Signed HTTP Requests

Provisioning a key-pair in the browser for secure HTTP requests that don’t require user interaction

Background

Web Monetization (*webmonetization.org*) is a WICG incubated project

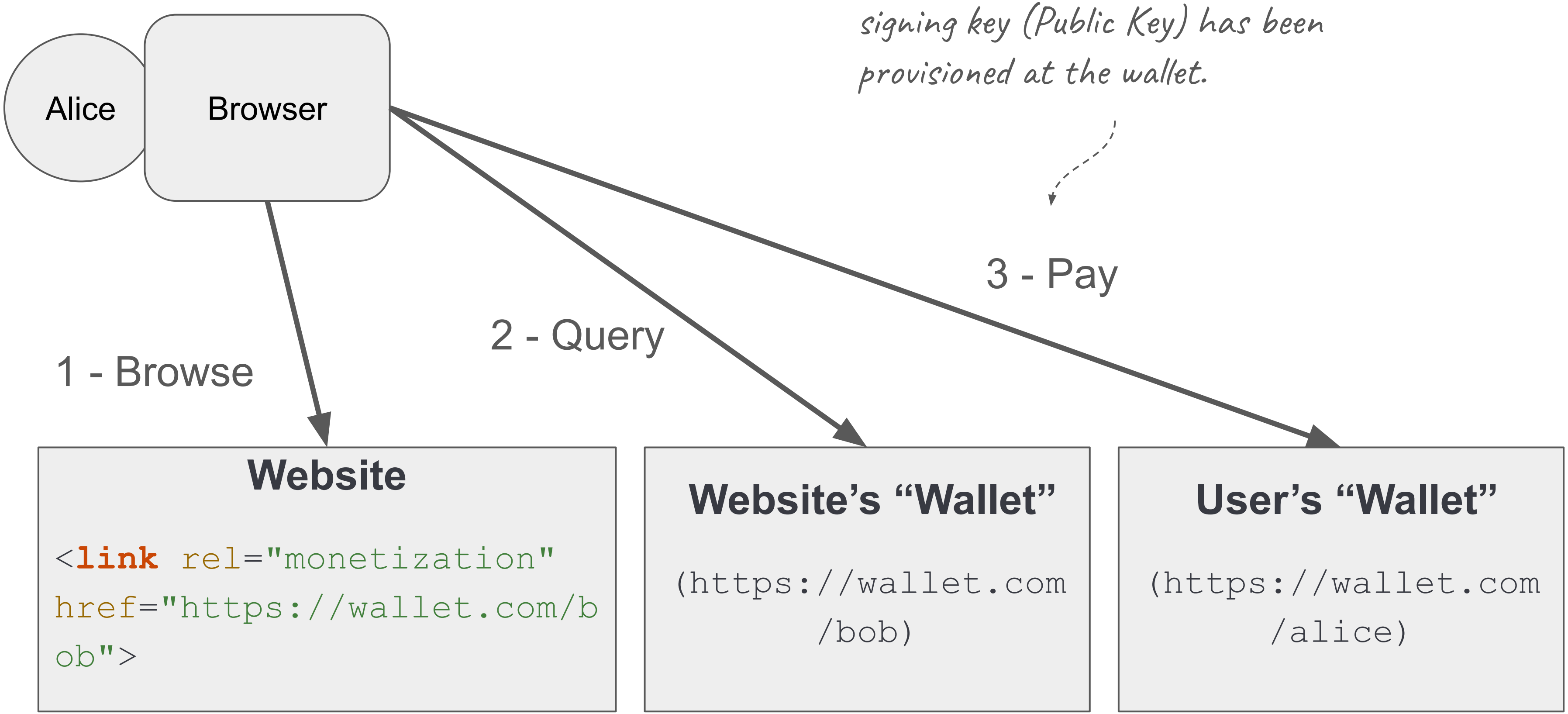
Declarative API for receiving payments at a website

Use case is **very low value (micropayments)**, **no user interaction** required

```
<link rel="monetization" href="https://example.com/pay">
```

Flow starts here

Payment flow



Alice

Browser

1 - Browse

2 - Query

3 - Pay

This is a signed HTTP request. The signing key (Public Key) has been provisioned at the wallet.

Website

```
<link rel="monetization" href="https://wallet.com/bob">
```

Website's "Wallet"

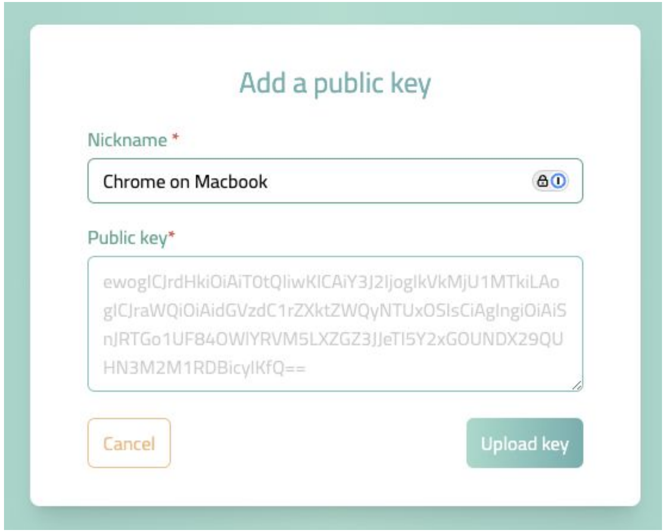
(https://wallet.com/bob)

User's "Wallet"

(https://wallet.com/alice)

Current Provisioning and Transaction Flow

1. Browser generates a key pair.
2. User loads the public key into their wallet.
3. Wallet associates key with user account.
4. Browser signs requests to wallet APIs using private key.
5. Wallet authenticates browser (client) by validating request signature.
6. Wallet authorizes browser as first-party acting on-behalf-of user by matching public key with key on record for user's account



Add a public key

Nickname *

Chrome on Macbook

Public key*

ewogICJrdHkiOiAiT0tQIiwKICAiY3J2IjogIkVkMjU1MTkiLAo
gICJraWQiOiAidGVzdC1rZXktZWQyNTUxOSIsCiAgIngiOiAiS
nJRTGo1UF84OWlYRVM5LXZGZ3JJeTI5Y2xGOUNDX29QU
HN3M2M1RDBicyIKfQ==

Use Case Requirements

1. User provisions a public key into the wallet (RP)
   - Need strong authentication of user at this time
   - Key pair generated by client (browser) and only public key is shared with wallet
   - Private key is stored securely (hardware security preferred)

(This is a very familiar set of primitives… WebAuthn?)

2. Browser uses private key to sign API requests **without user interaction**
3. Private key is stored securely and can’t be exfiltrated

Proposal #1

User logs into wallet via browser.

Wallet invokes browser API to generate key pair and extract public key.

WebAuthn/Passkeys and WebCrypto have the primitives.

Issues:

1. WebCrypto APIs have no mechanism for indicating to the browser that the key pair is for Web Monetization or API requests to a specific wallet API (i.e. specifying the RP)
2. WebAuthn/Passkeys doesn’t allow signing in future without user interaction

Proposal #2

Use hardware protected keys to exchange session keys.

When user starts browsing, user interaction is required to “unlock” their wallet for passive payments (without user interaction).

“Unlock” flow uses hardware protected keys to sign session key sent to wallet.

Issues:

1. Can WebAuthn/Passkeys be used to sign ad-hoc data (e.g. a session public key) as part of the login flow?

Comments or Questions

adrian@fynbos.dev

https://webmonetization.org

https://rafiki.money (demo wallet)