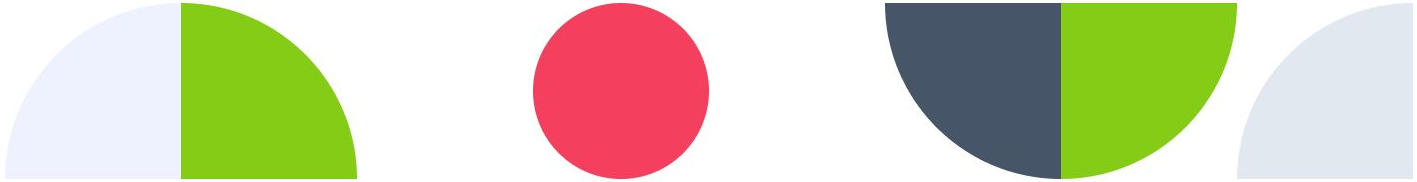
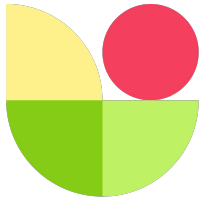


GNAP + SPC



fynbos.

Summary

fynbos.

Internet Draft describes a GNAP extension using SPC as an interaction mode

https://www.ietf.org/archive/id/draft-ozdemir-gnap-spc-extension-00.html

Appropriate for use when GNAP is used to authorize a payment.

ID will continue to evolve as SPC evolves within W3C.

We are requesting comments and implementation experience.

Secure Payment Confirmation (SPC)

fynbos.

Work item of the **Web Payments WG** at **W3C**

https://www.w3.org/TR/secure-payment-confirmation/

TL;DR: Signing transaction data (not just a challenge) using WebAuthn

- Invoked via Web API
- Cross-origin allowances to facilitate merchant initiation without redirect
- Payment specific UI to prevent abuse by trackers etc.

How does SPC work with GNAP?

fynbos.

1. **Client** tests if SPC is possible (is end user using a browser which supports SPC?)
2. **Client** requests a grant to perform a payment from authorization server (AS)
    a. Specifies SPC as a possible interaction mode.
    b. Provides user identity hints and/or assertions.
3. **AS** determines SPC is preferred interaction AND user has enrolled credentials.
4. **AS** requests client perform SPC and provides challenge and candidate credentials.
5. **Client** invokes SPC with payment details, candidate credentials and challenge.
6. **Client** returns SPC response to AS to finish interaction and continue grant request

Prerequisites for grant request

fynbos.

Client checks if SPC is supported before making grant request

- *Improvements to SPC API have been proposed to make this easier for clients.*

Client provides user identifiers and/or attestations

- *User identifiers and/or assertions MUST be passed in grant request to determine candidate credentials*

Client provides some device identification data

- *Many payment auth systems use device/browser fingerprints for risk signals or user recognition. How can these be passed in the grant request?*

GNAP Request + Response Examples

```
"access_token": {
  "access": ["make-payment"]
},
"client": "xyz-client-1234a",
"interact": {
  "start": [
    "spc"
  ]
},
"user": {
  "sub_ids": [{
    "subject_type": "email",
    "email": "user@example.com"
  }]
}
```

Grant Request

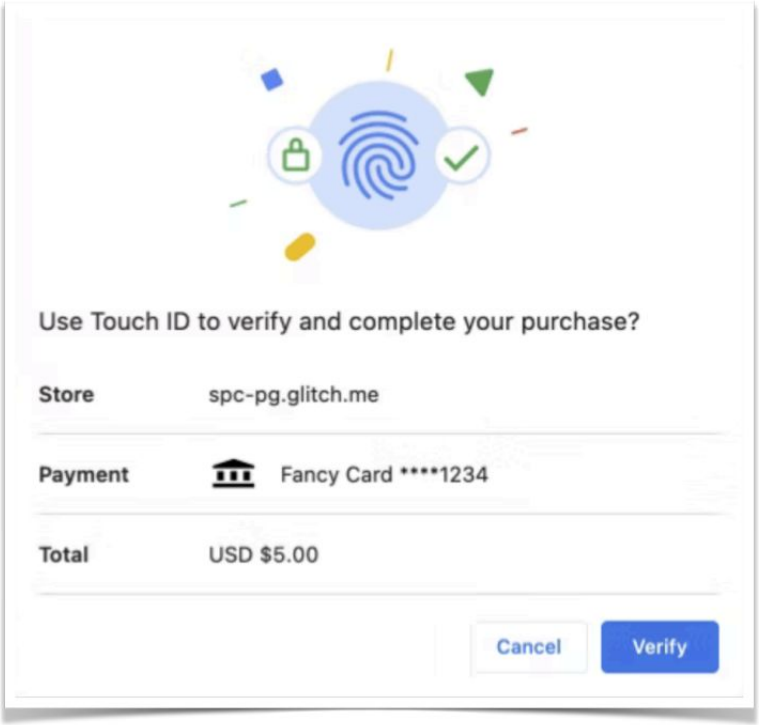
```
{
  "interact": {
    "spc": {
      "credential_ids": ["MTIzMjMxMzIyMz..."],
      "challenge": "dGhpcyBpcyBh...",
      "payment_instrument" : {
        "display_name": "Fancy Card *****1234",
        "icon": "https://fancybank.com/card-art.png",
        "icon_must_be_shown": true
      }
    }
  },
  "continue": {
    "access_token": {
      "value": "80UPRY5NM33OMUKMKSKU"
    },
    "uri": "http://fancybank.com/continue/5e69f364-b14d-4fdf-8b6b-3b6ffb52c339"
  }
}
```

