8 Appendix A – Using SDP for the Session Description

JSONverse needs some means to describe 3D Multiuser Sessions, such that a user, who posesses the Session Description (SD) and an Access Token, can easily access the Multiuser Session.

I did a small research in the IETF RFCs that describe the SDP and came to following conclusion: almost everything that is needed for the description of 3D Multiuser Sessions, is already in place. Take e.g. the following example:

```
v=0
o=yottzumm 123456789 123459999 IN IP4 lc-soc-lc.at
s=group1-petname
i=The Review of the Rose, Session#1
c=IN IP4 lc-soc-lc.at
t=3944246340  3944249940
m=application 8443 TCP/WSS/C3P *
a=setup:passive
a=connection:new
a=websocket-uri:wss://lc-soc-lc.at:8443/yottzumm/123456789
a=c3p-object:obj1
a=c3p-object:obj2
```

All these syntaxes are already specified in several RFCs, which are listed below, just

- a new proto field TCP/WSS/C3P (or TCP/WS/C3P) for the m-line and
- a new media-level attribute c3p-object

need to be defined specifically for our use case.

8.1 RFC 8866 – SDP: Session Description Protocol

The syntax is basically defined in RFC 8866, follow some citings from RFC 8866:

> When initiating multimedia teleconferences, voice-over-IP calls, streaming video, or other sessions, there is a requirement to convey media details, transport addresses, and other session description metadata to the participants.
>
> SDP provides a standard representation for such information, irrespective of how that information is transported. SDP is purely a format for session description -- it does not incorporate a transport protocol, and it is intended to use different transport protocols as appropriate, including the Session Announcement Protocol (SAP) [RFC2974], Session Initiation Protocol (SIP) [RFC3261], Real-Time Streaming Protocol (RTSP) [RFC7826], electronic mail [RFC5322] using the MIME extensions [RFC2045], and the Hypertext Transport Protocol (HTTP) [RFC7230].
>
> **SDP is intended to be general purpose** so that it can be used in a wide range of network environments and applications. **However, it is not intended to support negotiation of session content or media encodings**: this is viewed as outside the scope of session description.

So, our use case is the session announcement, where ONE session description is sent unchanged to all potential participants of the session.

The Session Description (SD) shall contain ALL of the information that is necessary to

A. download the scene and

B. connect the scene to the X3D Collaboration Server for the synchronization of the shared state and for the distribution of events.

As an exception to this idea, the access tokens (group access token GAT, session access token SAT, personal access token PAT) shall be transported outside of the session description.

Following additional citings:

```
Some lines in each description are required and some are optional,
but when present, they must appear in exactly the order given here.
(The fixed order greatly enhances error detection and allows for a
simple parser).  In the following overview, optional items are marked
with a "*".

   Session description
      v=  (protocol version)
      o=  (originator and session identifier)
      s=  (session name)
      i=* (session information)
      u=* (URI of description)
      e=* (email address)
      p=* (phone number)
      c=* (connection information -- not required if included in
           all media descriptions)
      b=* (zero or more bandwidth information lines)
      One or more time descriptions:
        ("t=", "r=" and "z=" lines; see below)
      k=* (obsolete)
      a=* (zero or more session attribute lines)
      Zero or more media descriptions

   Time description
      t=  (time the session is active)
      r=* (zero or more repeat times)
      z=* (optional time zone offset line)

   Media description, if present
      m=  (media name and transport address)
      i=* (media title)
      c=* (connection information -- optional if included at
           session level)
      b=* (zero or more bandwidth information lines)
      k=* (obsolete)
      a=* (zero or more media attribute lines)
```

The v-line

```
      v=0

The "v=" line (version-field) gives the version of the Session
Description Protocol.  This memo defines version 0.  There is no
minor version number.
```

The o-line

```
o=<username> <sess-id> <sess-version> <nettype> <addrtype>
<unicast-address>
```

The "o=" line (origin-field) gives the originator of the session (her username and the address of the user's host) plus a session identifier and version number:

<username> is the user's login on the originating host, or it is "-" if the originating host does not support the concept of user IDs. The <username> MUST NOT contain spaces.

<sess-id> is a numeric string such that the tuple of <username>, <sess-id>, <nettype>, <addrtype>, and <unicast-address> forms a globally unique identifier for the session. The method of <sess-id> allocation is up to the creating tool, but a timestamp, in seconds since January 1, 1900 UTC, is recommended to ensure uniqueness.

