Secure Payment Confirmation Update

WPWG @ TPAC, Oct 12, 2020

danyao@chromium.org
btidor@stripe.com
adrian@coil.com

Agenda

- Secure Payment Confirmation Overview
- Live Demo
- API at a Glance
- Pilot Roadmap
- Open Discussion

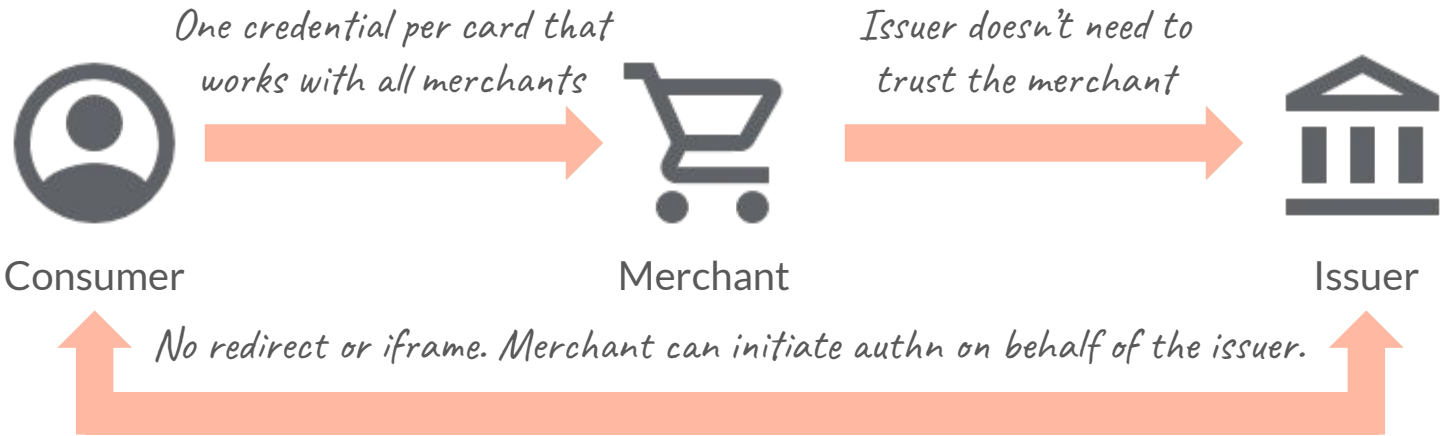
Secure Payment Confirmation

Goal:
Enable **strong customer authentication** during checkout that works at scale for **all payment methods** and **across all merchants**.

Design Principles:

1. Great consumer experience (fast and low-friction)
2. Ease of adoption for merchants and issuers
3. Strong technology that satisfies PSD2 SCA, including dynamic linking
4. Privacy preserving by default

Consumer, Merchant, Issuer



SPC allows **issuers or its representatives** (e.g. ACS, DS) to **directly authenticate** the consumer **via FIDO**

Secure Payment Confirmation

Approach:

- FIDO + Payment Request API: explainer
- Pilot use case: drop-in upgrade of 3D Secure step-up challenge

