Rendering Verifiable Credentials

By Manu Sporny <msporny@digitalbazaar.com>

The Verifiable Credentials ecosystem is experiencing increasing adoption in a variety of markets such as education, supply chain, retail, banking and finance, workforce training, and corporate and government identification cards. Each of these markets have issued credentials for hundreds of years and have pre-conceived notions around what their credentials should look like. Similarly, it is important to understand that visually representing a Verifiable Credential could accidentally exclude those with accessibility needs. We need to consider people with sight needs such as larger font sizes, or the need to use colors that do not create challenges for those that are color blind. We also need to design for those that cannot see, how do they navigate digital wallets and use Verifiable Credentials?

This paper explores ways in which the Verifiable Credentials data model could be extended to support visual, audio, and physical renderings for Verifiable Credentials.

The `render` Property

This paper proposes a new property that can be associated with a Verifiable Credential called the `render`. A rendering hint is a suggestion to a program that processes Verifiable Credentials that a rendering of some sort can be performed by combining the data in the Verifiable Credential with the rendering hint. An example of its usage is shown below:

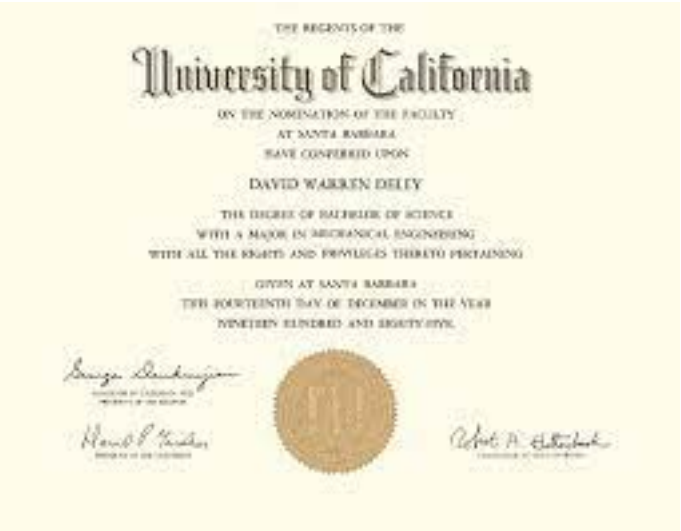
```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": ["VerifiableCredential", "UniversityDegreeCredential"],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "name": "Jane Smith",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts",
      "institution": "Example University"
    }
  },
  // The rendering hint
```

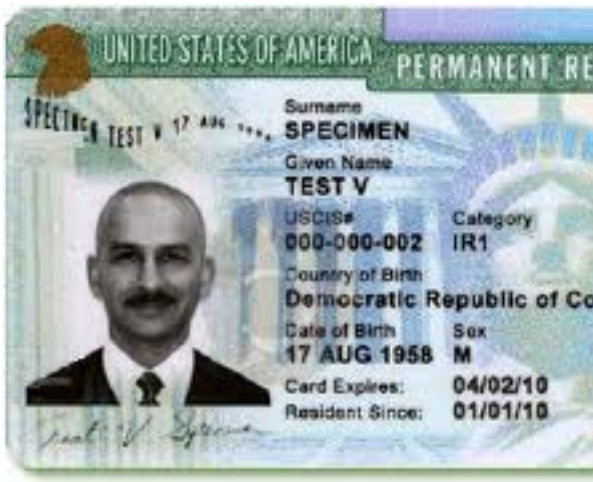
```
  "render": [{
    // An SVG file that can be used to render the credential
    "id": "https://svg.example/degree.svg",
    // The type of rendering hint
    "type": "SvgRenderingHint2022",
    // A multibase-encoded multi-hash of the SVG file
    "digestMultibase": "zQmAPdhyxzznFCwYxAp2dRerWC85Wg6wFl9G270iEu5h6JqW"
  }]
  "proof": { ... }
}
```

Visual Rendering

Visually rendering a Verifiable Credential could be done in a variety of ways. For example, a static bitmap image could be directly embedded or referenced. A templated-SVG file could be directly embedded or referenced. A list of properties that are important to display, along with their priorities for display could be included but without specific rendering instructions beyond that. A dynamic rendering of a subset of the data could be provided either statically or algorithmically (e.g., convert the entire VC to a CBOR-LD-encoded QR Code).

