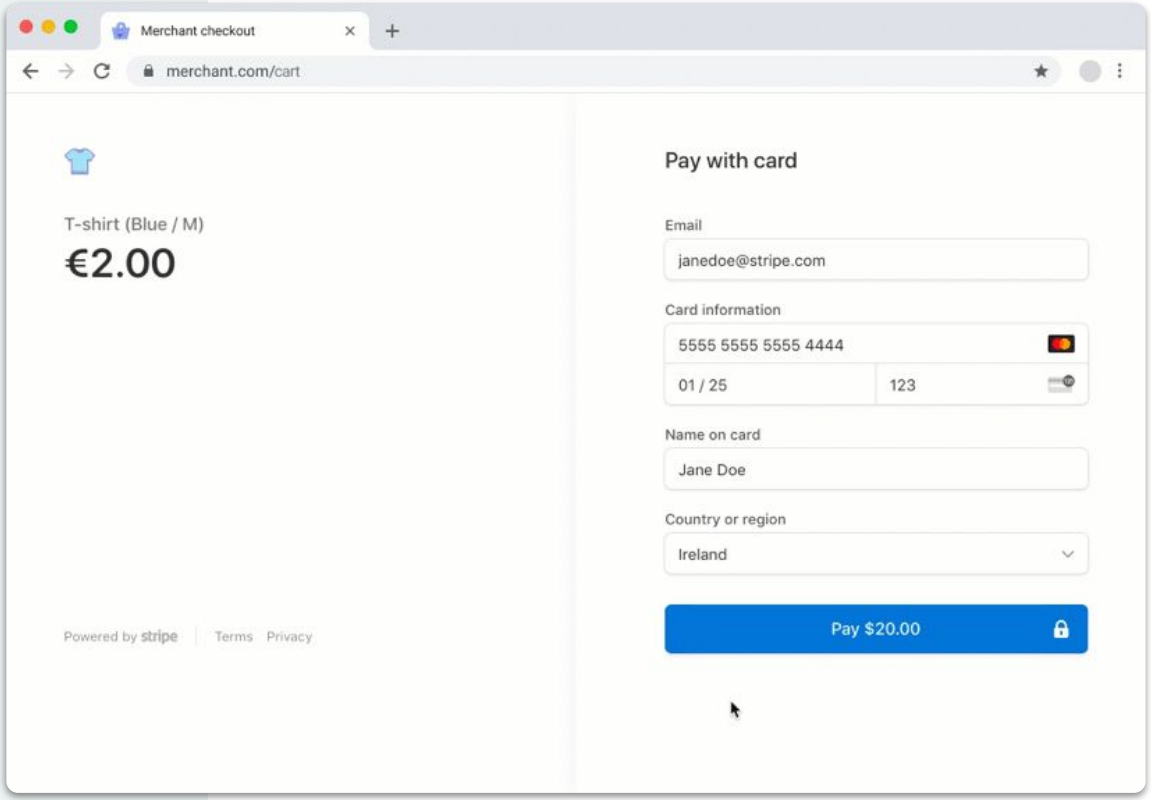
Web Payments Team

SPC: TPAC 2021 Update

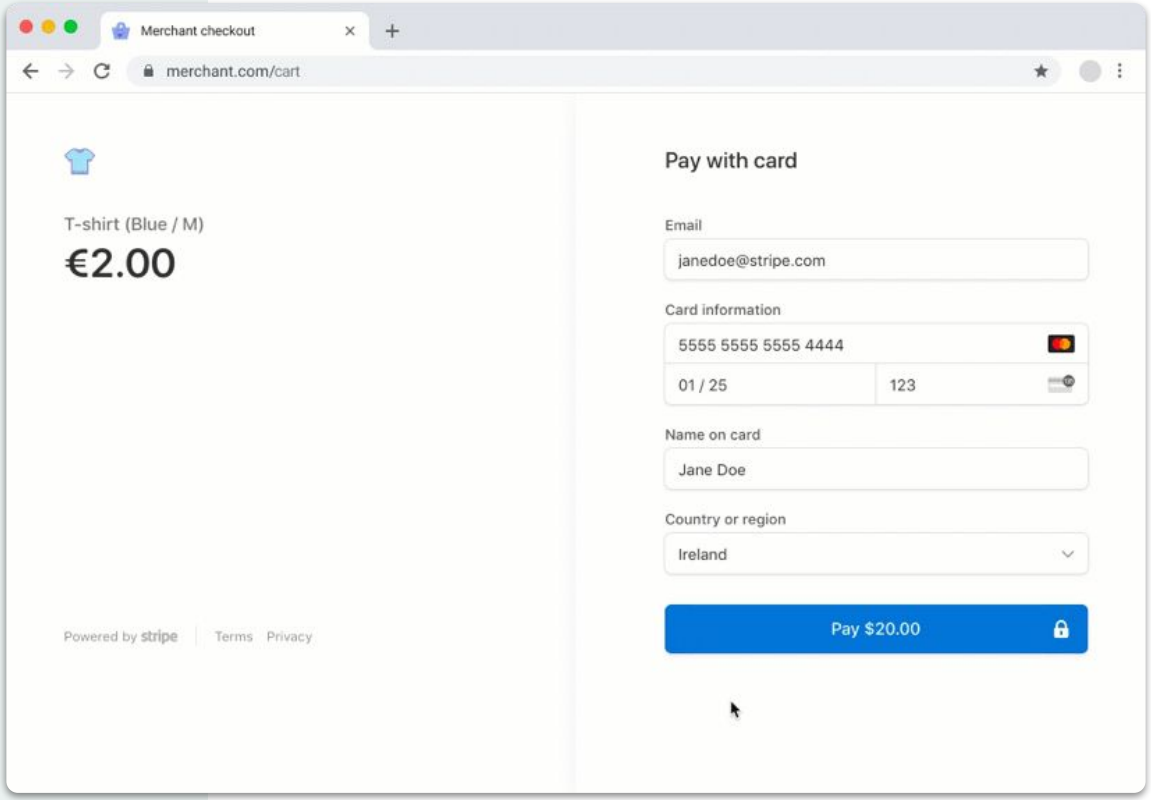
TPAC October 2021 #wpwg

smcgruer@google.com

Simple, seamless, and secure user authentication for payments on the web



Simple, seamless, and secure user authentication for payments on the web - in a privacy preserving way



Question

Why not just WebAuthn?

Answer

Why not just WebAuthn?

1. Requires first-party context
2. Lacks payment-specific integration

Where we started 01

What we've been up to 02

SPC Today 03

The Future 04

Where we started

- ~July 2020: first proposal for 'WebAuthn to Pay'

Where we started

- ~July 2020: first proposal for 'WebAuthn to Pay'
- ~Oct 2020: Chrome shipped support for SPC 'v0.1' in an Origin Trial

Where we started

- ~July 2020: first proposal for 'WebAuthn to Pay'
- ~Oct 2020: Chrome shipped support for SPC 'v0.1' in an Origin Trial
- Nov 2020 - Jan 2021: Stripe ran a pilot study comparing SPC against 3DS2 step-up challenges, with **great results**.

Where we started

- ~July 2020: first proposal for 'WebAuthn to Pay'
- ~Oct 2020: Chrome shipped support for SPC 'v0.1' in an Origin Trial
- Nov 2020 - Jan 2021: Stripe ran a pilot study comparing SPC against 3DS2 step-up challenges, with **great results**.