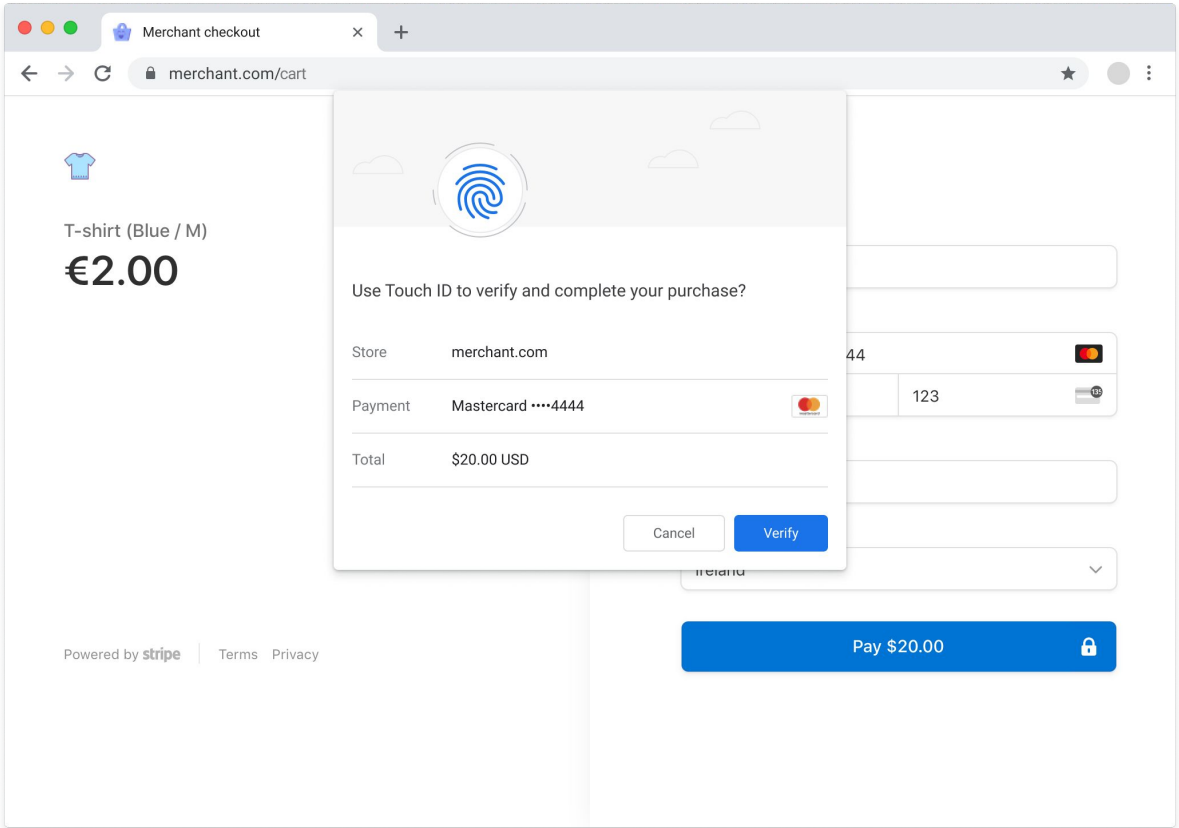
API Superpowers:

- Cryptographic binding of transaction details into the authentication assertion
- Allow any merchant to exercise Payment Credential owned by any issuer

