BBC
Design + Engineering

TTML LIVE CONTRIBUTION

PLATFORM MEDIA SERVICES

SCOPE

SCOPE

WHICH CLIENT?

Normal playback clients that the audience use in general **don't need to know** if the content is live or pre-recorded. They just receive media including audio, video and subtitles with enough timing information to play them back in a synchronised way.

The media might be delivered in a stream or a download. This allows support for pause, rewind etc.

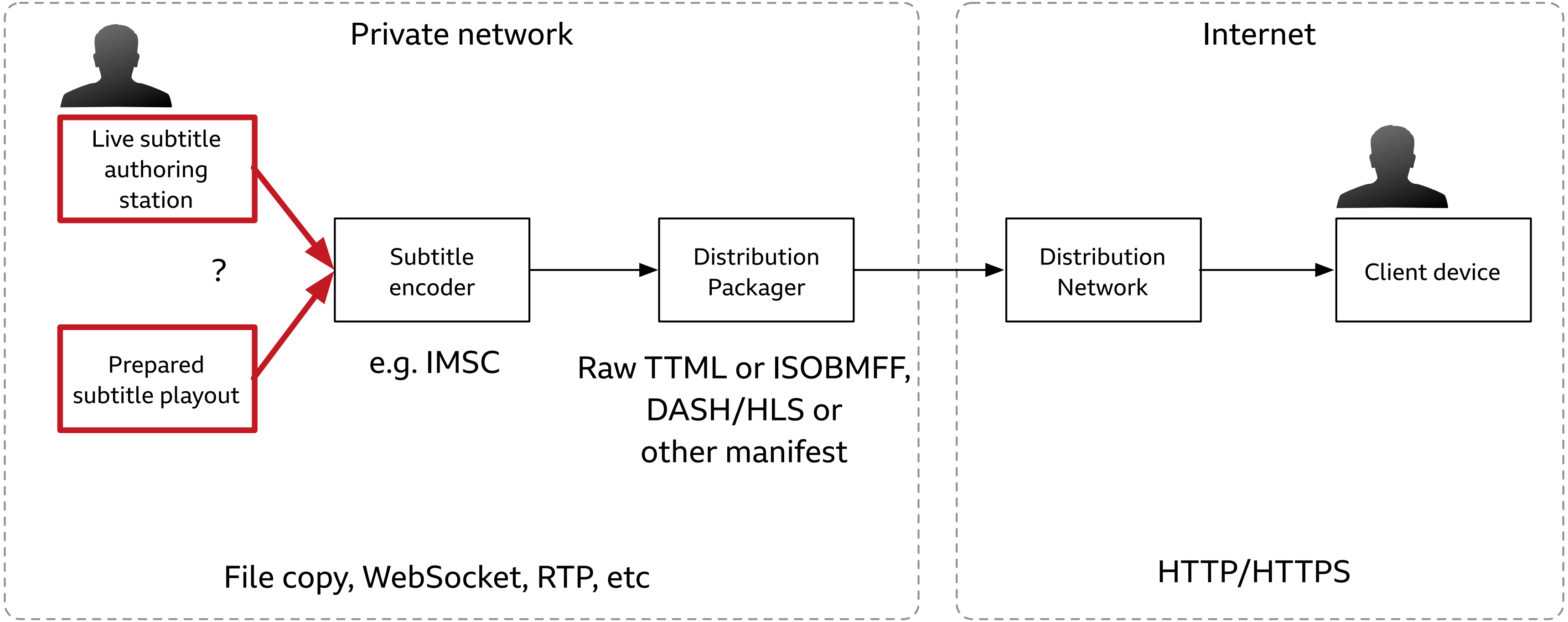
The problem space here is upstream:

How does the subtitle author or broadcast playout equipment contribute live subtitles to an emission encoder?

WebRTC currently offers a model where everything is “do it now” but even in that case, the media is in practice delivered in a set of contiguous chunks each of which has a defined duration. Those chunks may appear to be delivered in a continuous stream.

