Combining WebAuthn and Payment Handler Gestures

Adrian Hope-Bailie
March 2020

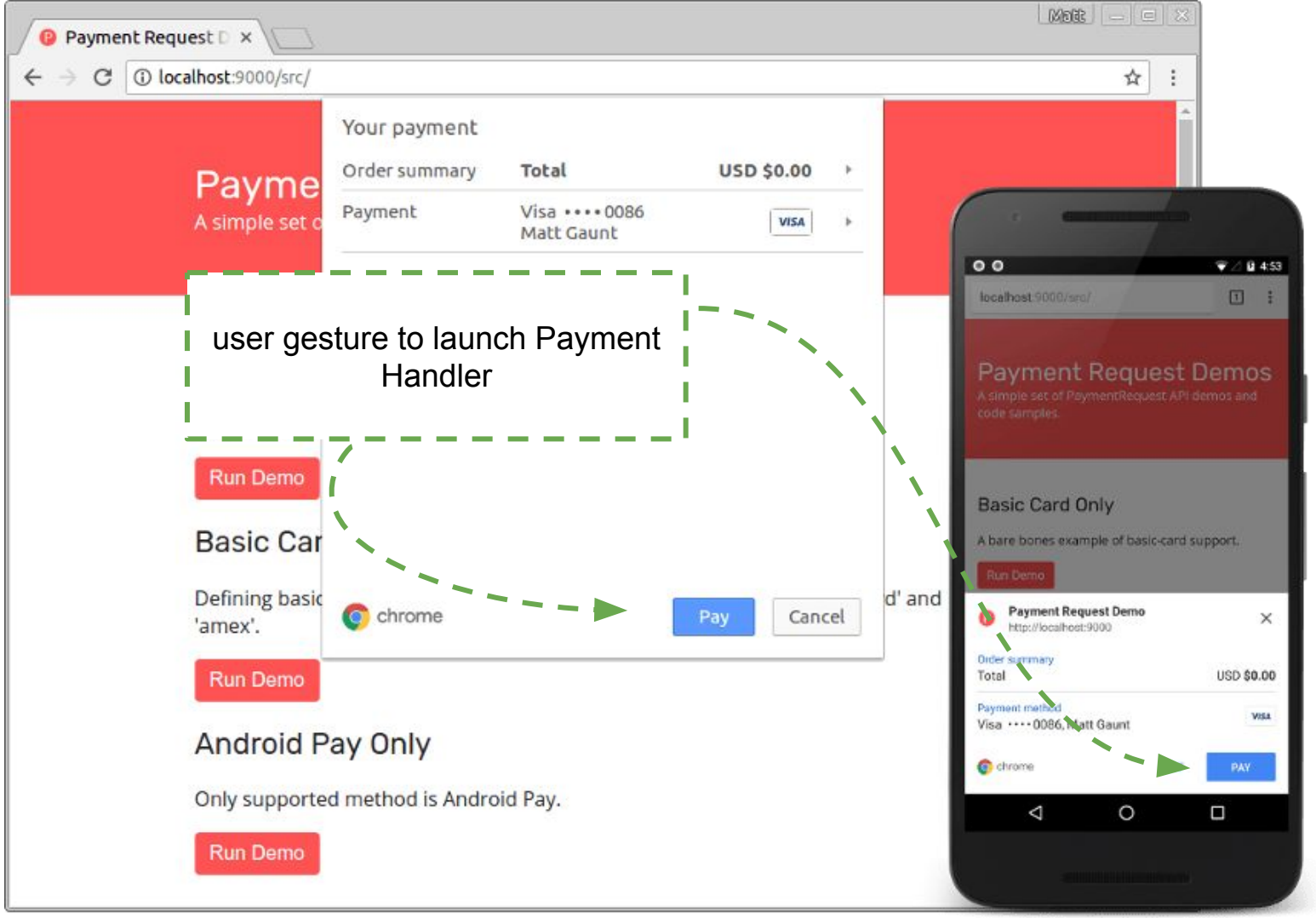
Goals

Increase security

- Use WebAuthn for authentication

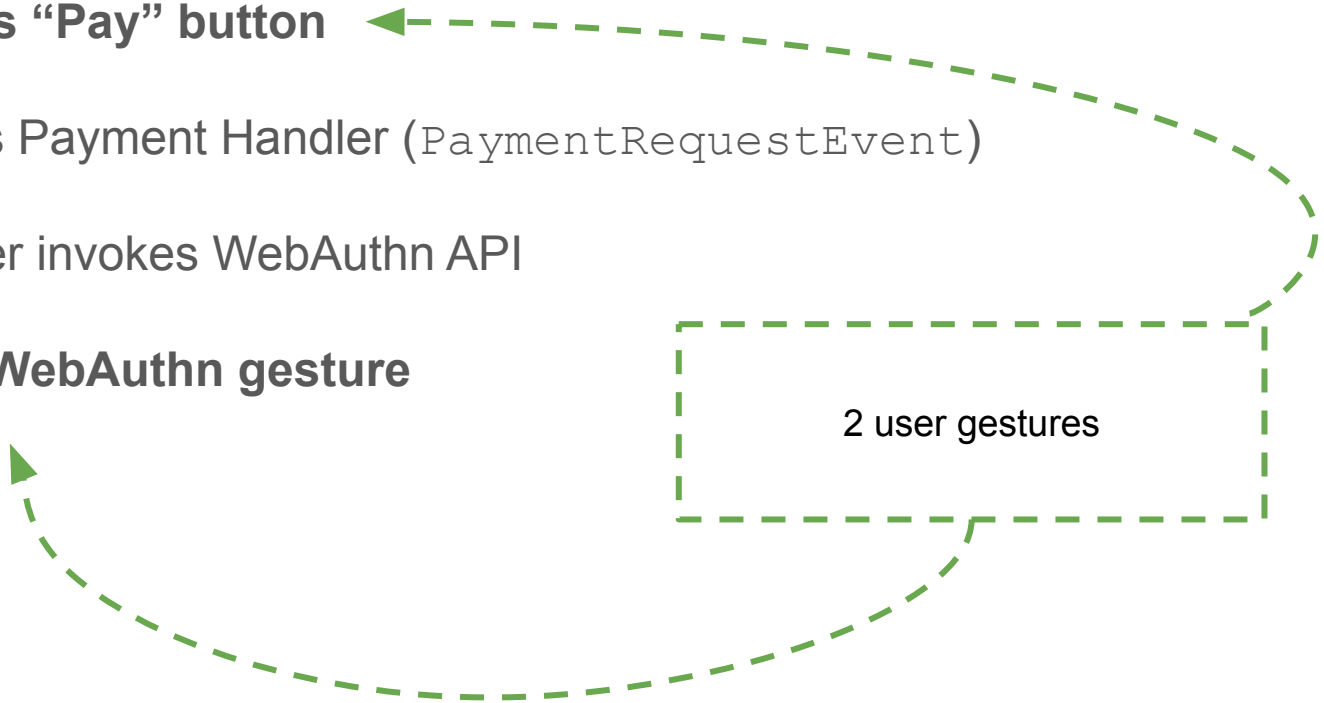
Reduce friction for users

- Reduce number of user interactions


Payment Request D
localhost:9000/src/
Your payment
Order summary
Total
USD $0.00
Payment
Visa ••••0086
Matt Gaunt
VISA
user gesture to launch Payment Handler
Run Demo
Basic Car
Defining basic
'amex'.
Run Demo
Android Pay Only
Only supported method is Android Pay.
Run Demo
chrome
Pay
Cancel
Payment Request Demos
A simple set of PaymentRequest API demos and code samples.
Basic Card Only
A bare bones example of basic-card support.
Run Demo
Payment Request Demo
http://localhost:9000
Order summary
Total
USD $0.00
Payment method
Visa ••••0086, Matt Gaunt
chrome
PAY