<sess-version> is a version number for this session description. Its usage is up to the creating tool, so long as <sess-version> is increased when a modification is made to the session description. Again, as with <sess-id> it is RECOMMENDED that a timestamp be used.

<nettype> is a text string giving the type of network. Initially, "IN" is defined to have the meaning "Internet", but other values MAY be registered in the future (see Section 8).

<addrtype> is a text string giving the type of the address that follows. Initially, "IP4" and "IP6" are defined, but other values MAY be registered in the future (see Section 8).

<unicast-address> is an address of the machine from which the session was created. For an address type of "IP4", this is either a fully qualified domain name of the machine or the dotted-decimal representation of an IP version 4 address of the machine. For an address type of "IP6", this is either a fully qualified domain name of the machine or the address of the machine represented as specified in Section 4 of [RFC5952]. For both "IP4" and "IP6", the fully qualified domain name is the form that SHOULD be given unless this is unavailable, in which case a globally unique address MAY be substituted.

In general, the "o=" line serves as a globally unique identifier for this version of the session description, and the subfields excepting the version, taken together identify the session irrespective of any modifications.

The s-, i- and c-line

```
s=<session name>
```

The "s=" line (session-name-field) is the textual session name. There MUST be one and only one "s=" line per session description. The "s=" line MUST NOT be empty. If a session has no meaningful name, then "s= " or "s=-" (i.e., a single space or dash as the session name) is RECOMMENDED. If a session-level "a=charset:" attribute is present, it specifies the character set used in the "s=" field. If a session-level "a=charset:" attribute is not present, the "s=" field MUST contain ISO 10646 characters in UTF-8 encoding.

```
i=<session information>
```

The "i=" line (information-field) provides textual information about the session. There can be at most one session-level "i=" line per session description, and at most one "i=" line in each media description. Unless a media-level "i=" line is provided, the session-level "i=" line applies to that media description. If the "a=charset:" attribute is present, it specifies the character set used in the "i=" line. If the "a=charset:" attribute is not present, the "i=" line MUST contain ISO 10646 characters in UTF-8 encoding.

......

The "i=" line is intended to provide a free-form human-readable description of the session or the purpose of a media stream. It is not suitable for parsing by automata.

......

```
c=<nettype> <addrtype> <connection-address>
```

The "c=" line (connection-field) contains information necessary to establish a network connection.

A session description MUST contain either at least one "c=" line in each media description or a single "c=" line at the session level. It MAY contain a single session-level "c=" line and additional media-level "c=" line(s) per-media-description, in which case the media-level values override the session-level settings for the respective media.

Attributes

```
a=<attribute-name>
a=<attribute-name>:<attribute-value>
```

Attributes are the primary means for extending SDP. Attributes may be defined to be used as session-level attributes, media-level attributes, or both. (Attribute scopes in addition to media-level and session-level scopes may also be defined in extensions to this document, e.g., [RFC5576] and [RFC8864].)

A media description may contain any number of "a=" lines (attribute-fields) that are media description specific. These are referred to as media-level attributes and add information about the media description. Attribute-fields can also be added before the first media description; these session-level attributes convey additional information that applies to the session as a whole rather than to individual media descriptions.

The time description

```
t=<start-time> <stop-time>
```

A "t=" line (time-field) begins a time description that specifies the start and stop times for a session. Multiple time descriptions MAY be used if a session is active at multiple irregularly spaced times; each additional time description specifies additional periods of time for which the session will be active. If the session is active at regular repeat times, a repeat description, begun by an "r=" line (see Section 5.10) can be included following the time-field -- in which case the time-field specifies the start and stop times of the entire repeat sequence.

The following example specifies two active intervals:

```
t=3724394400 3724398000 ; Mon 8-Jan-2018 10:00-11:00 UTC
t=3724484400 3724488000 ; Tue 9-Jan-2018 11:00-12:00 UTC
```

The first and second subfields of the time-field give the start and stop times, respectively, for the session. These are the decimal representation of time values in seconds since January 1, 1900 UTC. To convert these values to Unix time (UTC), subtract decimal 2208988800.

Some time representations will wrap in the year 2036. Because SDP uses an arbitrary length decimal representation, it does not have this issue. Implementations of SDP need to be prepared to handle these larger values.

If the <stop-time> is set to zero, then the session is not bounded, though it will not become active until after the <start-time>. If the <start-time> is also zero, the session is regarded as permanent.