Stripe Pilot (2020H2): validate UX

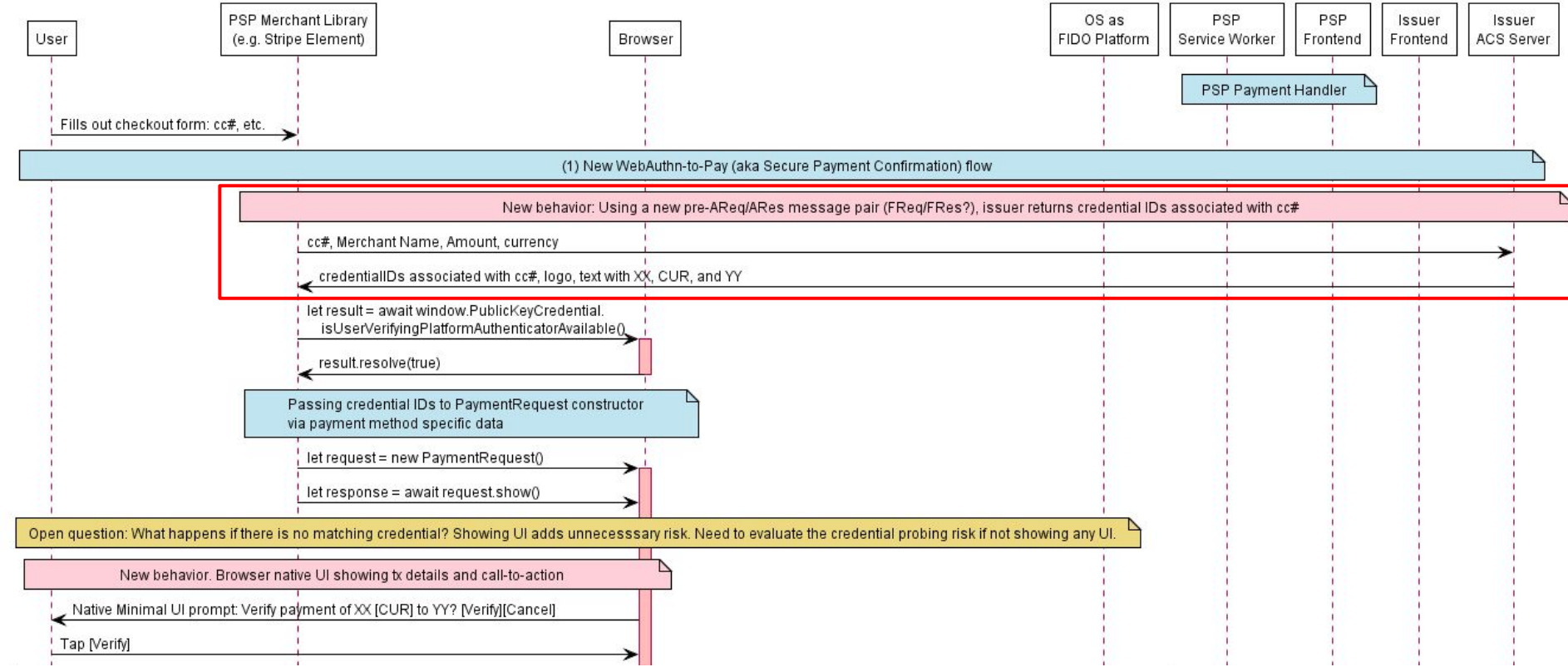
- Chrome 86 + macOS with user-verifying platform authenticator

Live Demo: Stripe

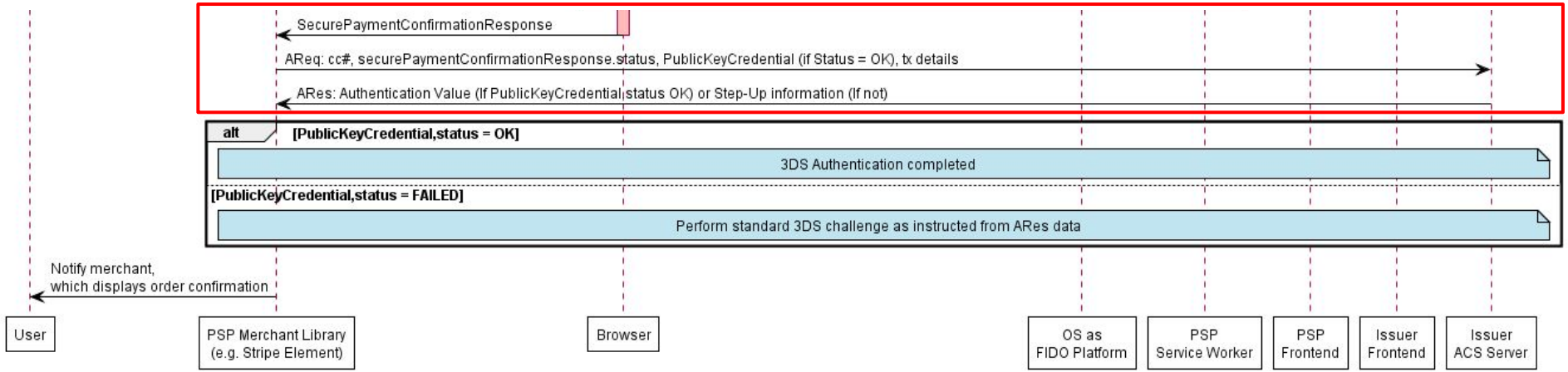

Merchant checkout
merchant.com/cart
T-shirt (Blue / M)
€2.00
Use Touch ID to verify and complete your purchase?
Store
merchant.com
Payment
Mastercard ····4444
Total
$20.00 USD
Cancel
Verify
44
123
Pay $20.00
Powered by stripe
Terms
Privacy

