

Workgroup: None
Internet-Draft: draft-fossati-svg-test-01
Published: 10 March 2020
Intended Status: Experimental
Expires: 11 September 2020
Authors: T. Fossati

SVG Tests

Abstract

This memo is for experimenting with SVG in the context of RFC production.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 September 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#)
 - [1.1. Conventions used in this document](#)
- [2. Code Layout](#)
- [3. From ASCII art to SVG](#)
- [4. Building the XML](#)
- [5. Examples](#)
 - [5.1. A Sequence Diagram](#)
 - [5.2. Lots of Boxes and Arrows](#)
 - [5.3. Even More Boxes and Numbered Arrows](#)
 - [5.4. And Another One](#)
- [6. IANA Considerations](#)
- [7. Security Considerations](#)
- [8. Acknowledgments](#)
- [9. Normative References](#)

[Author's Address](#)

1. Introduction

This memo is for experimenting with SVG in the context of RFC production.

This document assumes a kramdown-rfc2629 based editing flow.

1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

2. Code Layout

The code is structured as follows:

```

├─ Makefile
├─ art
│   ├── cdni-delegation.ascii-art
│   ├── cdni-dns-redirection.ascii-art
│   ├── e2e-flow.ascii-art
│   ├── stir-delegation.ascii-art
│   └─ stir-delegation.svg
└─ draft-fossati-svg-test.md

```

In particular, the art directory contains the diagrams in ASCII art.

3. From ASCII art to SVG

The Makefile contains bunch of variables and a pattern rule to deal with automatic generation of SVG from ASCII using a Golang tool called goat. Another tool, svgcheck, is used to make sure xml2rfc will like the SVG.

The "art" variables:

```

art_src := $(wildcard $(art_dir)/*.ascii-art)
art_svg := $(art_src:.ascii-art=.svg)

```

The pattern rule used to transform each and every ASCII
art into SVG:

```

%.svg: %.ascii-art
    @$(goat) $< | $(svgcheck) -r -o $@ 2>/dev/null || true

```

To install goat and svgcheck, do:

```

$ go get github.com/blampe/goat
$ pip install svgcheck

```

4. Building the XML

The Submit tool on the Datatracker wants the submitted XML to be self contained.

To inline the diagrams you need to do the following:

```

$ kramdown-rfc2629 --v3 \
    draft-fossati-svg-test.md > draft-fossati-svg-test.xml
$ xml2rfc --v3 --preptool draft-fossati-svg-test.xml
$ xml2rfc --v3 --expand draft-fossati-svg-test.prepped.xml
$ mv draft-fossati-svg-test.prepped.exp.xml \
    draft-fossati-svg-test.xml
$ rm -f draft-fossati-svg-test.prepped.xml

```

The "prepped" and "expanded" draft-fossati-svg-test.xml inlines both the ASCII and the SVG in the artset and is ready for submission.

Of course, from there you can also do the usual TXT / HTML generation:

```
$ xml2rfc --v3 draft-fossati-svg-test.xml \  
    draft-fossati-svg-test.txt  
$ xml2rfc --v3 --html draft-fossati-svg-test.xml \  
    draft-fossati-svg-test.html
```

5. Examples

5.1. A Sequence Diagram

kramdown does not support artset natively. So the artset must be inserted using native xml2rfc syntax. The SVG is included in artwork as a local file. The SVG file is created from its ASCII art equivalent as explained in [Section 3](#).

```
<t>  
  <figure anchor="fig-endtoend"  
    title="End to end STAR delegation flow">  
    <artset>  
      <artwork type="ascii-art"  
        src="art/e2e-flow.ascii-art" />  
      <artwork type="svg"  
        src="art/e2e-flow.svg" />  
    </artset>  
  </figure>  
</t>
```

The result is shown in [Figure 1](#).

