

OpenAttestationEmbeddedRenderer

OpenAttestationEmbeddedRenderer is used by an [issuer](#) to render a [verifiable credential](#). The verifiable credential is rendered in HTML within an embedded <iframe> through a Template Renderer website referenced in the document. This arrangement allows for interactive selective disclosure using OpenAttestationMerkleProofSignature2018.

There currently exists a number of OpenAttestationEmbeddedRenderer issuers and decentralized renderer implementations.

Data Model

When an issuer desires to specify an embedded rendering instructions for a verifiable credential, they *MAY* add a render property that uses the data model described below.

Property	Description
id	A URL that locates a website that implements the OpenAttestationEmbeddedRenderer Action API .
type	OpenAttestationEmbeddedRenderer
rendererName	Name of the template used by the website specified by “id” to render the document. A different template can be used for the decentralized renderer to present a different HTML view of the verifiable credential.

We provide an example of the data model below. The verifiable credential specifies a decentralized renderer at "<https://demo-renderer.opencerts.io>", using the template named GOVTECH_DEMO.

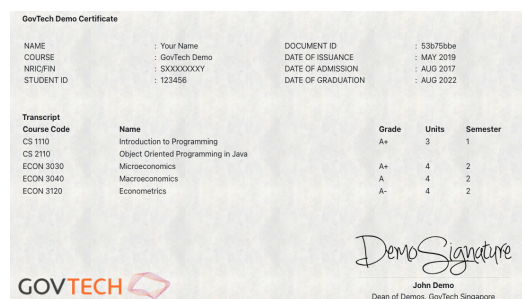
```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://schema.openattestation.com/com/openattestation/4.0/alpha-context.json"
  ],
  "type": ["VerifiableCredential", "OpenAttestationCredential"],
  "validFrom": "2021-03-08T12:00:00+08:00",
  "name": "Republic of Singapore Driving Licence",
  "issuer": {
    "id": "did:ethr:0xB26B4941941C51a4885E5B7D3A1B861E54405f90",
    "type": "OpenAttestationIssuer",
    "name": "Government Technology Agency of Singapore (GovTech)",
    "identityProof": { "identityProofType": "DNS-DID", "identifier":
```

```

"example.openattestation.com" }
},
"renderMethod": [{
  "id": "https://demo-renderer.opencerts.io",
  "type": "OpenAttestationEmbeddedRenderer",
  "rendererName": "GOVTECH_DEMO"
}],
"credentialSubject": {
  "id": "urn:uuid:a013fb9d-bb03-4056-b696-05575ecef42",
  "type": ["DriversLicense"],
  "name": "John Doe",
  "address": "123 Choa Chu Kang Road, Singapore 123456",
  "licenses": [
    {
      "class": "3",
      "description": "Motor cars with unladen weight <= 3000kg",
      "effectiveDate": "2013-05-16T00:00:00+08:00"
    },
    {
      "class": "3A",
      "description": "Motor cars with unladen weight <= 3000kg",
      "effectiveDate": "2013-05-16T00:00:00+08:00"
    }
  ]
}
}
}

```

The Template Renderer can support different templates that can provide different views of the verifiable credentials. Below are two renderings of the same verifiable credential using different templates.