Current Flow with WebAuthn

1. **User clicks/taps “Pay” button**
2. Browser invokes Payment Handler (`PaymentRequestEvent`)
3. Payment Handler invokes WebAuthn API
4. **User provides WebAuthn gesture**



Proposed Flow with WebAuthn (at Registration)

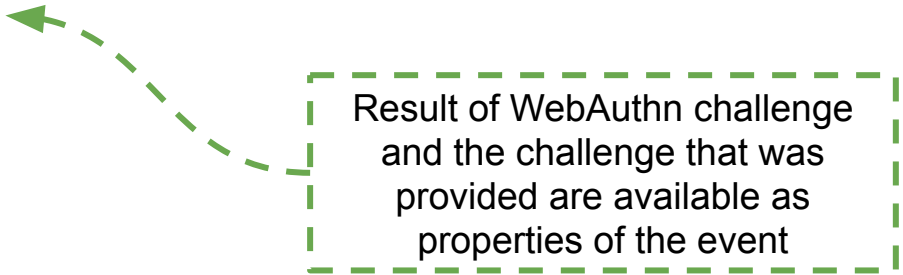
```
registration.paymentManager.instruments.set(
    "dc2de27a-ca5e-4fbd-883e-b6ded6c69d4f",
    {
        name: "Visa ending ****4756",
        method: "basic-card",
        credentials: [{
            publicKey: {
                allowCredentials: [ … ],
                timeout: 60000
            }
        }]
    }),
```

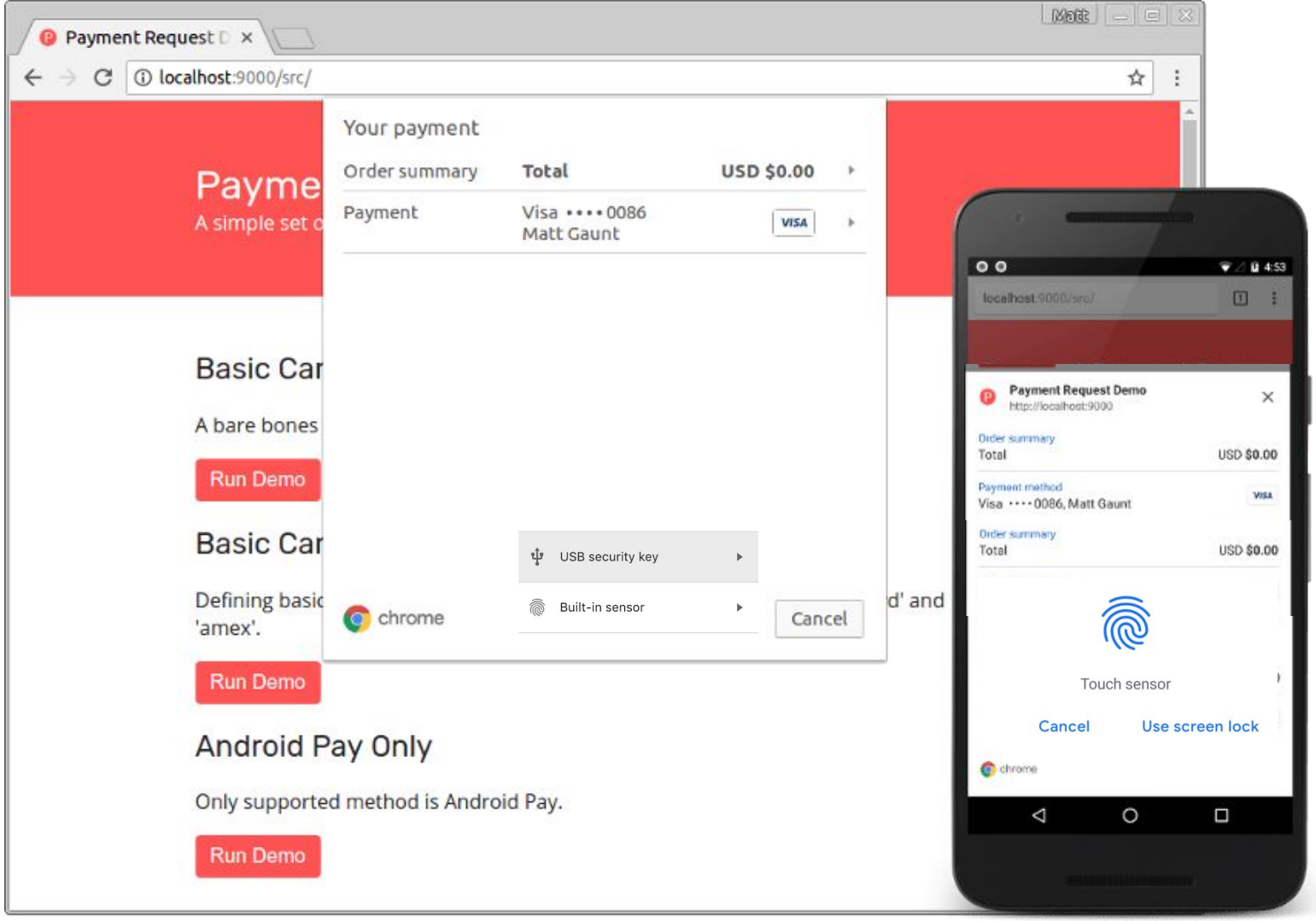
Proposed Flow with WebAuthn (at Payment)

