# ALTO Incremental Updates Using Server-Sent Events (SSE)

draft-ietf-alto-incr-update-sse-15

W. Roome
**Y. Richard Yang**
Dawn Chen

December 11, 2018

ALTO Interim Meeting

# Updates Summary

- Many small edits to improve the reading, e.g.,
  - Rearrange the order of the terms in Section 2
- Reorganize the sections to make the design easier to follow
- Clarify the overall ALTO SSE architecture
- Better clarification on the event field/id field of SSE
- Updates on Security Considerations to be more complete
- Update on Requirements on Future ALTO Services if using SSE

- Details see:
  https://tools.ietf.org/rfcdiff?url1=https://www.ietf.org/archive/id/draft-ietf-alto-incr-update-sse-13.txt&url2=https://www.ietf.org/archive/id/draft-ietf-alto-incr-update-sse-15.txt

# Re-arranged Sections

# Revised ALTO SSE Architecture

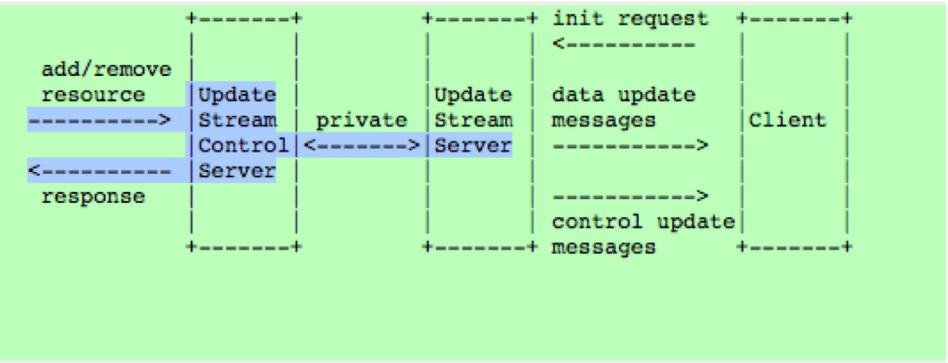


Figure 2: ALTO SSE Overview

In addition to state changes triggered from the update stream server side, in a flexible design, an ALTO client may initiate changes as well, in particular, by adding or removing ALTO resources receiving updates. An ALTO client initiates such changes using the stream control service. For an update stream service supporting update stream control, the update stream server responds by sending an event (a control update message) with the URI of the stream control service. The ALTO client can then use the URI to ask the update stream server to (1) send data update messages for additional resources, (2) stop sending data update messages for previously requested resources, or (3) gracefully stop and close the update stream altogether.

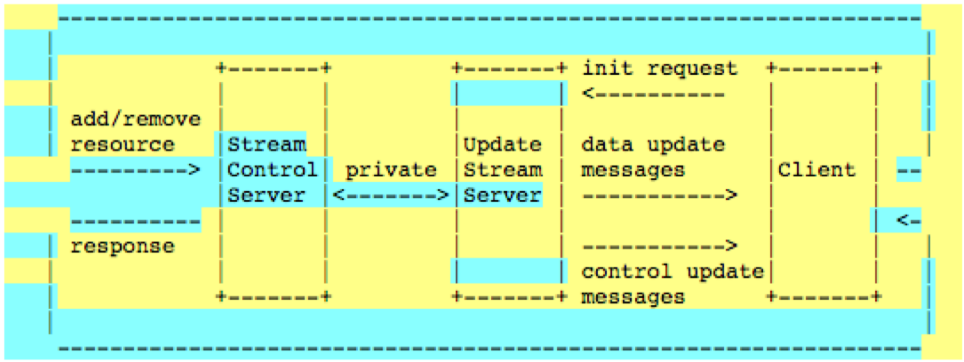


Figure 2: ALTO SSE Architecture.

In addition to control changes triggered from the update stream server side, in a flexible design, an ALTO client may initiate control changes as well, in particular, by adding or removing ALTO resources receiving updates. An ALTO client initiates such changes using the stream control service. Although one may use a design that the client uses the same HTTP connection to send the control requests, it requires stronger server support such as HTTP pipeline. For more flexibility, this document introduces stream control service. In particular, the update stream server of an update stream uses the first message to provide the URI of the stream control service. The ALTO client can then use the URI to ask the update stream server to (1) send data update messages for additional resources, (2) stop sending data update messages for previously requested resources, or (3) gracefully stop and close the update stream altogether. Figure 2 shows the complete ALTO SSE architecture.

# Clarification on event/id in SSE

The two sub-fields (media-type and client-id) of the event field are encoded as comma-separated strings:

media-type [ ',' client-id ]

Note that media-type names may not contain a comma (character code 0x2c). [Dawn: may not or MAY NOT]

Note that an update message does not use the SSE "id" field.

The two sub-fields (media-type and client-id) of the event field are encoded as comma-separated strings:

media-type [ ',' client-id ]

The media-type name used by ALTO SSE MUST NOT contain a comma (character code 0x2c). [TODO: conform to Section 4.2 of [RFC6838]; confirm with IANA.]

Note that an update message does not use the SSE "id" field. [TODO: Discuss to move client-id to SSE id?]

# Security Considerations

- Add common issues

- Organize into 3 types of new security issues
  - Update Stream Server: Denial-of-Service Attacks
  - ALTO Client: Update Overloading or Instability
  - Stream Control: Spoofed Control Requests

# Update on Requirements on Future ALTO Services if using SSE

- Split into two levels
  - At the high level, the key requirements are that (1) each data update message is for a single resource; (2) incremental changes can be applied only to a resource that is a single JSON object, as both JSON merge patch and JSON patch can apply only to a single JSON object. Hence, if a future ALTO resource can contain multiple objects, then either each individual object also has a resource-id or an extension to this design is made.
  - At the low level encoding level, new line in SSE has its own semantics. Hence, the requirement is that the encoding does not include new line that can confuse with SSE encoding. In particular, the data update message MUST NOT include "event: " or "data: " at a new line as part of data message.

# Next Step?

- Feel quite comfortable with design and willing to move forward

# Thank you!