High Level Flows

- **Enrollment**
  - User starts on an issuer/ACS page as part of an existing step-up challenge
  - Issuer/ACS is the Relying Party
  - User registers a PaymentCredential to their device, which consists of:
    - A vanilla PublicKeyCredential -> private key in authenticator
    - Payment instrument metadata in Browser storage
- **Checkout**
  - Merchant uses a new 3DS message pair (details TBD) to exchange PAN for credential IDs
  - Merchant submits credential IDs to Browser via PR API to initiate SPC
  - Browser binds transaction details into the WebAuthn challenge and prompts the user to confirm transaction.
  - Browser returns the generated Web Payment Assertion to Merchant.
  - Merchant submits the Web Payment Assertion to ACS via AReq/ARes (details TBD).



Tentative 3DS Flow Proposal (credit: Christian Aabye, Dough Fisher)



Tentative 3DS Flow Proposal (cont'd)

Enrollment

```
const instrument = {
  displayName: 'FancyCard ····1234',
  icon: 'https://fancybank.com/card-art.png',
};

const rp = {
  id: 'fancybank.com',
  name: 'Fancy Bank',
};

const pubKeyCredParams = [{
  type: 'public-key',
  alg: -7,
}];
```

```
const payment = {
  rp,
  instrument,
  challenge,
  pubKeyCredParams,
};

const publicKeyCredential = await
navigator.credentials.create({payment});

console.log(publicKeyCredential.rawId);
```

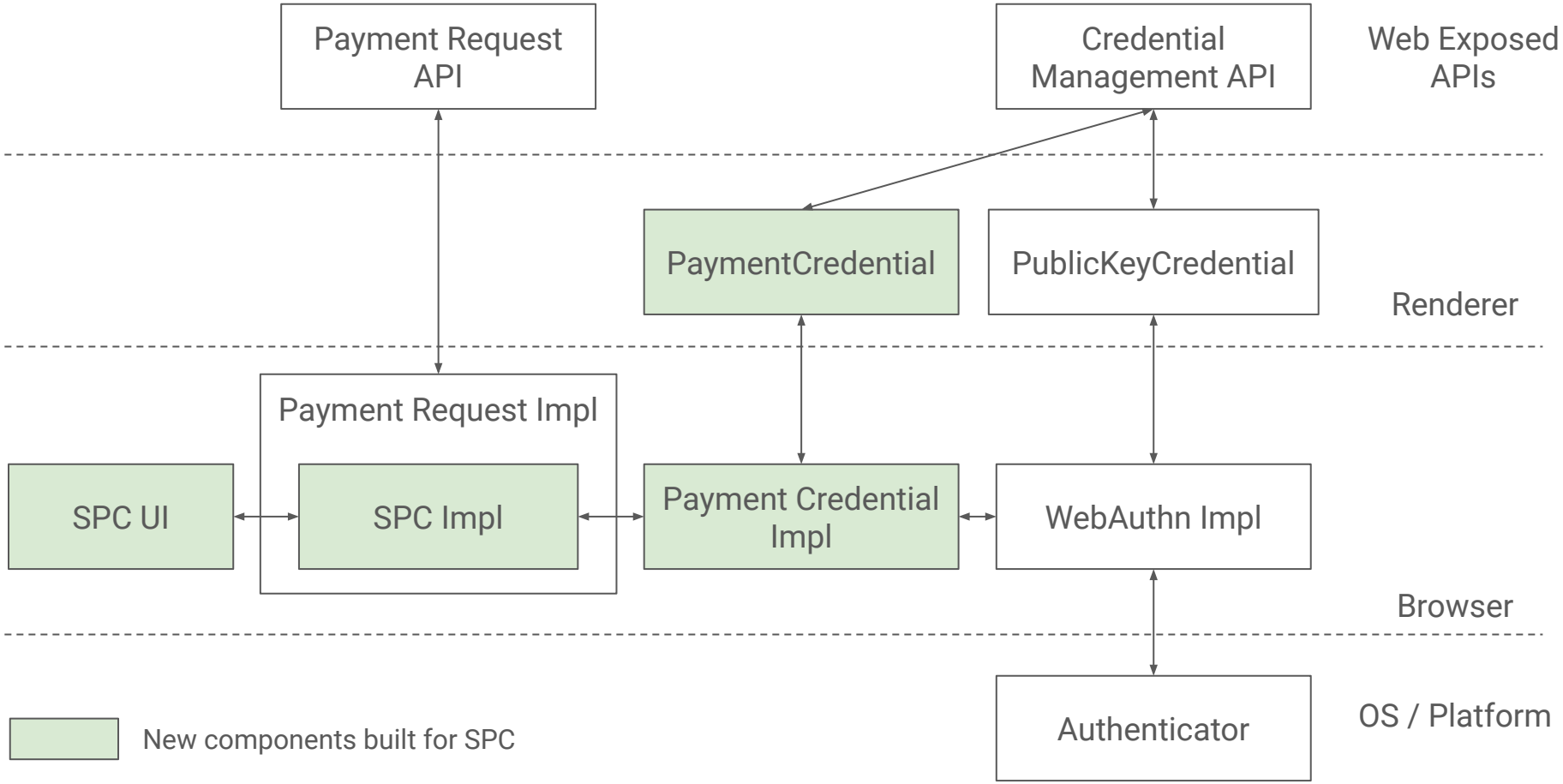
Checkout

```
let request = new PaymentRequest([{
  supportedMethods:
'secure-payment-confirmation',
  data: {
    action: 'authenticate',
    credentialIds,
    networkData,
    timeout: 60000
  }], {
  total: {
    label: 'Total',
    amount: { currency: 'USD', value: '0.01' },
  },
});
```

```
let response = await request.show();
console.log(response);
```

