first interaction with a site would be after fetching it from a domain.

Many features are designed around this, creating some interesting challenges.

At the same time, the web has greatly expanded its capabilities, including offline.

There is simply a single significant disconnect: The assumption that first content always comes from online.

How can I bridge the two worlds?

Today, embedded uses wind up solving/resolving much complexity to bridge this gap.

What we really want is to simply bootstrap a service worker with its offline content, and then let all of the Web technologies work fluidly as they do elsewhere..

This use case does not require a model of universal trust and signing - simply a uniform way to configure a specific device or browser startup to trust a package itself.

No local servers necessary
No "is it the web or isn't it?" disparity

It's just a standard offline webapp.

... the ability to provide with my image, a service worker and bundle for foo.com, and have it initialize.