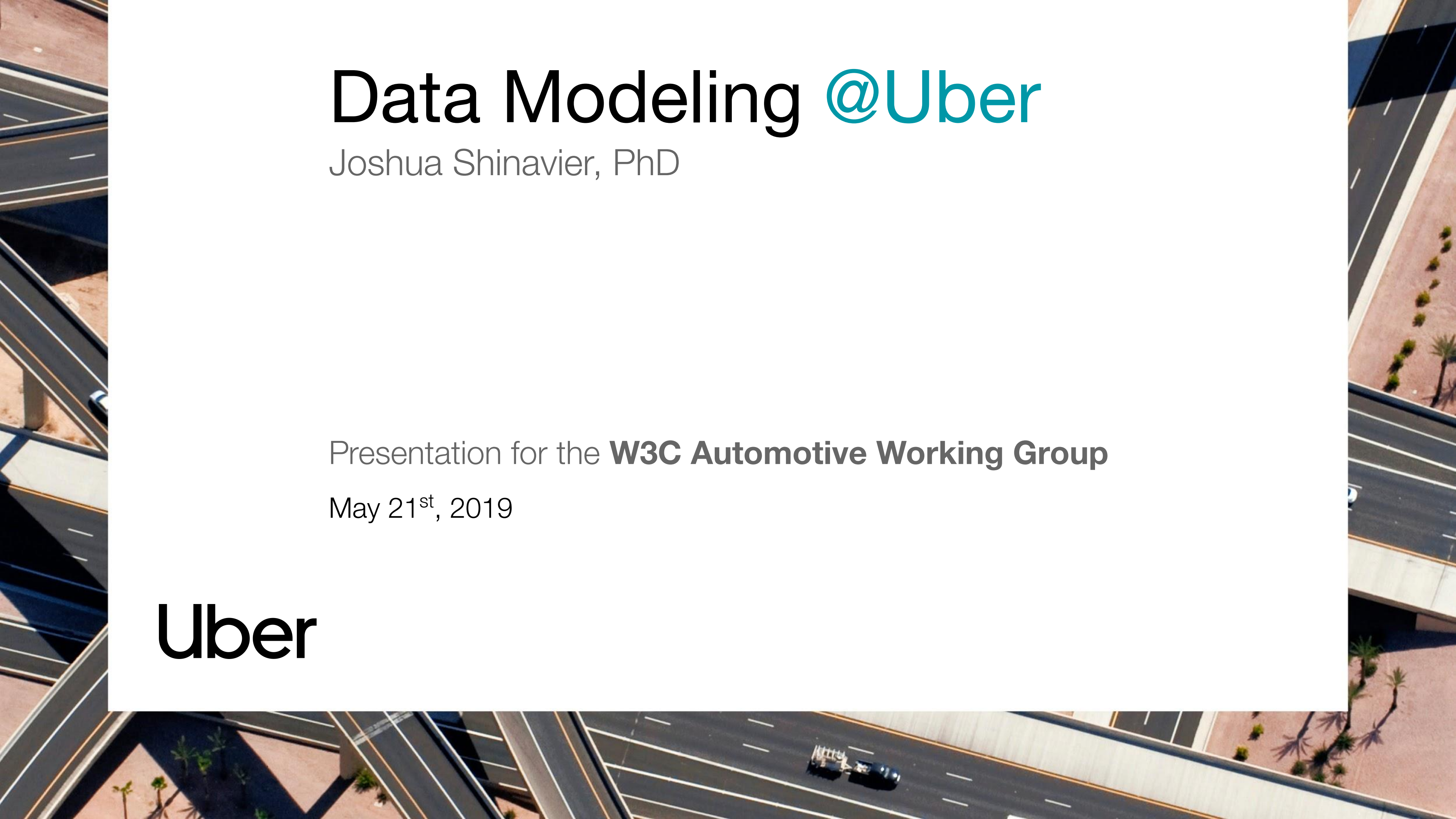


Data Modeling @Uber

Joshua Shinavier, PhD

Presentation for the **W3C Automotive Working Group**

May 21st, 2019

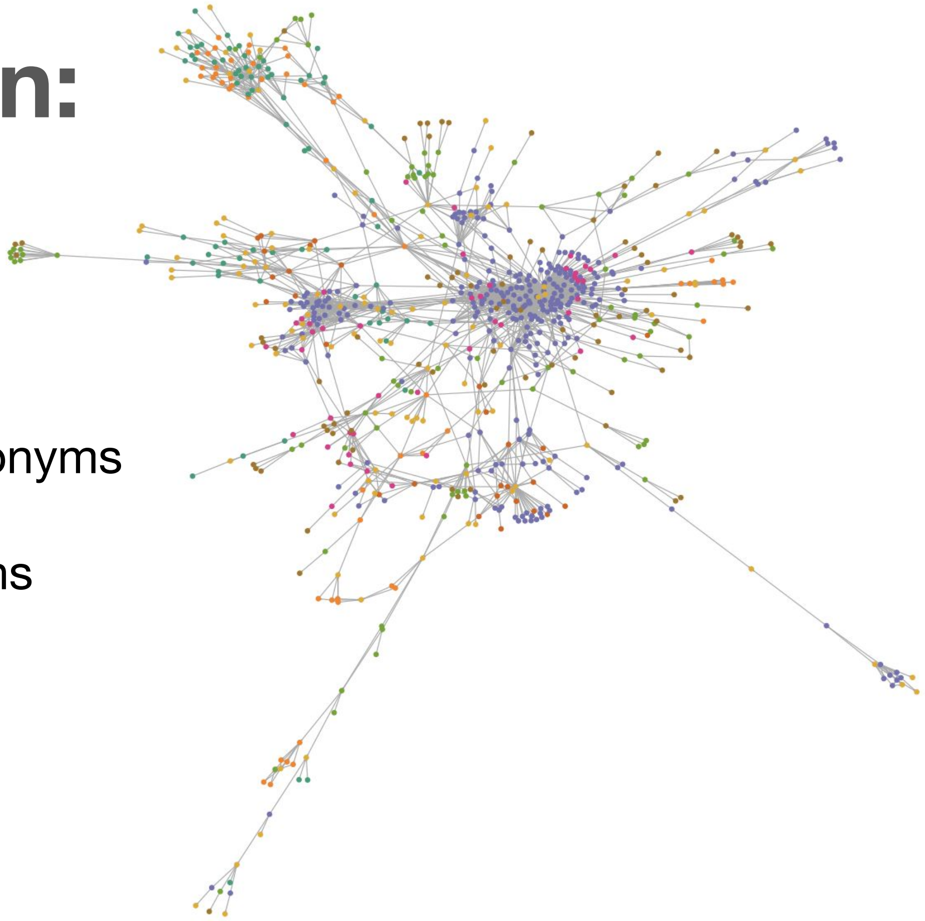
Uber

Data and schemas @Uber

- 200k managed data sets
- 10+ billion trips
  - Low thousands of new entities per second
- Even more sensor data
  - Use cases for graph stream processing
- On-demand, streaming, RPC
  - Each dataset, stream, and service has its own schema

Schema integration: challenges

- Data sources are not composable
- Strong identifiers, weak semantics
    - Duplicate types, homonyms, synonyms
- Per-language data islands