User interfaces SHOULD strongly discourage the creation of unbounded and permanent sessions as they give no information about when the session is actually going to terminate, and so make scheduling difficult.

Media Descriptions

```
m=<media> <port> <proto> <fmt> ...
```

A session description may contain a number of media descriptions. Each media description starts with an "m=" line (media-field) and is terminated by either the next "m=" line or by the end of the session description. A media-field has several subfields:

<media> is the media type. This document defines the values "audio", "video", "text", "application", and "message". This list is extended by other memos and may be further extended by additional memos registering media types in the future (see Section 8). For example, [RFC6466] defined the "image" media type.

<port> is the transport port to which the media stream is sent. The meaning of the transport port depends on the network being used as specified in the relevant "c=" line, and on the transport protocol defined in the <proto> subfield of the media-field. Other ports used by the media application (such as the RTP Control Protocol (RTCP) port [RFC3550]) MAY be derived algorithmically from the base media port or MAY be specified in a separate attribute (for example, the "a=rtcp:" attribute as defined in [RFC3605]).

......

<proto> is the transport protocol. The meaning of the transport protocol is dependent on the address type subfield in the relevant "c=" line. Thus a "c=" line with an address type of "IP4" indicates that the transport protocol runs over IPv4. The following transport protocols are defined, but may be extended through registration of new protocols with IANA (see Section 8):

* udp: denotes that the data is transported directly in UDP with no additional framing.

* RTP/AVP: denotes RTP [RFC3550] used under the RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551] running over UDP.

* RTP/SAVP: denotes the Secure Real-time Transport Protocol [RFC3711] running over UDP.

* RTP/SAVPF: denotes SRTP with the Extended SRTP Profile for RTCP-Based Feedback [RFC5124] running over UDP.

The main reason to specify the transport protocol in addition to the media format is that the same standard media formats may be carried over different transport protocols even when the network protocol is the same -- a historical example is vat (MBone's popular multimedia audio tool) Pulse Code Modulation (PCM) audio and RTP PCM audio; another might be TCP/RTP PCM audio. In addition, relays and monitoring tools that are transport-protocol-specific but format-independent are possible.

<fmt> is a media format description. The fourth and any subsequent subfields describe the format of the media. The interpretation of the media format depends on the value of the <proto> subfield.

We use a media description with the values

- <media> = application
- <proto> = TCP/WS/C3P or <proto> = TCP/WSS/C3P
- <fmt> = *

This decision is based on following considerations.

1. The implementation of the JSONverse is based on socket.io, which uses WebSocket transport
2. A similar media type has been defined in RFC 8856 - Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams and RFC 8857 - Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
   a) <media> = application shall be used
   b) RFC 8857 defines <proto> = TCP/WS/BFCP or <proto> = TCP/WSS/BFCP
   c) <fmt> = *
3. We call our WebSocket sub-protocol the C3P = Collaborative 3D Profile

8.2 RFC 4145 – TCP-Based Media Transport in the Session Description Protocol (SDP)

The attributes connection and setup are defined in this RFC. Follow some citings from RFC 4145

```
      setup-attr           =  "a=setup:" role
      role                 =  "active" / "passive" / "actpass"
                              / "holdconn"

   'active': The endpoint will initiate an outgoing connection.

   'passive': The endpoint will accept an incoming connection.

   'actpass': The endpoint is willing to accept an incoming
   connection or to initiate an outgoing connection.

   'holdconn': The endpoint does not want the connection to be
   established for the time being.
```

......

```
      connection-attr        = "a=connection:" conn-value
      conn-value             = "new" / "existing"
```

8.3 RFC 8124 - The Session Description Protocol (SDP) WebSocket Connection URI Attribute

The websocket-uri attribute is defined in RFC 8124, follows a citing:

```
This section defines a new SDP media-level attribute, "websocket-
uri", which can appear in any of the media sections.

Example:

   a=websocket-uri:wss://example.com/chat

Where "wss://example.com/chat" is the ws-URI defined in Section 3 of
[RFC6455].

When the "websocket-uri" attribute is present in the media section of
the SDP, the IP address in "c=" line SHALL be ignored and the full
URI SHALL be used instead to open the WebSocket connection.  The
clients MUST ensure that they use the URI to open the WebSocket
connection and ignore the IP address in the "c=" line and the port in
the "m=" line.
```