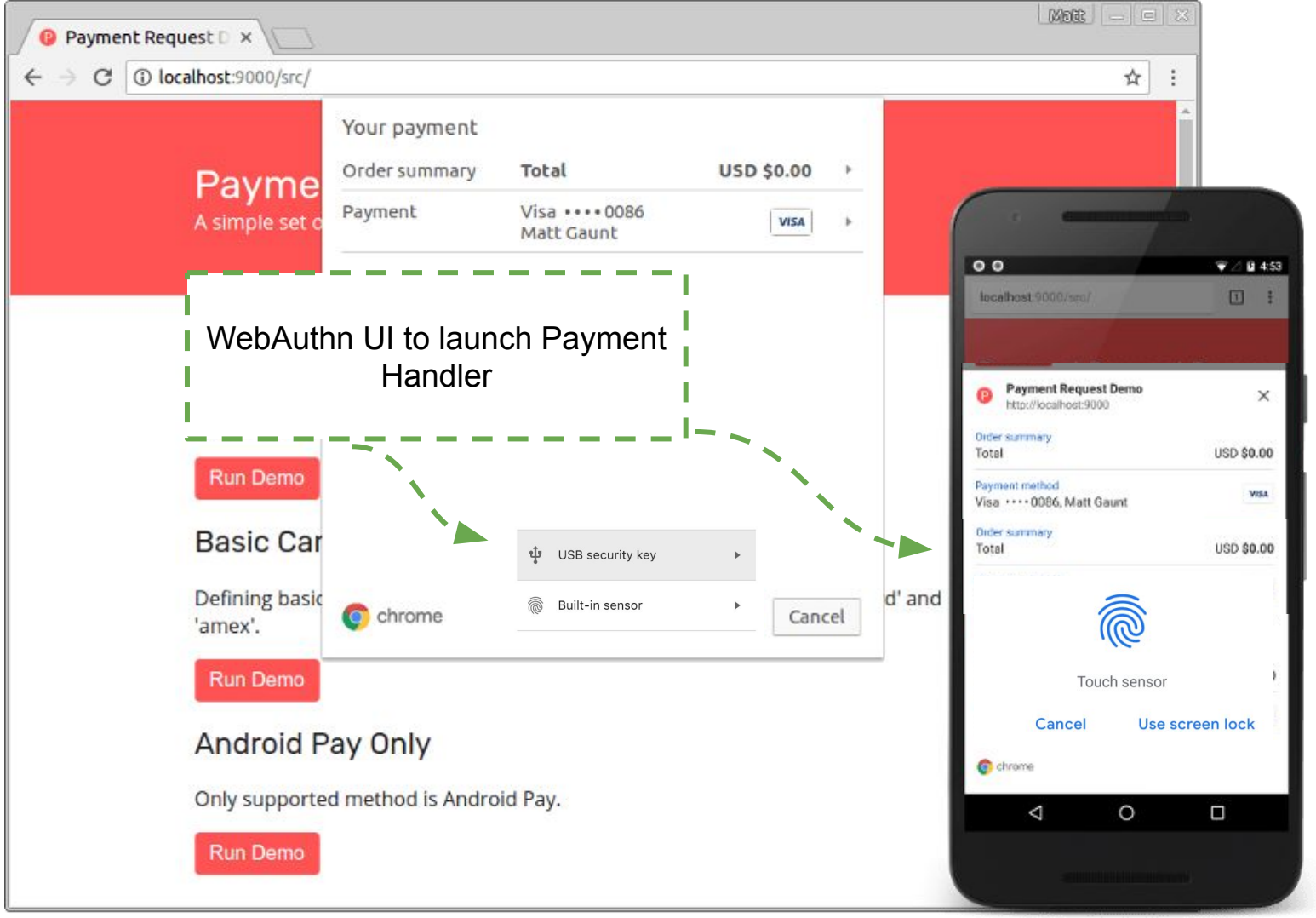
Browser performs `navigator.credentials.get(x)` algorithm where `x` is `credentials` provided during registration with challenge added.