- Diversity of data modeling conventions



Data Standardization

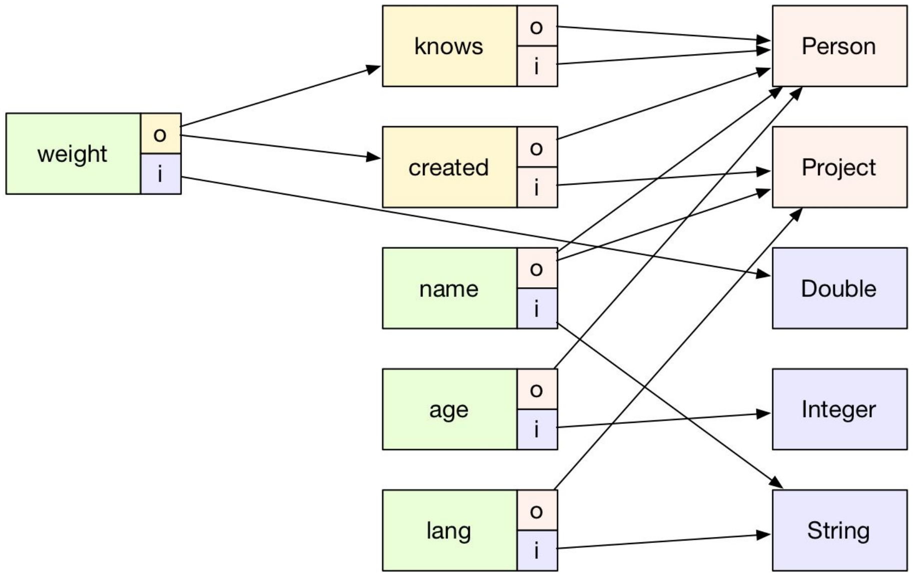
- Controlled vocabularies for all of Uber
    - Basic type aliases (URL, UUID, timestamps, etc.)
    - Structured types (sensor events, currency values, etc.)
    - Entities and relationships (User, Vehicle, Trip, etc.)
    - Metadata vocabularies
- Shared logical data model
- Basic domain vocabularies
    - E.g. time, geometry and geolocation, addresses and contact info, sensors, money, etc.
- Tooling carries schemas between data representation languages
    - Protobuf, Thrift, Avro, RDF, PG, etc.
    - Schema and data transformations are composable

