

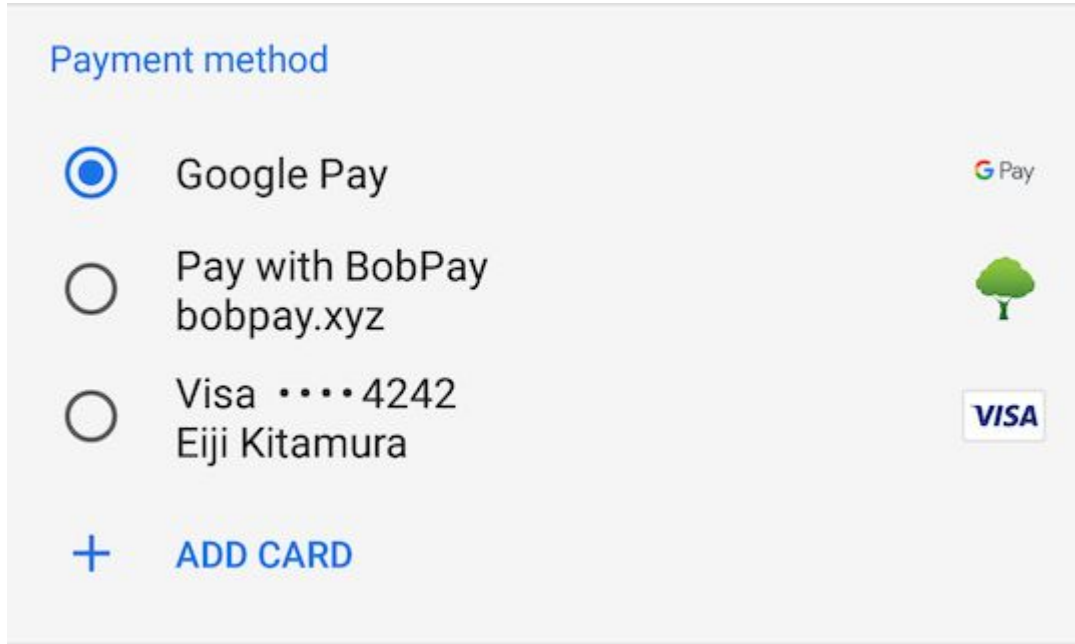


Handling Payments

A fresh look at the Payment
Handler API



Basic Card payment method has a bad smell...



← Payment Handler 👍



← Payment Handler 👍



← Not a Payment Handler
💩

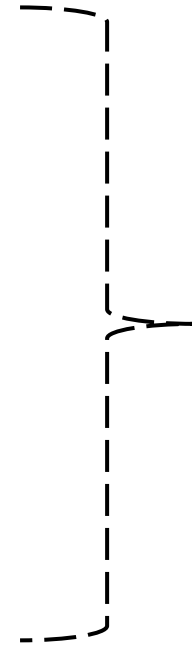
Delegating to Payment Handlers changes the game

Things payment handlers will be able to do

- Collect shipping details
- Collect name
- Collect email address
- Collect phone number

Things payment handlers do already

- Display amount (SCA etc.)



Currently
done by
the
Payment
Sheet

Do we even need the payment sheet?

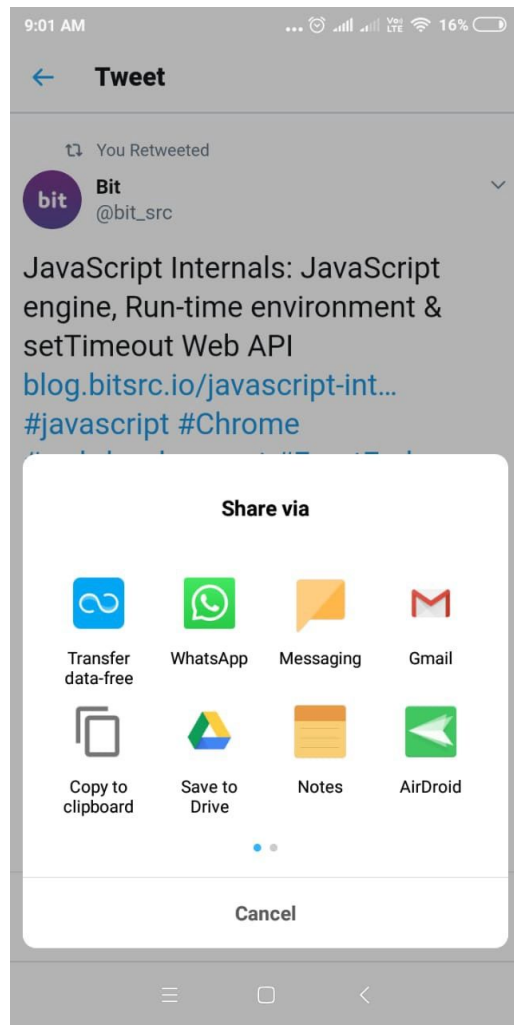
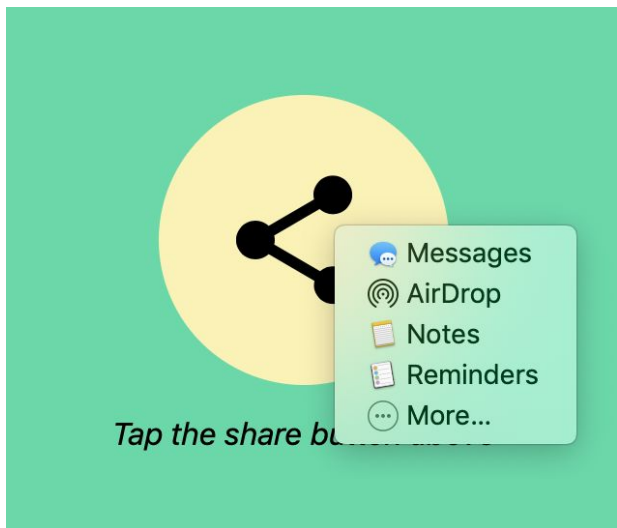
- *Payment Sheet* is just a UI for a built-in ***Payment Handler for Basic Card***
- Adds an unnecessary extra click
- Requires LOTS of work from browsers each time we propose changes to PR API