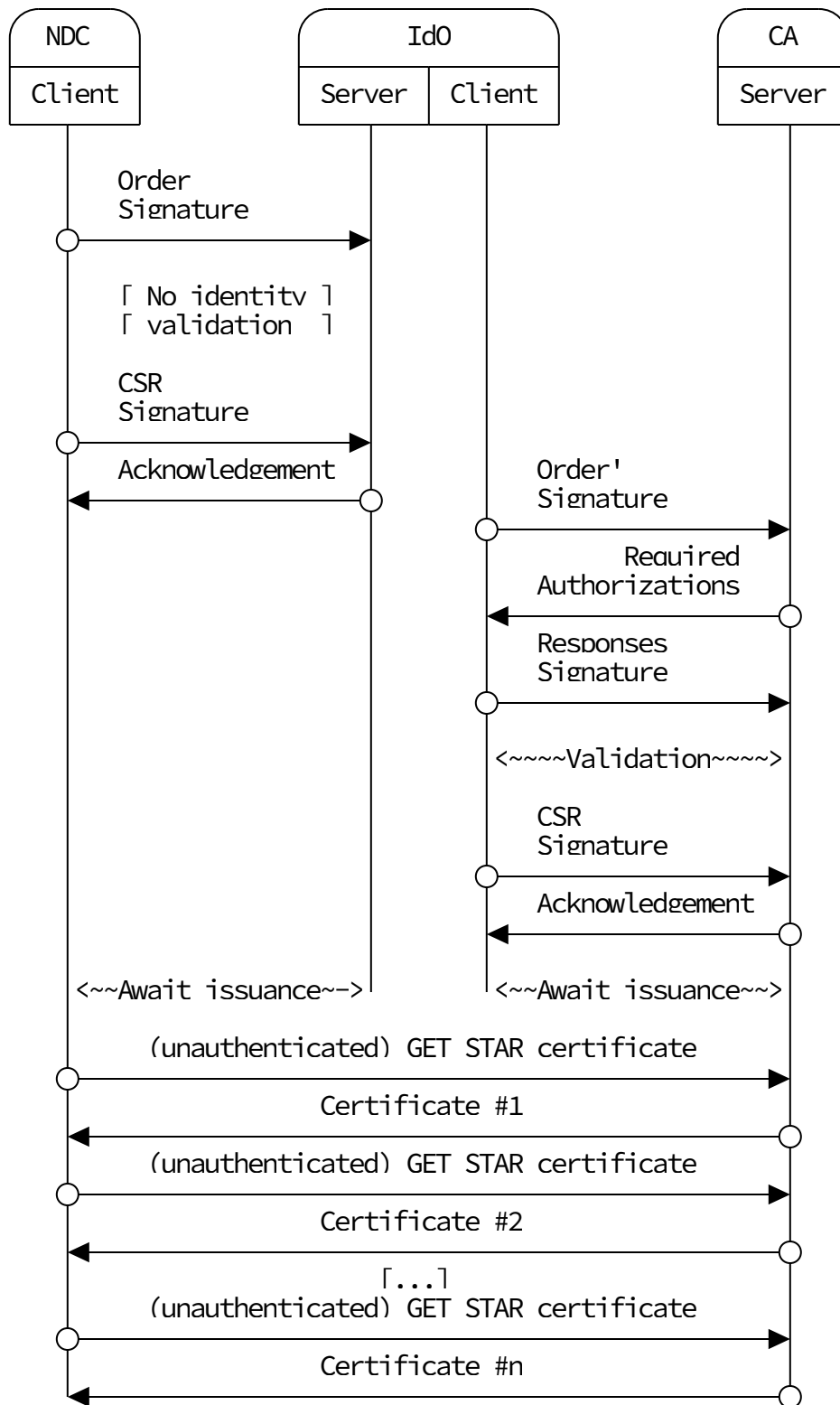


Figure 1: End to end STAR delegation flow

5.2. Lots of Boxes and Arrows

```
<t>
  <figure anchor="fig-cdni-dns-redirection"
    title="DNS Redirection">
    <artset>
      <artwork type="ascii-art"
        src="art/cdni-dns-redirection.ascii-art" />
      <artwork type="svg"
        src="art/cdni-dns-redirection.svg" />
    </artset>
  </figure>
</t>
```

