SRC Payment Method Data Flow

Visualization of data flows for anticipated SRC Payment Method invoked by Payment Request API

September 14, 2019
Tomasz Blachowicz
Consumer Identity
Consumer Identity

Identity of the consumer must be known to the Payment Handler to fetch cards from SRC System(s) and facilitate checkout.

Preferred option for the browsers that maintain user profile

Consumer Identity managed by Payment Handler

- Proprietary method of Consumer Identity verification managed by the Payment Handler
- Payment Handler uses SRC Identity Services as per SRC API specification

Provided Consumer Identity

- Consumer identity provided by the browser
  - User agent provides the credentials into the Payment Handler before initialisation
  - Payment Handler requests user credentials via API e.g. Web Credentials API

- Consumer identity provided by the Merchant/Payee in the Payment Request

Note: Provider of the Consumer Identity must be trusted by the Payment Handler and underlying SRC Systems and should provide verifiable attestation/assurance.
Request/Response
interface PaymentRequest {
    constructor(
        sequence<PaymentMethodData> methodData,
        PaymentDetailsInit details,
        optional PaymentOptions options = {})
    }

dictionary PaymentMethodData {
    required DOMString supportedMethods;
    object data;
}
Payment Options

Payment Options specify level of requested data that should be provided by the SRC Payment Handler

```
dictionary PaymentOptions {
    boolean requestPayerName = false;
    boolean requestBillingAddress = false;
    boolean requestPayerEmail = false;
    boolean requestPayerPhone = false;
    boolean requestShipping = false;
    PaymentShippingType shippingType = "shipping";
};
```

Notes on SRC

- Billing address is associated with selected card
- Shipping address can be provided by SRC or DCF
- Consumer details i.e. name, email and phone can be provided by SRC
- Payment Options attributes map on corresponding attributes of DpaTransactionOptions directory in SRC specification

Alternative solution is to include individual parameters related to consumer details in SRC Payment Method
Payment Method Request

Request parameters specific to SRC defined in anticipated SRC Payment Method specification

Based on the SRC wiki
Payment Response

Payment Response contains final response form checkout with SRC system.

```javascript
interface PaymentResponse {
    readonly attribute DOMString requestId;
    readonly attribute DOMString methodName;

    readonly attribute object details;
    readonly attribute PaymentAddress? shippingAddress;
    readonly attribute DOMString? payerName;
    readonly attribute DOMString? payerEmail;
    readonly attribute DOMString? payerPhone;
    readonly attribute DOMString? shippingOption;
}
```

Notes on SRC

- Shipping address and consumer details are included in encrypted payload and cannot be extracted by Payment Handler into Payment Response.
- Details are specified by separate SRC Payment Method.

SRC 1.0 does not support shipping option selection.
Payment Method Response

 SRC-specific response data provided by SRC System

dictionary SrcCardResponse {
  required DOMString srcCorrelationId;
  DOMString payload; Encrypted Payload
  DOMString maskedEmailAddress;
  DOMString maskedPhoneNumber;
  DOMString countryCode;
  DOMString panBin; Displayable Data
  DOMString panLastFour;
  DOMString panExpirationYear;
  DOMString panExpirationMonth;

  //Assurance data Assurance Data
  //Event history Event History Data
  object customOutputData; Custom Output Data
}

SRC - specific response data provided by SRC System
Sample Request
const srcMethodData = {
  supportedNetworks: ["visa", "mastercard"],
  payeeIdentifier: "kicks.com-1",
  payeeName: "Kicks.com",
  payeeCountryCode: "JP",
  srciTransactionId: "0123-4567-890A-BCDE",
  payloadTypeIndicator: "FULL",
  dynamicDataPreference: [
    "CARD_APPLICATION_CRYPTOGRAM_SHORT_FORM",
    "DYNAMIC_CARD_SECURITY_CODE"
  ],
  dpaAcceptedShippingCountries: ["JP","AU"],
  assurancePreference: "SELF"
};

const methodData = [
  {
    supportedMethods: "src-card",
    data: srcMethodData,
  }
];

const options = {
  requestPayerEmail: false,
  requestPayerName: true,
  requestPayerPhone: false,
  requestShipping: true,
};
const details = {
  id: "kicks.com-123-567-789",
  displayItems: [
    {
      label: "Urban Sneakers",
      amount: { currency: "USD", value: "100.00" },
    },
    {
      label: "Sales Tax",
      amount: { currency: "USD", value: "10.00" },
      type: "tax"
    },
  ],
  total: {
    label: "Total due",
    amount: { currency: "USD", value: "110.00" },
  },
};

const request = new PaymentRequest(methodData, details, options);

// Handling PaymentMethodChangeEvent
request.onpaymentmethodchange = ev => ev.updateWith(details);

// Handling ShippingAddressChangeEvent
request.onshippingaddresschange = ev => ev.updateWith(details);

const response = await request.show();