Web Monetization

A new business model for the Web

Adrian Hope-Bailie
@ahopebailie
Agenda

1. The Problem Statement
2. How does Web Monetization work
3. The impact on W3C Web Payments
The Internet Economy is Incomplete

Today: The Web is a **Barter Economy**
Barter is Extremely Inefficient

There may be many people wanting, and many possessing those things wanted; but to allow of an act of barter, there must be a double coincidence, which will rarely happen.

- William S. Jevons, 1875
Money Solves the Barter Problem

Money: Better than Barter Since 5000 BC
Fixed Costs Make Micropayments Unviable
So We Use Hacks to Get Around the Problem

Hacks: Ads, Surveys, Bundling, ...
When an online service is free, you're not the customer.

You're the product.

Tim Cook, CEO Apple
Users Are Opting Out

Mobile Adblock Usage Overtakes Desktop Use
Dominance of Marketplaces
Interledger Solves the Micro-Payments Problem

Streaming Payments using Interledger allows for transfers of tiny amounts
DEMO

Streaming Micropayments for Content
How does it work?

Websites request payment via a **meta tag**
How does it work?

Coil (through the browser extension):
1. Parses the meta tag
2. Resolves the payment destination of the website
3. Opens an Interledger connection to the payee
4. Starts paying

The website is notified of incoming payments via events raised.

For more info on the proposed standard:
https://interledger.org/rfcs/0028-web-monetization/
DEMO

Streaming Payments
browser mockup
How it should work...

A Streaming Payments provider is registered via the Payment Handler APIs

Payments are requested by the website using Payment Request API

Website is able to request a non-interactive flow (fail silently)

User is aware that payments are streaming
Concrete Proposals

1. **stream**
   New short-string payment method for requesting a streaming payments connection

2. **paymentRequest.requestNonInteractive**
   New property of Payment Request that instructs the browser to abandon the flow if there is a need to prompt the user.

3. **paymentResponse.onPaymentReceived**
   New event on the payment response that is raised via a method on the PaymentRequestEvent in the handler.