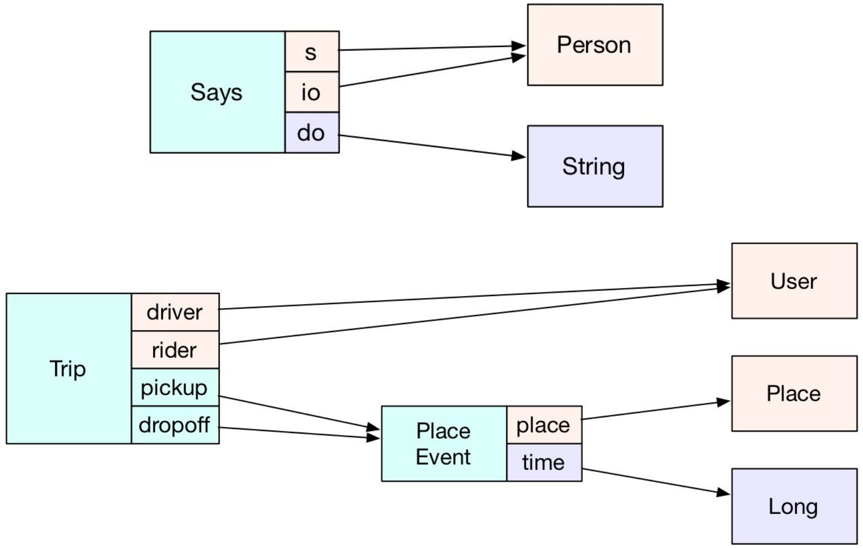
Metadata graph

- Need metadata for each dataset at Uber
- Data protections and user trust
  - GDPR and other regulations, Uber’s own data policies
  - What kind of user data? Where is it?
  - Heroic numbers of manual annotations
    - Limited expressivity, limited guarantees
    - *Inference* is required
- In annotating datasets, standardize and compose schemas