Where we started 01

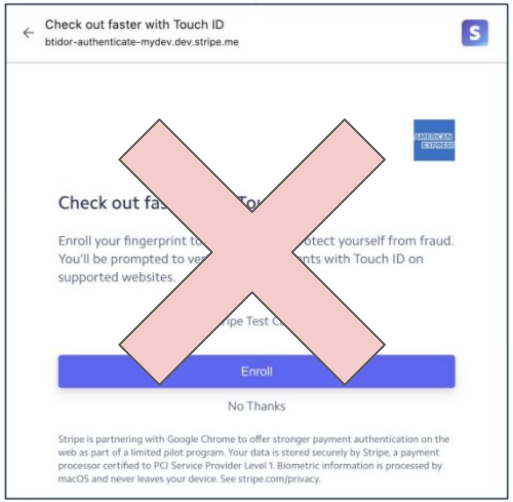
What we've been up to 02

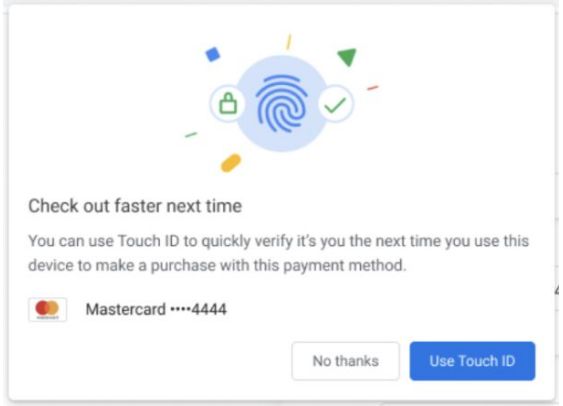
SPC Today 03

The Future 04

Iterate...