Sample Response


```
{
  "requestId": "3183fbc3-fbbe-43b2-938e-a289af16af66",
  "methodName": "secure-payment-confirmation",
  "details": {
    "challenge":
"{\"merchantData\":{\"merchantOrigin\":\"https://fancybank.com\",\"total\":{\"currency\":\"USD\",\"va
lue\":\"0.01\"}},\"networkData\":\"bmV0d29ya19kYXRh\"}",
    "signature":
"MEUCIAuUeARWZRB8yu9yCqN3cZp4k1UOWCY8hulJN2SZYcChAiEAmGaofxIUVPpBPqmKoR6IujAWeWa+aeK7SVMX6JWCmvk=",
  },
}
```


Payment Request API
Credential Management API
Web Exposed APIs
PaymentCredential
PublicKeyCredential
Renderer
Payment Request Impl
SPC UI
SPC Impl
Payment Credential Impl
WebAuthn Impl
Browser
Authenticator
OS / Platform
New components built for SPC

Pilot Roadmap

- **Oct 6**: SPC released in Chrome 86, behind an Origin Trial
- **End of Oct:** Stripe starts production experiment (~8 weeks)
- **Nov 2020**: early data
- **January 2021**: full analysis

Next Steps in 2021

1. **Productionize SPC for 3D Secure**
   - EMVCo 3DSWG: adoption into 2.3 & backward compatibility
   - ACS Implementation
   - Exit Origin Trial
2. **SPC on more platforms (Windows, Android)**
   - Exploring interoperability with Android native SDK
   - Standardization @ W3C
3. **SPC for other payment methods**

Looking for more **developer feedback** on priorities & adoption partners, and **browser interests** for standardization.

FAQ

Q. Does SPC require 3D Secure?
A. No. Parties that use SPC can leverage WebAuthn credentials from other sources.

Q. Can enrollment happen on the bank's website?
A. Yes! "Inline" and "direct" enrollments are complementary to each other.

Q. Is SPC a payment method?
A. It is so for simplicity of the pilot. If there is a need, it can be turned into a feature available to other payment methods.

Q. Will payment apps be able to use SPC?
A. We'd like to support that, but it's not yet implemented. Please talk to us!

Open Discussion

- **Merchant & issuer developers:**
  - Can this API be adopted into your existing 3DS and/or SCA flows?
  - What is missing? What can be improved?
  - What should we work on first?
- **Browser vendors:**
  - Would you be interested in implementing SPC? If not, why not?
- **Feedback?** Please file issues on GitHub.