We're already **skipping the sheet**
let's **ditch the sheet...**

What would that look like?

WebShare API as reference:

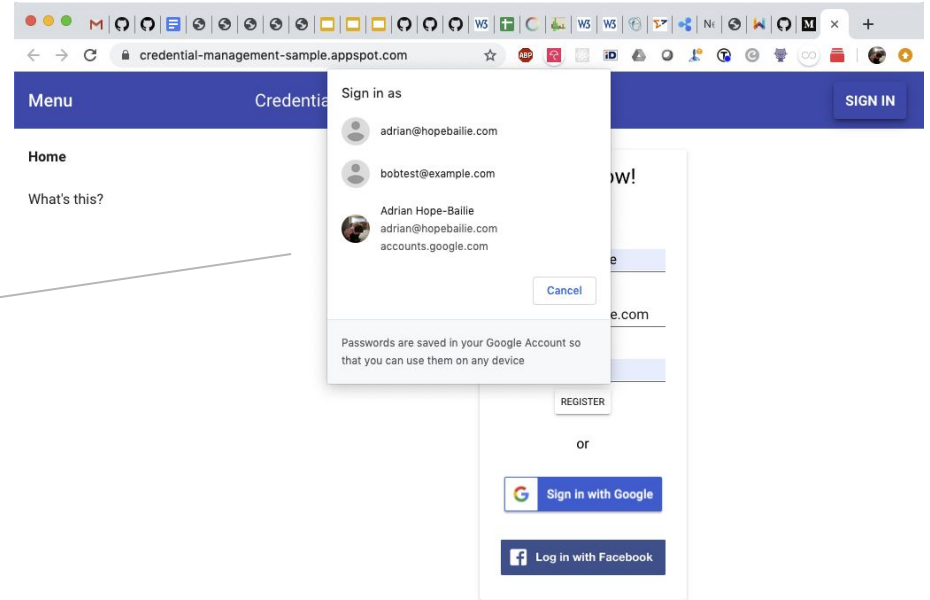
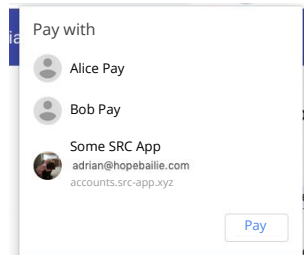
<https://web-share-demo.glitch.me>



What would that look like?

Credentials API account picker as reference:

<https://credential-management-sample.appspot.com/>



And while we're at it... Extensible Web!

`PaymentRequestEvent.openWindow()` should be a **Web platform feature** for opening **modal dialogs** not a special *payments* thing.

Powerful feature for delegated... anything across origins

We don't need instrument detail if users are just given a list of apps, hugely simplifying the API.

"This thing is becoming a beast" - smart guy describing Payment Handlers

Would it unblock implementers?

Native payment apps could be invoked via platform APIs potentially unlocking existing app ecosystem

No need to “solve” basic card for Mozilla

Secure Modal Dialog investment would be a widely used platform feature so doing the work is worth it anyway

What next?

Spin ***Modal Dialogue*** off into own spec.

Test our assumptions around dropping the sheet.

Trim down Payment Handler:

- Drop Instruments
- Drop Modifiers
- Drop `openWindow()`