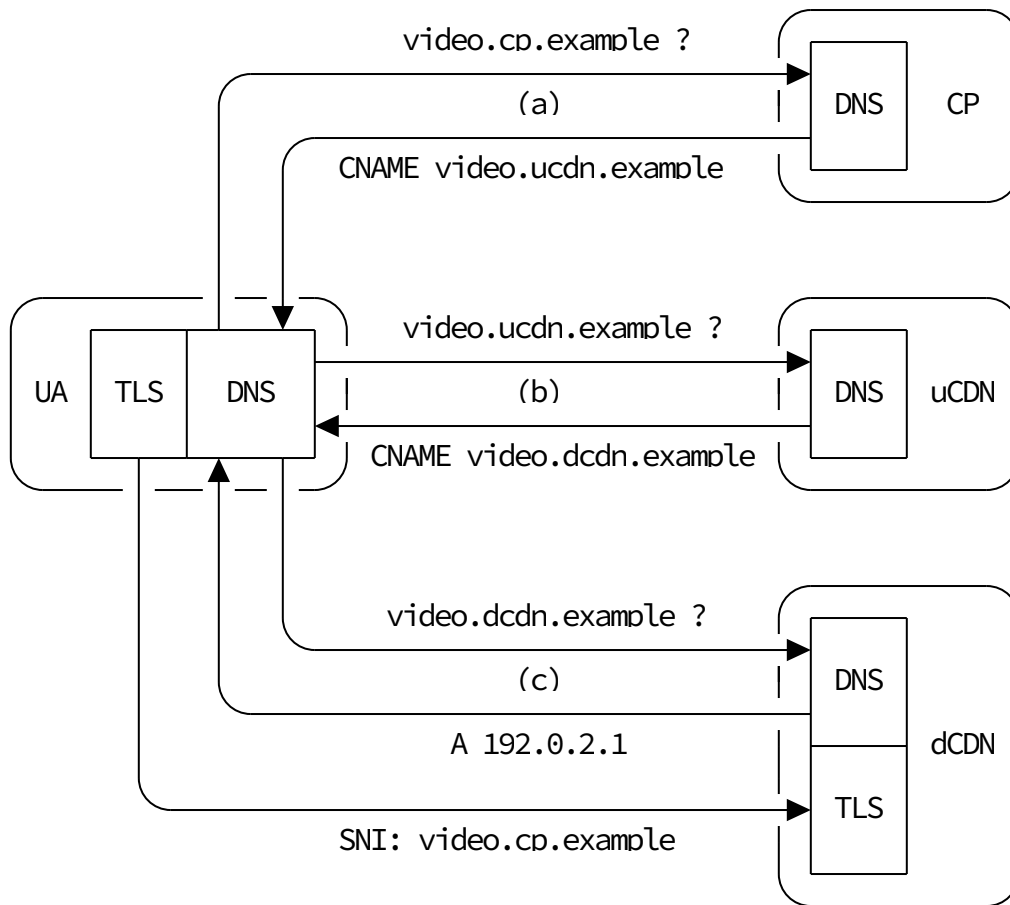


Figure 2: DNS Redirection

5.3. Even More Boxes and Numbered Arrows

```

<t>
  <figure anchor="fig-cdni-flow"
    title="Two levels delegation in CDNI">
    <artset>
      <artwork type="ascii-art"
        src="art/cdni-delegation.ascii-art" />
      <artwork type="svg"
        src="art/cdni-delegation.svg" />
    </artset>
  </figure>
</t>