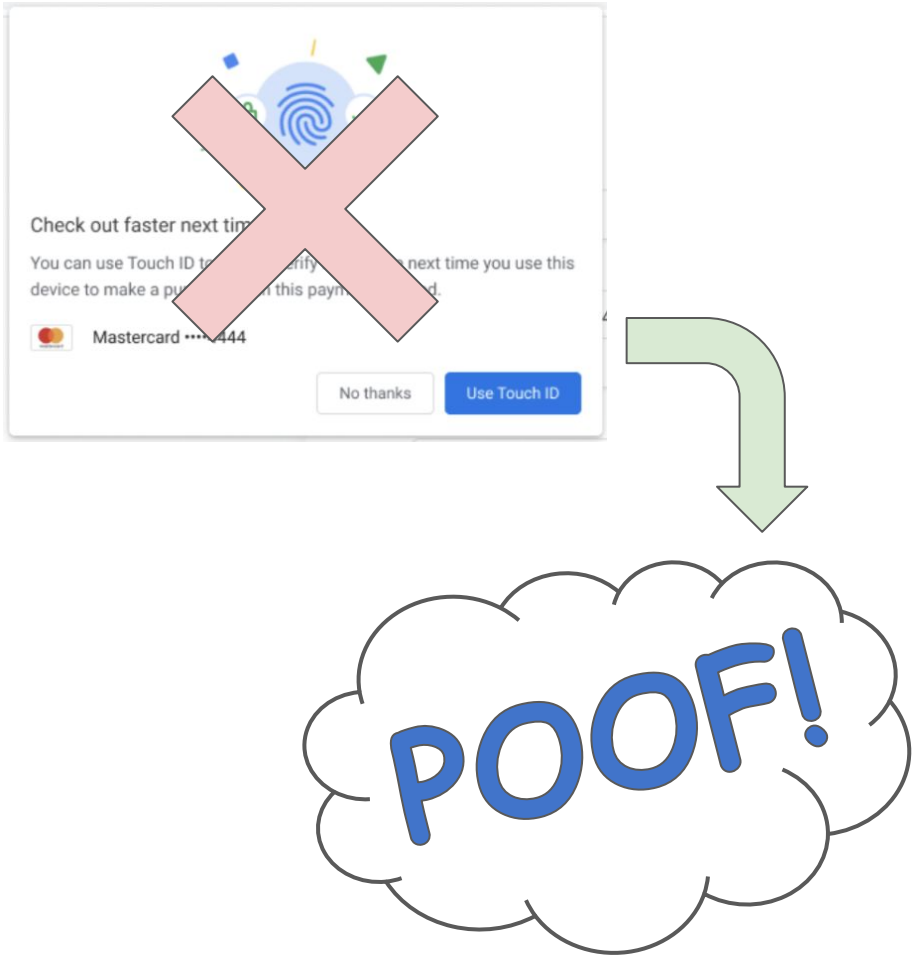
1. Enrollment: Payment Handler → Iframe





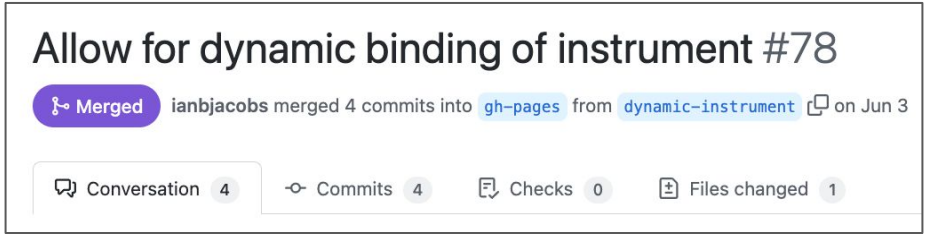
Iterate...

1. Enrollment: Payment Handler → Iframe
2. Enrollment: Iframe w/ UX → Iframe w/out UX



Iterate...

1. Enrollment: Payment Handler ⟶ Iframe
2. Enrollment: Iframe w/ UX ⟶ Iframe w/out UX
3. API changes



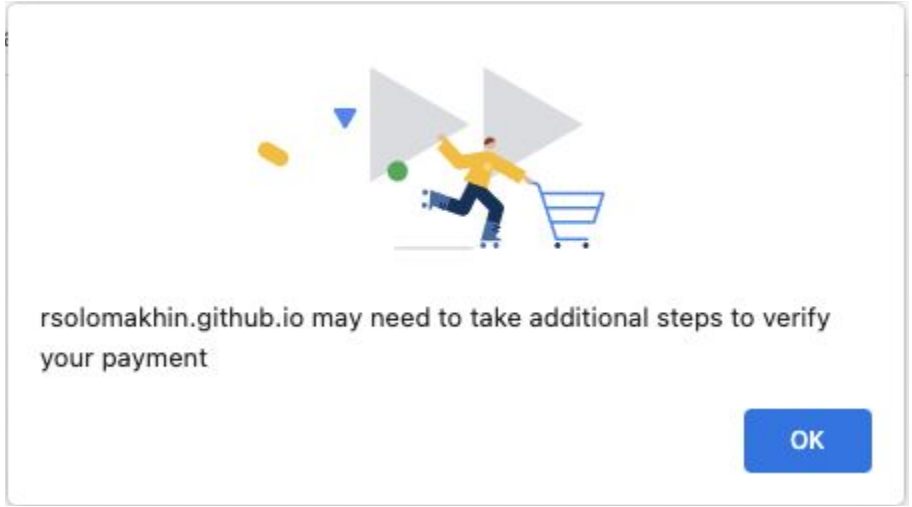

```
PaymentCredential → 'payment' extension

Custom output format → PublicKeyCredential

Challenge encoding → ClientDataJSON augmentation
```

Iterate...

1. Enrollment: Payment Handler ⟶ Iframe
2. Enrollment: Iframe w/ UX ⟶ Iframe w/out UX
3. API changes
4. Defeating privacy attacks



... and launch!

... and launch!

(Chrome M95, Mac + Windows)

Where we started 01

What we've been up to 02

SPC Today 03

The Future 04

Registration

1. WebAuthn entrypoint, with a 'payment' extension

2. Payment extension:
   a. Unlocks registration in cross-origin iframe
   b. Marks credential as SPC-enabled

3. Call returns a normal WebAuthn credential!

```
const publicKey = {
  user: {
    id: userId,
    name: 'Stephen Acc ***1234',
    displayName: 'Stephen'
  },

  rp: {
    name: 'Fancy Bank'
  },

  pubKeyCredParams: [{
    type: 'public-key', alg: -7,
  }],

  authenticatorSelection: {
    residentKey: 'required',
    authenticatorAttachment: 'platform',
    userVerification: 'required',
  },

  challenge: new Uint8Array([...]),
  timeout: 60000,

  extensions: { payment: { isPayment: true } }
};

const credential = await
    navigator.credentials.create({publicKey});
```