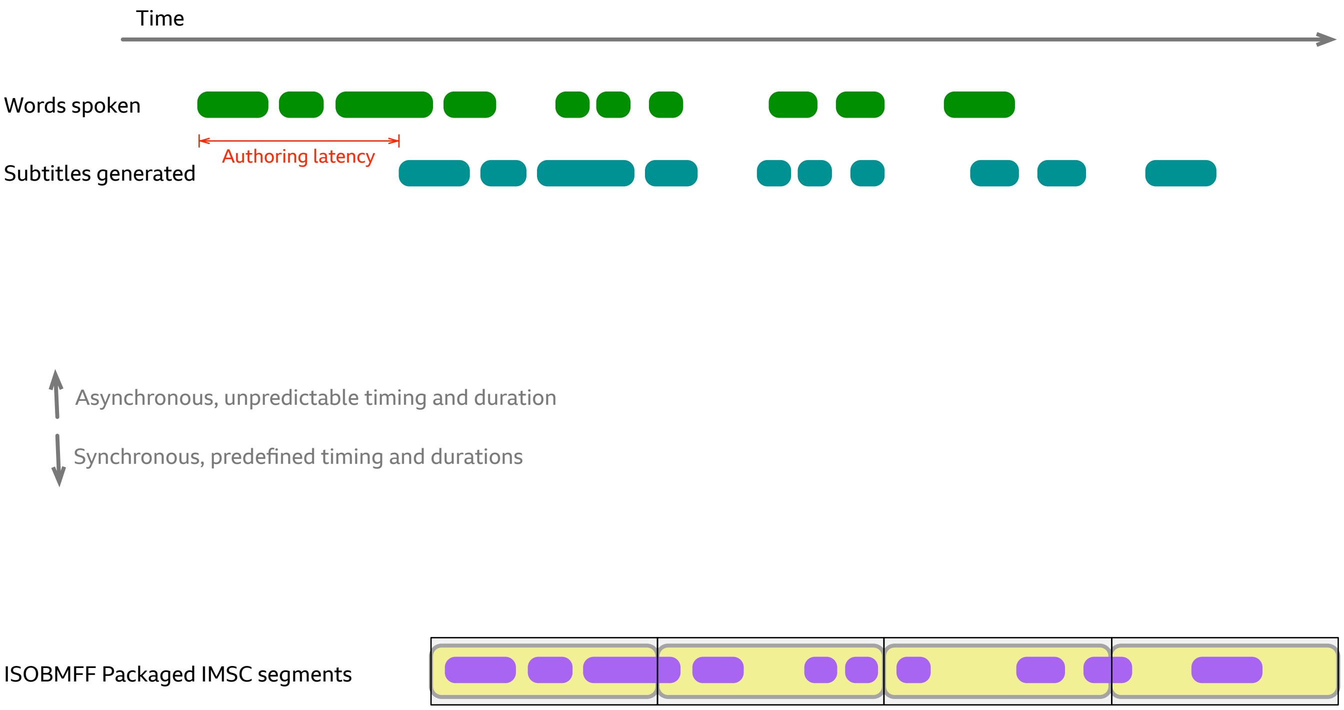
SCOPE

AUTHOR TO AUDIENCE


Private network
Internet
Live subtitle authoring station
?
Prepared subtitle playout
Subtitle encoder
Distribution Packager
Distribution Network
Client device
e.g. IMSC
Raw TTML or ISOBMFF, DASH/HLS or other manifest
File copy, WebSocket, RTP, etc
HTTP/HTTPS

SCOPE

TIMING



Time

Words spoken

Authoring latency

Subtitles generated

Asynchronous, unpredictable timing and duration

Synchronous, predefined timing and durations

ISOBMFF Packaged IMSC segments

SCOPE

PROBLEMS/DESIRED CHARACTERISTICS

- Minimal Latency
- Deliver to encoder asynchronously across a variety of transports
- Defined semantics for transforming asynchronous to synchronous – "resequencing"
- Support none, some or all presentation specification like positioning, styling
- Allow subtitle processing for example to improve timing, positioning, styling, spelling, etc

- Live subtitle authoring is always effectively predictive; predictions are sometimes wrong: it must be possible to correct a previously made prediction.