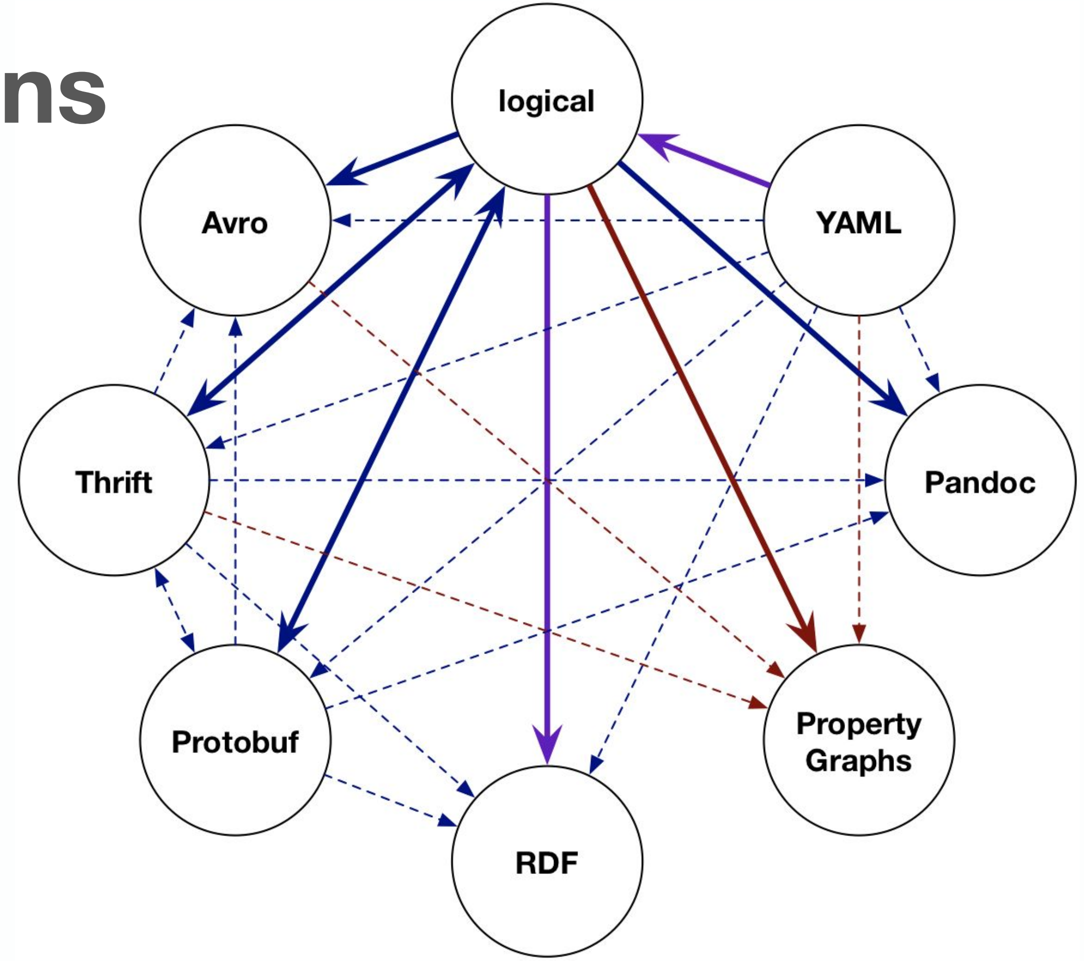
Algebraic Property Graphs

- Common data model for RPC, storage, and KR at Uber
- In progress: alignment with the **Property Graph Schema Working Group**
- In progress: "Universal structure" of **TinkerPop4**