```

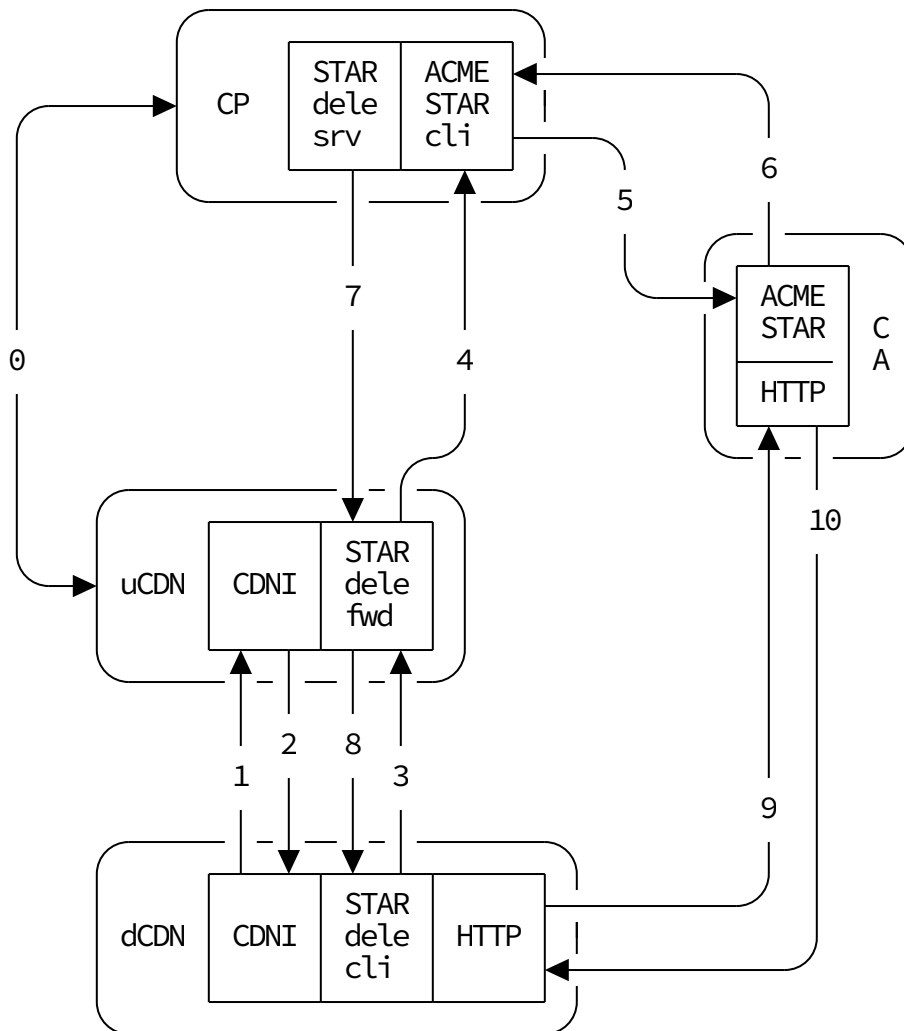


Figure 3: Two levels delegation in CDNI

5.4. And Another One

```
<t>
  <figure anchor="fig-stir-flow"
    title="Delegation in STIR">
    <artset>
      <artwork type="ascii-art"
        src="art/stir-delegation.ascii-art" />
      <artwork type="svg"
        src="art/stir-delegation.svg" />
    </artset>
  </figure>
</t>
```

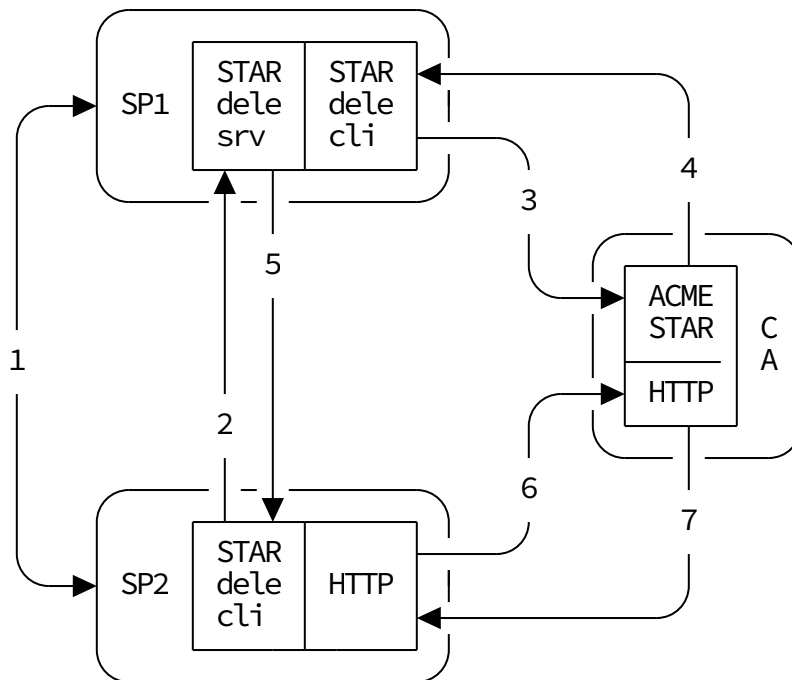


Figure 4: Delegation in STIR

6. IANA Considerations

No requests are made to IANA.

7. Security Considerations

There are none.

8. Acknowledgments

Yaron for pointing out the current limitations in the tooling and providing the workaround.

9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Author's Address

Thomas Fossati

Email: thomas.fossati@arm.com