Grant Response

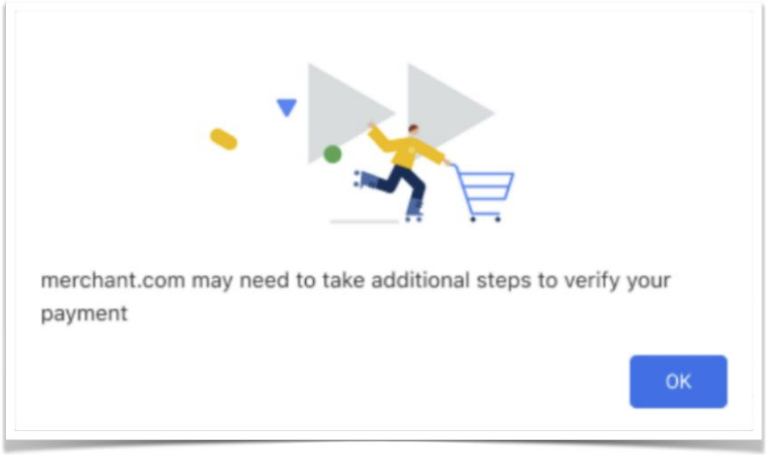
SPC API invoked in browser

```
const request = new PaymentRequest([{
  supportedMethods: "secure-payment-confirmation",
  data: {
    // List of credential IDs obtained from the AS.
    credentialIds,
    rpId: "fancybank.com",
    // The challenge is also obtained from the AS.
    challenge: new Uint8Array([21,31,105 /* 29 more random bytes generated by the AS */]),
    instrument: {
      displayName: "Fancy Card ****1234",
      icon: "https://fancybank.com/card-art.png",
    },
    payeeName: "Merchant Shop",
    payeeOrigin: "https://merchant.com",
    timeout: 360000,  // 6 minutes
  }], {
    total: {
      label: "Total",
      amount: {
        currency: "USD",
        value: "5.00",
      },
    },
  });
```

The SPC dialogue (Chrome examples)



Credential Match



No Credential Match

SPC response + Grant continuation

```
console.log(scpResponse);

PublicKeyCredential {
    id: 'ADSUllKQmbqdGtpu4sjseh4cg2TxSvrbcHDTBsv4NSSX9...',
    rawId: ArrayBuffer(59),
    response: AuthenticatorAssertionResponse {
        authenticatorData: ArrayBuffer(191),
        clientDataJSON: ArrayBuffer(118),
        signature: ArrayBuffer(70),
        userHandle: ArrayBuffer(10),
    },
    type: 'public-key'
}
```

SPC Response

```
{
  "public_key_cred": {
    "client_data_json": "ZXhhbXBsZSBjbGllbnRkYXR...",
    "authenticator_data": "YXV0aGVudGljYXRvckRhdGEg...",
    "signature": "c2lnbmF0dXJlIGV4YW...",
    "user_handle": "dXNlckhhbmRsZSBleG..."
  }
}
```

Grant Continue

Conclusion

fynbos.

We will continue to update the ID as SPC evolves.

- May define an access token schema that maps cleanly to the SPC request schema
  e.g. same total amount etc.

Is there interest in a generalised WebAuthn interaction mode (not payment specific)?

Example video: **https://www.youtube.com/watch?v=Bjr0T3apg7E**

- Fynbos uses URLs as payment instruments (called payment pointers)
  e.g. ***$fynbos.me/adrian*** *(equivalent to https://fynbos.me/adrian)*