Considerations with respect to the A/V:

- The subtitle presentation needs to be synchronised against the audio, in the end, or at least be "close enough".
- Typically there is a complete system delay for processing audio and video that is comparable to, or longer than, the subtitle processing delay. However in low latency scenarios the subtitle *authoring* delay may exceed that.

SCOPE

TRANSPORT PROTOCOLS

Simplest transport options offer no timing semantics – delivery as soon as possible.

TCP-based transport guarantees delivery at the potential expense of latency.

UDP-based transport minimises latency at the potential expense of out-of-order or partial delivery.

Both can be mitigated within carefully managed networks.

Examples

WebSocket is TCP-based.

- Easy to set up and use
- Little usage in broadcast industry for contribution of audio and video
- Specifies nothing regarding timing.

RTP is UDP-based

- Requires careful configuration
- Gaining traction in SMPTE 2110 for IP-based contribution of audio and video
- Each packet has a timestamp

EBU-TT LIVE SOLUTION

EBU-TT LIVE SOLUTION

KEY CHARACTERISTICS

EBU-TT Live, Tech3370
https://tech.ebu.ch/publications/tech3370

- Stateless encoder – can begin working as soon as it receives documents
- Subtitles are delivered as a **sequence**
- Each sequence consists of a set of **documents**
- Each document is a TTML document instance with:
  - A **sequence identifier**
  - A **sequence number**
- A **node** is a processor that emits or receives (or both) sequences
- A **stream** is the delivery of a sequence from one node to another node
- Exactly zero or one document is active at any given time
- To resolve which document is active, semantics are defined to calculate the **Document resolved begin time** and the **Document resolved end time**.
- TTML documents individually, in the absence of other processing rules and context, define content from time zero until an unspecified end time.
- EBU-TT Live uses *heuristics* based on the “earliest computed begin time” and “latest computed end time” of a document. Those terms are further defined.
- A precedence rule operates in case a document temporally overlaps another one: the document with the later sequence number wins.
- This means that the document resolved end time can change due to the later availability of a newer document.
- Documents that are not yet available cannot be active.
- The temporal granularity is arbitrary – a variety of *issuing strategies* are possible.
- SMPTE timebase is not supported.

EBU-TT LIVE SOLUTION

WIDER APPLICABILITY

Could the sequence semantics from EBU-TT Live be used for other profiles, such as IMSC?

- Yes, they are orthogonal to any other presentation semantics
- Yes, EBU would be willing to contribute this IPR

Are there any problems?

No blockers; there is one problem in TTML that needs to be worked around:

Asymmetry between definition of the begin time and the end time of a TTML document instance

TTML documents always define begin of the document at time zero.

TTML documents *can* specify an end to defined content, with an implied presentation beyond that of "present nothing". Or, TTML documents can have an undefined, indefinite, end time.