Challenge is generated using payment details and Payment Request ID (random UUID) using a standard deterministic algorithm.

```
self.addEventListener("paymentrequest", function(e) {
  e.respondWith(new Promise(function(resolve, reject) {
    const assertion = e.credential;
    const challenge = e.challenge;
    resolve({assertion, challenge})
  }));
});
```




Payment Request D
localhost:9000/src/
Matt
Your payment
Order summary
Total
USD $0.00
Payment
Visa ••••0086
Matt Gaunt
VISA
Payme
A simple set o
Basic Car
A bare bones
Run Demo
Basic Car
Defining basic
'amex'.
d' and
Run Demo
Android Pay Only
Only supported method is Android Pay.
Run Demo
chrome
USB security key
Built-in sensor
Cancel
4:53
localhost:9000/src/
Payment Request Demo
http://localhost:9000
Order summary
Total
USD $0.00
Payment method
Visa ••••0086, Matt Gaunt
VISA
Order summary
Total
USD $0.00
Touch sensor
Cancel
Use screen lock
chrome


Payment Request D
localhost:9000/src/
Your payment
Order summary
Total
USD $0.00
Payment
Visa ••••0086
Matt Gaunt
VISA
WebAuthn UI to launch Payment Handler
Run Demo
Basic Car
Defining basic
'amex'.
Run Demo
Android Pay Only
Only supported method is Android Pay.
Run Demo
USB security key
Built-in sensor
chrome
Cancel
Payment Request Demo
http://localhost:9000
Order summary
Total
USD $0.00
Payment method
Visa ••••0086, Matt Gaunt
Order summary
Total
USD $0.00
Touch sensor
Cancel
Use screen lock
chrome

Notes and Discussion

- This proposal is not likely to work for JIT install as WebAuthn requires an explicit credential enrollment step.
- Question: Challenge is generated by the browser, which saves a round-trip to the server. Will that work?
- Question: Are scenarios where this flow should not be followed (e.g., the payment handler is not yet set up for this)?
- Would it be useful to augment this proposal for other forms of authentication through the Credentials API?