Research Interests

Aslak Johansen
Software Engineering Section, University of Southern Denmark

Introduction

Systems work involving streaming data with requirements on latency and availability.

Relevant qualities:

- ***Concurrency*** The ability to allow work to be defined in units that can be executed in parallel (concurrency models).
- ***Latency*** The ability to provide robust low latency (e.g, by bounding concurrency).
- ***Availability*** The ability to provide service despite the occurrence of unforeseen events across the lifetime of the deployment (gracefull degradation, hot updates, distribution).

Interesting aspects:

- Metadata.
- Query languages and interaction models.
- In-network processing.

Problem

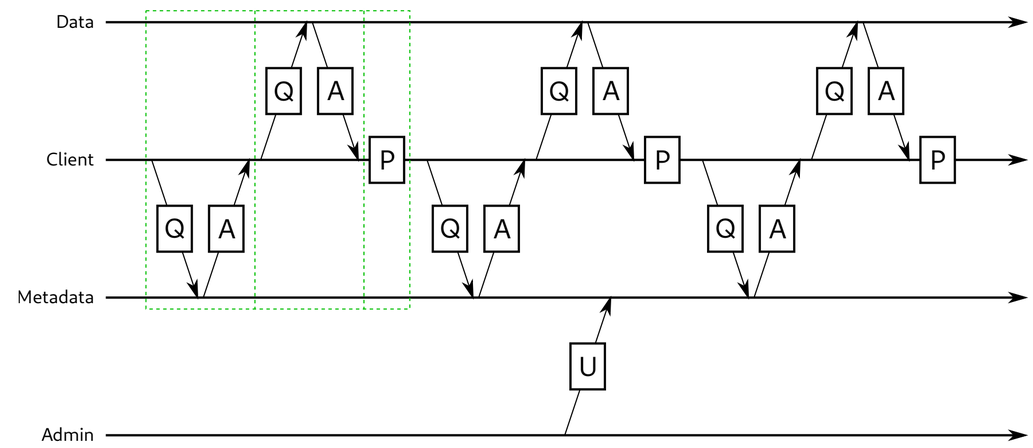
It is applicable to any domain where: How should a generic interface to building latency-intolerant applications on top of streaming data be constructed. In particular, where:

- There is a significant number of data streams (or historical timeseries data).
- There is metadata associated with context shared between multiple data streams.
- Multiple applications should operate on top of the same data model.
- The application interface should be simple.
- On-demand virtual datastreams are relevant.
- Availability, concurrency and latency matter.

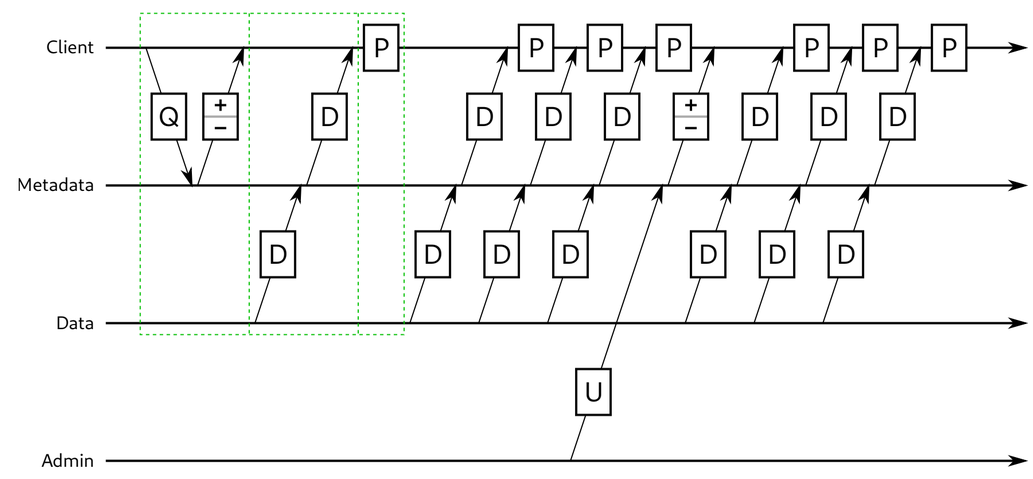
The initial motivation came from the building domain as support for an app ecosystem.

Interaction Model

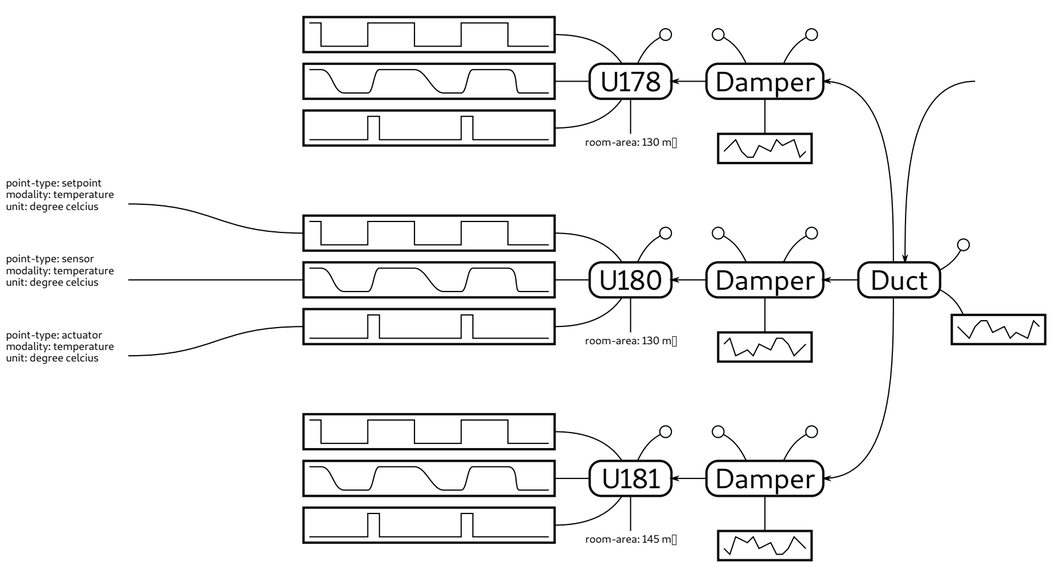
Typical model of polling information model:



Proposed model of subscribing to the result set of a query over the information model:



Domain Metadata Model Example



Technology

Query Anatomy The query format should be an extended version of a standardized query language (e.g., OpenCypher).

- ***Pattern*** A pattern definition to query the information model.
- ***Data Subscription*** A list of the data streams from each match site to forward.
- ***Unit Preferences*** A mapping from modality to unit for in-flight automatic convertion.
- ***Temporal Range*** The temporal range of interest. Historical data should be replayed until live data can be forwarded live. Indicators for the transitions.

Distributed Model: Once the service hosting the model becomes distributed the metadata essentially becomes software-defined.

- Support for improved availability.
- Support for model federation (equipment comes with own models, query results can be dependent on physical state).
- The subscription acts as a session, allowing for demand-driven soft/virtual sensors.

Design