Web Payments Architecture

Next steps for the Web Payments WG
Disclaimer

This does not represent consensus of the WG

What follows is a conceptual architecture for the payment APIs based on feedback, discussion and experience of the WG
Goals

- **Lower Friction**
  
  *fewer clicks, swipes, taps, no typing*

- **High Security**
  
  *cryptographic certainty, risk-based policies, two-factor authN*

- **Protect Privacy**
  
  *only share data as required, always with consent*
Immovables

- Origin Security Policy
- Risk is evaluated based on data
- Data collection is bad for privacy
- Payment initiation often done by 3rd-party from 3rd-party context (iframe)
- Browsers limiting 3rd-party access to data/cookies to prevent tracking
4 party model

Acquirer

Network

Issuer

Merchant

User
4 party model (generic)
Interoperability

3 domains

PSP
Merchant
Network
RP
User
3 domains

- PSP
- Merchant
- Browser
- Network
- RP
- User
Old Thesis - Interoperability

PSP

Network

Merchant

Browser

RP

User
New Thesis

Provide More Options

Break a payment into stand-alone functions

Provide primitives to cater for each function:

1. Instrument/Payment Method Selection
2. Authentication of User and Payment Details
3. Authorization of Payment
Instrument/Payment Method Selection

1. User decides how to pay
2. RP is selected implicitly
3. RP is invoked
Instrument/Payment Method Selection

EXAMPLES
- Type credit card number into a form
- Click/tap digital wallet button
- Select stored credential
Instrument/Payment Method Selection via PH

Use Payment Handler to invoke RP
Instrument/Payment Method Selection via Merchant

PSP

Network

Use payment network to invoke RP

Merchant

Browser

User

RP
Discussion

- Could we add an event to the Payment Request to reveal selection to merchant if no Payment Handler associated?
- What is the best UI: Payment sheet, minimal UI, other?
- Select what?
  - Payment Method
  - Payment Handler
  - Payment Instrument
  - Payment Credential
Authenticate via Payment Handler

- openWindow and own UI
- Secure Payment Confirmation
Authenticate via Merchant & Network

- Secure Payment Confirmation
  - Embed RP iframe
  - Redirect to RP
Authenticate via Merchant & Network

- Secure Payment Confirmation
  - Embed RP iframe
  - Redirect to RP
  - openWindow
Discussion

- Is it safe to allow the merchant to call `openWindow` on the Payment Request AFTER the user has selected an instrument/method?

- What if we limit the origin of the modal window to be “same origin” as the selected
Authorize via Payment Handler

PaymentRequestResponse contains details of completed payment
Authorize via Merchant & Network

PaymentRequestResponse contains data to authorize payment via network
Some Concepts
Payment Context

1. Merchant has called Payment Request API
2. User has been shown browser rendered payment UI

Browser can expose powerful features and reveal select user data within this context (even to the merchant origin):

- Enable Modal Window
- Secure Payment Confirmation allows non-RP WebAuthn invocation
- Selected payment instrument/method revealed to merchant

Only inside Payment Handler event OR Payment Request event
Browser Primitives exposed in Payment Context

- Modal Window:
  
  *In-context display of cross-origin UI*

- Secure Payment Confirmation:
  
  *Streamlined secure authentication using WebAuthn + PaymentRequest data*

- Payment Instrument/Method Selection:
  
  *Stored identifiers/logos/labels from RP for re-use and low friction*
Modal Window

Display UI from the RP or PSP origin in context (no redirects or pop-ups)

Superior mobile experience

Renders top-level context (access to cookies and storage)

Privacy concerns mitigated if UI makes it clear that the window is a new context and new origin (UA must show origin and context - “Paying $5 to abc.com”)

Secure Payment Confirmation

Invoke WebAuthn from payment context

Display payment details (amount, payee) in authenticator prompt

Include payment details in signed client data

Allow non-RP origin to invoke and get attestation from inside payment context

RP not required to ship UI for authN
Stored Credential

● A “Payment Cookie” or “Payment Credential”
  ○ icon and label for display in selection lists
  ○ associated with one or more payment methods
  ○ has a unique non-sensitive identifier (URL?) that is shared with merchants and/or Payment Handlers after selection by user
  ○ can be associated with a Payment Handler (or be a Payment Handler)?

● Controlling origin (relying-party) can enumerate, delete and add and link to PublicKeyCredential or Payment Handler from top level context

● Managed independently of cookies and storage in browser settings
Example 1: Selection + 3DS + SPC (with fallback)

User selects a Payment Credential representing a payment card.
Example 1: Selection + 3DS + SPC (with fallback)

Merchant uses 3DS and gets credential list for SPC
Example 1: Selection + 3DS + SPC (with fallback)

RP (ACS) provides credentials to use for SPC
Example 1: Selection + 3DS + SPC (with fallback)

Empty List!

Use `openWindow` to do out-of-band authN and SPC enrollment and/or payment handler installation.
Example 2: Digital Wallet via Payment Handler

User selects a Payment Credential representing a digital wallet (optional step)
Example 2: Digital Wallet via Payment Handler

Credential is associated with PH which is invoked
Example 2: Digital Wallet via Payment Handler

Payment Handler calls `openWindow` to show wallet UI and complete payment
Example 3: SRC via Payment Handler + SPC

User selects a Payment Credential representing a card (linked to a payment handler from a DCF)
Example 3: SRC via Payment Handler + SPC

DCF payment handler is invoked and passed selected credential identifier.
Example 2: Digital Wallet via Payment Handler

PSP

Payment Handler shows no UI.

Invokes SPC

Merchant

Browser

User

RP
Discussion

- Could we add an event to the Payment Request to reveal selection to **merchant** if no Payment Handler associated?
- What is the best UI: Payment sheet, minimal UI, other?
- Is it safe to allow the merchant to call `openWindow` on the Payment Request AFTER the user has selected an instrument/method?
- What if we limit the origin of the modal window to be “same origin” as the selected credential?
- What about `hasEnrolledInstrument` and `skip-the-sheet`?