Actions API

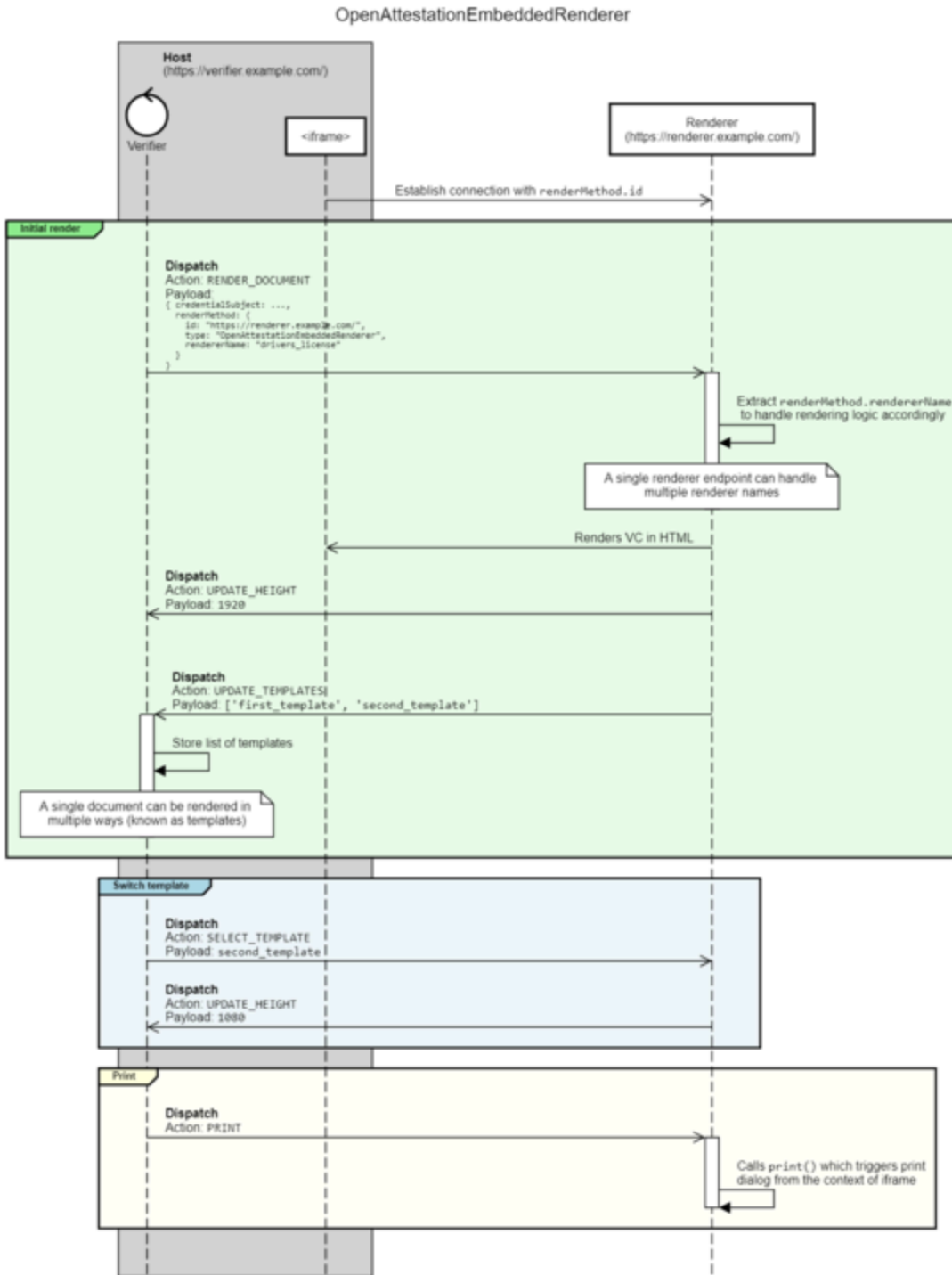
The Host is the application that displays the document with the help of the Template Renderer. The Template Renderer is a web application embedded within an iframe specified by `renderMethod.id`. It communicates with the Host application using the [postMessage API](#) to perform actions.

All actions follow the same structure. They are composed of `type` and `payload`:

- `type` indicates the kind of action being executed, for instance, `RENDER_DOCUMENT` means rendering a document. The type of an action is mandatory.
- `payload` indicates optional data associated to the type, for instance, the content of the document to render.

An open source reference implementation is available on [Github](#).

The interaction between the Host and the Template Renderer is illustrated in the following diagram.



Host-to-Frame Actions

When the iframe is first displayed, the host sends commands to the iframe to render the document. The 4 types of actions supported are described below.

Type	Payload	Action
GET_TEMPLATES	{	Obtain a list of templates supported

Type	Payload	Action
	<pre> type: "GET_TEMPLATES", } </pre>	by the renderer for the given document. The list of templates is returned from UPDATE_TEMPLATES call from the iframe.
SELECT_TEMPLATE	<pre> { type: "SELECT_TEMPLATE", payload: "CUSTOM_TEMPLATE", } </pre>	Select the template to be used for rendering. It should be from the list returned by UPDATE_TEMPLATES. If not found, a default template is used.
RENDER_DOCUMENT	<pre> { type: "RENDER_DOCUMENT", payload: { document: { credentialSubject: ..., renderMethod: ..., ... }, }, } </pre>	Render the verifiable credential inside the IFRAME using the selected template. Document is JSON object form of the verifiable credential.
PRINT	<pre> { type: "PRINT", } </pre>	Must show the print dialog so the contents of the IFRAME can be printed.

Frame-to-Host Actions

These are used by the iframe to update the host to make adjustments on formatting, or selective redaction.

Type	Payload	Action
OBFUSCATE	<pre> { type: "OBFUSCATE", payload: "a[0].b.c", } </pre>	This is used for selective redaction. The Host is informed of the path of the field that has been obfuscated so that the host can create an updated version of the document with the selected field obfuscated.
UPDATE_HEIGHT	<pre> { type: "UPDATE_HEIGHT", payload: 150, } </pre>	Notify the Host of the height of the embedded iframe in pixels so the Host can adjust the size on the browser.

Type	Payload	Action
	}	
UPDATE_TEMPLATES	<pre>{ type: "UPDATE_TEMPLATES", payload: [{ id: "template1", label: "template1", }, { id: "template2", label: "template2", }] }</pre>	Notify the Host of the list of template names that are usable from RENDER_METHOD, or GET_TEMPLATES calls.

References

- Github repo for [OpenAttestation decentralized renderer components](#) - the iframe portion for verifiers to use.
- Github repo for [OpenAttestation decentralized renderer](#) - the react server that determines what templates look like.
- [Guide on creating OpenAttestation decentralized renderer.](#)
- [OA Gallery with multiple different sample documents pointing to different renderers.](#)