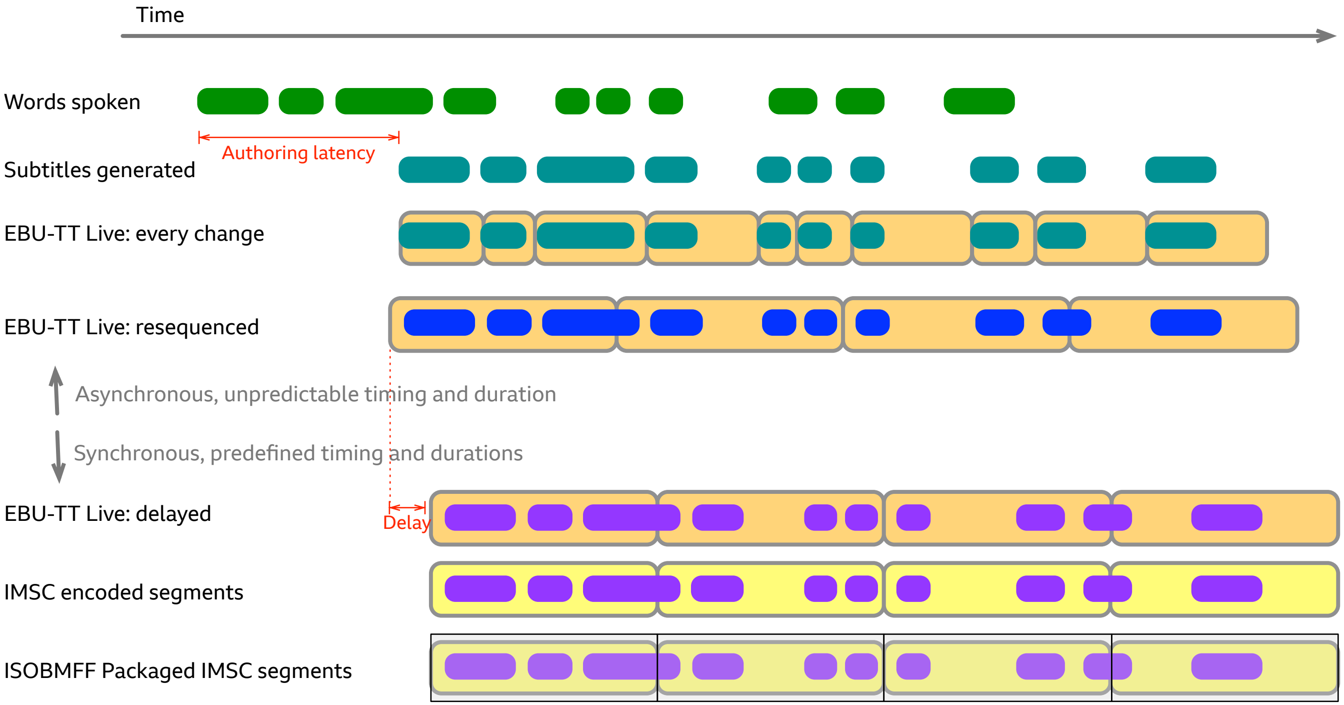
The asymmetry is that it is not possible to define a non-zero begin time but it is possible to define a definite end time (using dur, e.g. on the body element)

EBU-TT Live goes to some lengths to define a heuristic for this. Arguably it would be simpler if every document had root element attributes that simply define the period during which that document is intended to define some behaviour; any content timed outside that interval would be ignored.

That would make downstream computation of begin and end times much simpler.

EBU-TT LIVE SOLUTION

EXAMPLE OF EBU-TT LIVE SOLUTION


Time
Words spoken
Authoring latency
Subtitles generated
EBU-TT Live: every change
EBU-TT Live: resequenced
Asynchronous, unpredictable timing and duration
Synchronous, predefined timing and durations
EBU-TT Live: delayed
Delay
IMSC encoded segments
ISOBMFF Packaged IMSC segments

EBU-TT LIVE SOLUTION

IMPLEMENTATION

The EBU-TT Live semantics were implemented in full as part of the process for validating the EBU-TT Live specification, in the EBU-TT Live Interoperability Toolkit (“LIT”) at http://ebu.github.io/ebu-tt-live-toolkit/

This implements the WebSocket carriage mechanism (EBU Tech3370s1), as well as a file-system based carriage mechanism and a direct in-memory transfer route between nodes.

PROPOSAL AND DISCUSSION

PROPOSAL AND DISCUSSION

PROPOSAL

Produce a new technical report describing the semantics of live contribution of temporally fragmented TTML using:

1. the sequence identifier and sequence number extensions of EBU-TT Live (in their existing namespace); and
2. the document resolved begin and end time calculation algorithms; and
3. the constraint that these calculations do not apply in smpte discontinuous mode (and therefore are best avoided in smpte timebase altogether); and
4. clarify that these semantics are *only* needed when the TTML is not being wrapped in some other thing that already defines timing semantics of each document, e.g. ISOBMFF, Smooth etc.
5. Reference EBU Tech3370 for further discussion and explanations of issuing strategies, the impact of availability time vs media time etc.

This is the minimal set of semantics to define. EBU-TT Live also defines semantics for some processing nodes such as Delay Nodes, Handover Manager Nodes, etc, which can be considered, and are also orthogonal to any presentation semantics other then timing.

PROPOSAL AND DISCUSSION

DISCUSSION

Q&A