Authentication

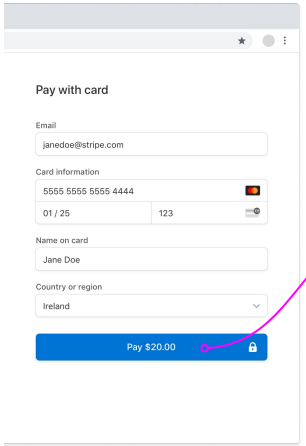
1. Still using PaymentRequest.

2. Pass in:
   a. List of credential IDs
   b. Challenge
   c. Instrument info
   d. Payee origin

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

const request = new PaymentRequest([{
  supportedMethods:
      'secure-payment-confirmation',
  data: {
    credentialIds,
    challenge,
    instrument,
    payeeOrigin: 'https://merchant.com',
    timeout: 60000
  }], {
    total: {
      label: 'Total',
      amount: {
        currency: 'USD',
        value: '5.00'
      },
    },
});

const response = await request.show();
await response.complete('success');
```

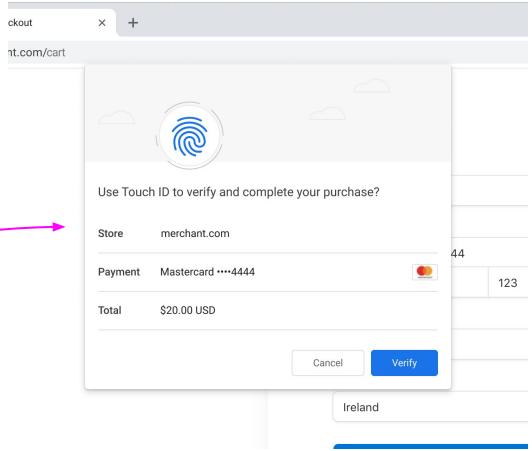


1 Merchant checkout

Credentials match?

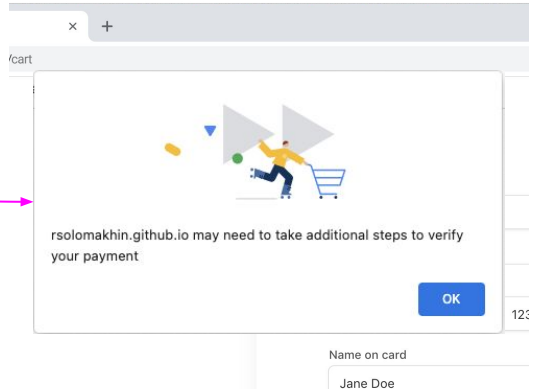
2 Internal Logic (invisible to users)

Match; show SPC transaction UX



3a SPC Authentication

No match; show SPC 'error' UX



3b SPC 'Error' Screen

Authentication

1. Return value is a WebAuthn PublicKeyCredential, with payment-specific information.

```
const response = await request.show();
await response.complete('success');

response.details === {
    id: "Aa3tQPG7...",
    rawId: ArrayBuffer<>,
    response: {
      authenticatorData: ArrayBuffer<>,
      clientDataJSON: {
        type: 'payment.get',
        challenge: '...',
        origin: 'https://psp.com',
        crossOrigin: true,
        payment: {
          rp: 'fancybank.com',
          topOrigin: 'https://merchant.com',
          total: { ... },
          instrument: { ... },

        },
      signature: 'MEUCICoR3m...',
    },
    type: "public-key"
}
```

Where we started 01

What we've been up to 02

SPC Today 03

The Future 04

1. Solve the cross-browser problem
2. Android support
3. More ergonomic API
4. More privacy considerations
5. More: dark mode support, other OSes, ...

Thank you