GEO DATA STREAMS WITH GeoMQTT

Stefan Herle
RWTH Aachen University
Chair for Computing in Civil Engineering and Geo Information Systems & Geodetic Institute
20th AGILE Conference on Geographic Information Science
Workshop on Event-based Dissemination and Processing of Geospatial Information - 9th May 2017, Wageningen
Outline

1 GeoEvents & GeoPipes
2 Message Queue and Telemetry Transport (MQTT)
3 Extension GeoMQTT
   ▪ Subscription mechanism
   ▪ GeoMQTT in the EarlyDike project
4 Processing with GeoPipes
5 Conclusion and future developments
1 GeoEvents & GeoPipes

- **GeoEvent:**
  
  "an occurrence of a change of state associated to a phenomenon of interest, which is related to a geographic location and a specific time"  
  
  Garcia & Morales (2015)

  1) Phenomenon of interest
  2) Geographic space
  3) Time

- **Spatiotemporal data stream:** unbounded sequence of GeoEvents

→ **GeoPipes:** Linking different producers and consumers of real-time geospatial data in a push-based manner

Which protocol to define GeoEvents & GeoPipes? What are the requirements?
Addressing with **topics**

- Publish event with a topic name: `room/237/temperature`
- Subscribe to topic filters
  - Entire topic name → `room/237/temperature`
  - "+": single level wildcard → `room/+/temperature`
  - "#": multi level wildcard → `room/237/#`
Topic-based publish/subscribe model, message transmission in near real-time

Open standard by OASIS 2013

Low Overhead (2 bytes header) → lightweight messaging

Standard gives guidance for security implementation

Quality of Service (QoS) & Last Will and Testament (LWT) feature provides reliability

Scalable with TCP load balancers

MQTT for Sensor Networks for constrained devices and connectionless environments → interoperability
3 Extension GeoMQTT

- GeoPipe message:
  
  \[(<\text{time stamp/period}>, <\text{geometry}>, <\text{eventname}>, <\text{payload}>)\]

- Spatiotemporal data stream: stream of GeoEvent tuples

\[(2015-09-22T11:10:21+00:00; \text{POINT}(6.06799 50.77906); \text{temperature}; 20.752)\]
\[(2015-09-22T11:10:22+00:00; \text{POINT}(6.06799 50.77906); \text{temperature}; 20.760)\]
\[(2015-09-22T11:10:23+00:00; \text{POINT}(6.06799 50.77906); \text{temperature}; 20.769)\]
\[(2015-09-22T11:10:24+00:00; \text{POINT}(6.06799 50.77906); \text{temperature}; 20.759)\]
... 
\[(2015-09-22T11:10:52+00:00; \text{POINT}(6.06799 50.77906); \text{temperature}; 20.750)\]
Based on ISO8601 standard for (repeating) time intervals:

- `<start>/<end>`
- `<start>/<duration>`
- `<unixtime>/<duration>`
- `<unixtime>/<seconds>`
- `(<CRON>)/<duration>`
- `(<CRON>)/<seconds>`

Examples:

1) 2015-08-31T12:00:00/2015-09-15T10:00:00
2) 2015-08-31T12:00:00/PT2H
3) 1441017813/7200
4) (0 0 8 ? * SAT)/PT2H30M

with: `<start>,<end>` in ISO time stamps, `<duration>` in ISO duration, `<unixtime>` in Epoch time, `<CRON>` according to Quartz scheduler

Evaluate if time stamp of GeoPublish message is **within** time interval of GeoSubscription
3 GeoMQTT: Spatial Filter

- Filters event by *geometry and spatial relation* (DE-9IM)
- Geometry is specified in *Well-known Text (WKT)*, *EWKT*, *GeoJSON* or *GML*.

**Examples:**
1) SRID=6931;POINT(453467.126590921,4265581.16477956)
2) POLYGON((11.59 51.88, 11.76 51.37, 12.61 51.41, 12.16 52.35, 11.98 52.18, 11.59 51.88))
3) BUFFER(POINT(6.06799 50.77906), 1.0)

→ Geometry of publish message evaluated with relation and geometry of subscription

→ Publish message has *CRS of the geometry* in the subscription and geometry is encoded in the specified format.
Demo
Extending MQTT-SN to GeoMQTT-SN:

- Added also three message type: GeoPublish, GeoSubscribe & GeoUnsubscribe
- Adjusted to constrained devices (Arduinos, Waspmotes, etc.)
3 EarlyDike – Geo Event Bus

Geo Event Bus

MQTT

Observation layer

Integration layer

Presentation layer

SOS  SPS  SES
SWE Services

WPS' for data streams

QGIS Client

REST

3rd party services/data

Simulators

Bridges

…

Read Events

Publish Events

Request Service

Recv. Data
Goal: Implementation of WPS services performing functions on spatiotemporal data streams

- Publisher
- Publisher
- Publisher
- Publisher

4 Processing with GeoPipes

- MQTT Broker
- MQTT Broker
- MQTT Broker

- InGeoPipes
- OutGeoPipe

- Service A
- Service B
- Service C
- Service D

- Send events
- Publish events
- Service request
- Service response

WPS 1.0 Client

WPS 1.0 Server
Demo - Computing the moving average

Size of Sliding Window (max 10)
3

TTL (max 300 sec.)
30

Input GeoPipe
wpsdemo.gia.rwth-aa
1883
node/m1/temperature
temporalfilter
BBOX(-180 -90, 180 9)

Output GeoPipe
wpsdemo.gia.rwth-aa
1883
movingaverage/temperature

POINT (6.065472 50.778583)

Values
10
8
12:44:15
12:44:30

magapp_3azej9kixwmu/movingaverage/temperature/node/m1/temperature

Highcharts.com
Demo - Computing the dyn. convex hull
Demo - Map matching cars
4 Processing with SPE

Taxi trajectory points

Stream Processing Engine

Visualization in a MapViewer

Stream Processing Engine

Map Matching

Visualization in a MapViewer
GeoMQTT encapsulates GeoEvents to initiate spatiotemporal data streams

GeoPipes for couple IoTS with GIS and geo web service

Enables real-time spatial workflows from data collection till visualization

→ EarlyDike project for early warning of dike failures

WPS extension to set up services performing functions on spatiotemporal data streams

Future developments: Couple GeoPipes with stream processing engines: spatial mining, interpolation…