



Subscription Methods

By: Scott Kolodzieski, Architect

All subscriptions are not the same...

Normally, in a stream based system, a client subscription delivers a lossless stream of records at a rate determined by the server's computational capacity. For very fast servers, this rate may be throttled by the bandwidth of the network. Frequently, it is handy to allow several basic modifications to the standard lossless subscription mechanism. While many applications require a lossless stream of events delivered with minimal latency, not all clients can make use of such a data stream. Take for example, a trading dashboard, that will display several charts of market statistics. GUI widgets, that are delivered tens of thousands of updates per second, would not only fail to update in real-time, but would waste valuable network bandwidth. There is no reason for this type of client to receive data at rates that cannot be realistically rendered, or perceived by the human eye.

The Aleri Streaming Platform provides several different subscription modes, allowing individual clients to tune their subscription for optimal application performance. The following modes are supported:

LOSSY – allow older data to be discarded as new data arrives to the gateway. If the Aleri platform is delivering data to the gateway so fast that the internal buffers would overflow, either because the client cannot keep up with the data rate or the network interface does not have the appropriate bandwidth, the oldest data in the output buffer is shed and replaced by the most recent data. A slow client that is operating in LOSSY mode will never affect the overall performance of the Streaming Platform.

PROJECT/SELECT – the subscription data delivered to the client is driven by an SQL statement. Selection could enhance (or prune) the delivered fields for the records (select symbol, price, price*FX from RICS). Projection acts like a typical filter (select * from RICS where symbol='ORCL') . The project/select subscription mode thus allows for further enrichment of events as they leave the Streaming Platform.

DROPPABLE – a client connection that is declared droppable, will have its connection terminated if it cannot keep up with the data delivery rate of the Streaming Platform. This is useful when the client must under all circumstances receive every data record from a stream, but the overall server cannot afford to be slowed if a client falls behind.



PULSED – a pulsed client receives blocks of optimally coalesced changes to a stream every $<n>$ seconds. A pulsed subscription is a perfect solution for driving any client application that is interested in periodically getting a set of updates (for the most recent set of changes) to a stream. While the subscribed client does not receive each event produced by the Streaming Platform, it does receive a sufficient set of events to maintain the full state of the subscribed stream.

Example:

Suppose that a client would like the most recent price updates for the stock symbols ORCL and JAVA delivered every ten seconds, so cells of a spreadsheet representing the position in these two stocks could be displayed. One could accomplish this via a ten second pulsed subscribe driven by the sql statement “select * from spotPrices where (symbol='ORCL') or (symbol='JAVA')”. Even if each of these symbols moves many hundreds or thousands of times in a 10 second window, the client would receive at most 2 records (one for each symbol), every ten seconds.

UNITED STATES

Two Prudential Plaza, 41st Floor
Chicago, IL 60601
t +1 312 540 0100

430 Mountain Ave.
Murray Hill, NJ 07974
t +1 908 673 2400

UNITED KINGDOM

New Baltic House
65 Fenchurch Street
London EC3M 4BE
United Kingdom
t +44 (0) 20 7821 1110