There are many types of visual rendering that issuers desire. Some of these visual representations are shown below:





It is also important to balance the variety of ways in which to visually render a Verifiable Credential with the implementation complexity of providing too many mechanisms.

For this reason, it is suggested that only one mechanism is provided that maximizes the number of use cases that can be achieved: A templated SVG format.

Given the following example instantiation of the format:

```
// The rendering hint
"render": [{
  // An SVG file that can be used to render the credential
  "id": "https://svg.example/degree.svg",
  // The type of rendering hint
  "type": "SvgRenderingTemplate2022",
  // A multibase-encoded multi-hash of the SVG file
  "digestMultibase": "zQmAPdhyxzznFCwYxAp2dRerWC85Wg6wFl9G270iEu5h6JqW"
}]
```

A subset of the Handlebars format is suggested for use in the associated SVG file, due to its simplicity. The JSONPath format is utilized

Use of that format in an SVG file is provided as an example below:

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg">
  <rect width="300" height="100"
    style="fill:rgb(255,255,255);stroke-width:4;stroke:rgb(0,0,0)" />
  <text x="150" y="25" font-size="12" text-anchor="middle" fill="black">
    This {{$.credentialSubject.degree.name}} is conferred to
  </text>
  <text x="150" y="50" font-size="16" text-anchor="middle" fill="black">
    {{$.credentialSubject.name}}
  </text>
  <text x="150" y="75" font-size="12" text-anchor="middle" fill="black">
    by {{$.credentialSubject.degree.institution}}.
```

```
  </text>
</svg>
```

When rendered, the following visual representation will be generated:

Figure 1: A visual depiction of the SVG image above

Clearly, more elaborate renderings can be created by graphical illustrators. It is also important to note that bitmap images can be embedded in SVG files, thus there might not need to be a visual format other than SVG.

Items for further consideration at RWoT 11 include:

- Is embedding an SVG document in a Verifiable Credential rendering a desired feature?
- Are there other forms of visual rendering that are not accomplished via this approach?
- Are there alternatives to the Handlebar template language? What subset is appropriate for us?
- Should another graphical format other than SVG be considered?
- Should ARIA be included in the SVG file?
- Should an "action label" be used to render buttons the individual can press such as "View Certificate" or "Present QR Code"?

Audio Rendering

Visual rendering might not always be suitable for all people or scenarios. At times, it might be useful to provide audio-based rendering. Consider the following rendering instruction:

```
"render": [{
  // An rendering hint
  "type": "AudioRendering2022",
  "description": "This Bachelor of Science and Arts degree is
                  conferred to Jane Smith by Example University."
}]
```

Items for further consideration at RWoT 11 include:

- Should a phoneme-based mechanism, such as the Speech Synthesis Markup Language (SSML) be provided to ensure proper pronunciation of the text content?
- The author of this paper has very minimal accessibility training and requires expert assistance in order to ensure that the proper considerations are made for an audio-based rendering mechanism for Verifiable Credentials.

Physical Rendering

There are times when both visual and audio rendering are not possible. For these scenarios, a braille-based rendering might be appropriate.Consider the following rendering instruction:

```
"render": [{
  // An rendering hint
  "type": "BrailleRendering2022",
  "description": ",? ,ba*elor ( ,sci;e & ,>ts degree is 3f}r$ 6,jane ,smi? 0,example ,univ
}]
```

Items for further consideration at RWoT 11 include:

- The author of this paper has very minimal accessibility training and requires expert assistance in order to ensure that the proper considerations are made for a braille-based rendering mechanism for Verifiable Credentials.

Collaboration at and Beyond RWoT 11

The author of this paper seeks individuals that are interested in rendering Verifiable credentials in many forms. It would be good to explore more use cases such as the display of 1-D barcodes (PDF417 data), 2D barcodes (QR Codes), optically scanned data (MRZ), and other interaction patterns used by people that need to exchange credential information..

The team that furthers this work will also need to coordinate with organizations such as the Web Accessibility Initiative (WAI) at the World Wide Web Consortium as well as other national and international bodies that focus on ensuring that people with accessibility needs are not excluded from using technologies such as Verifiable Credentials.