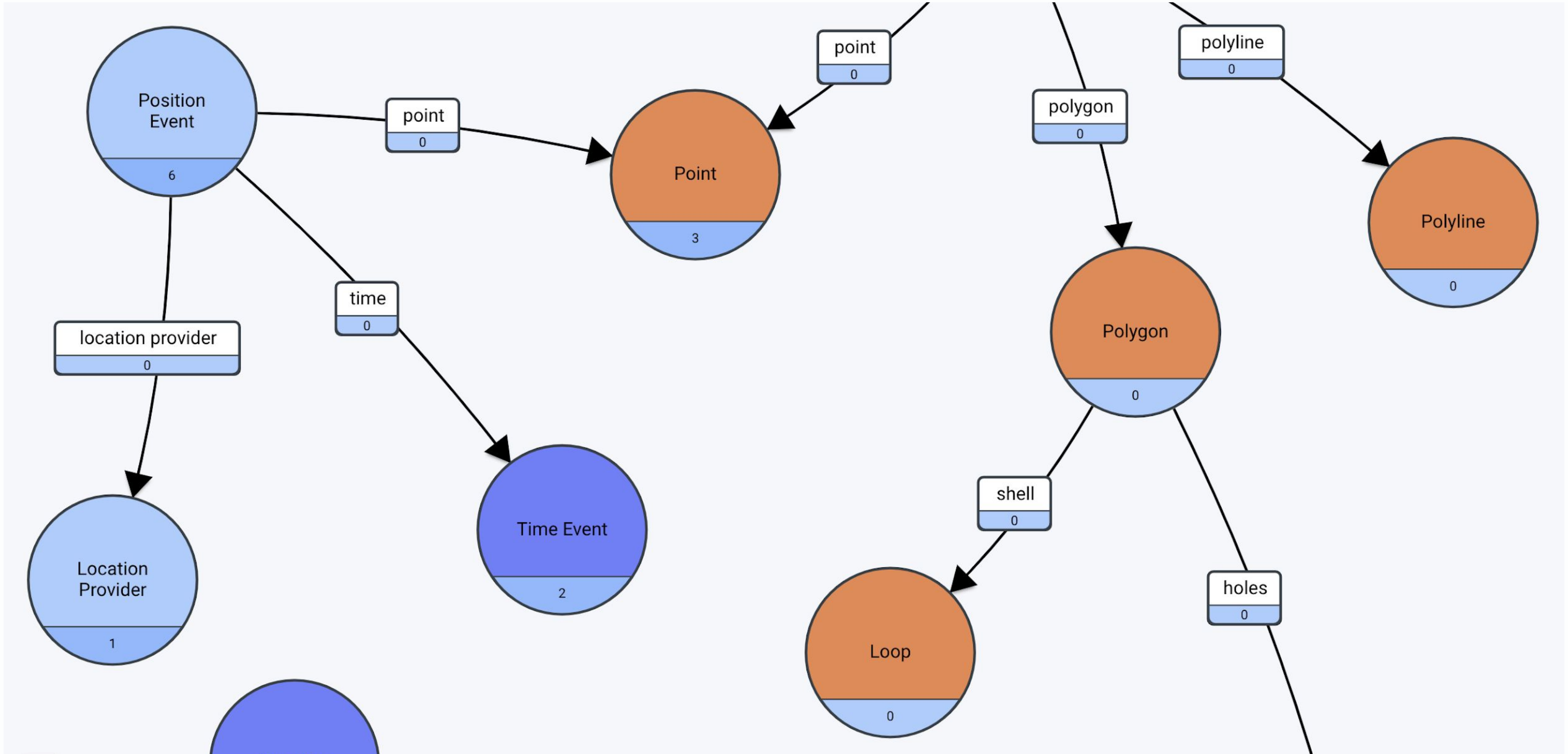




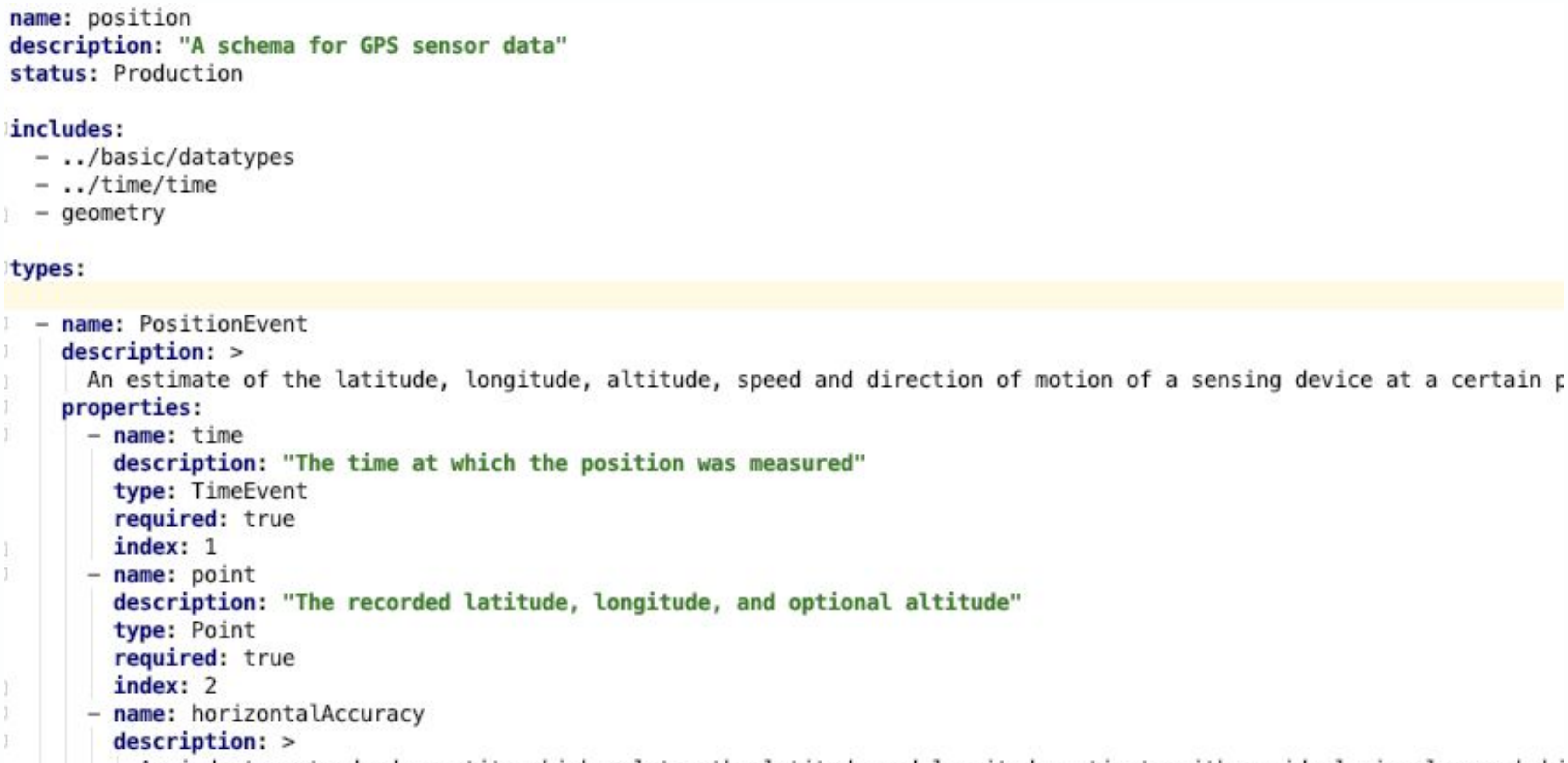
Transformations



Logical

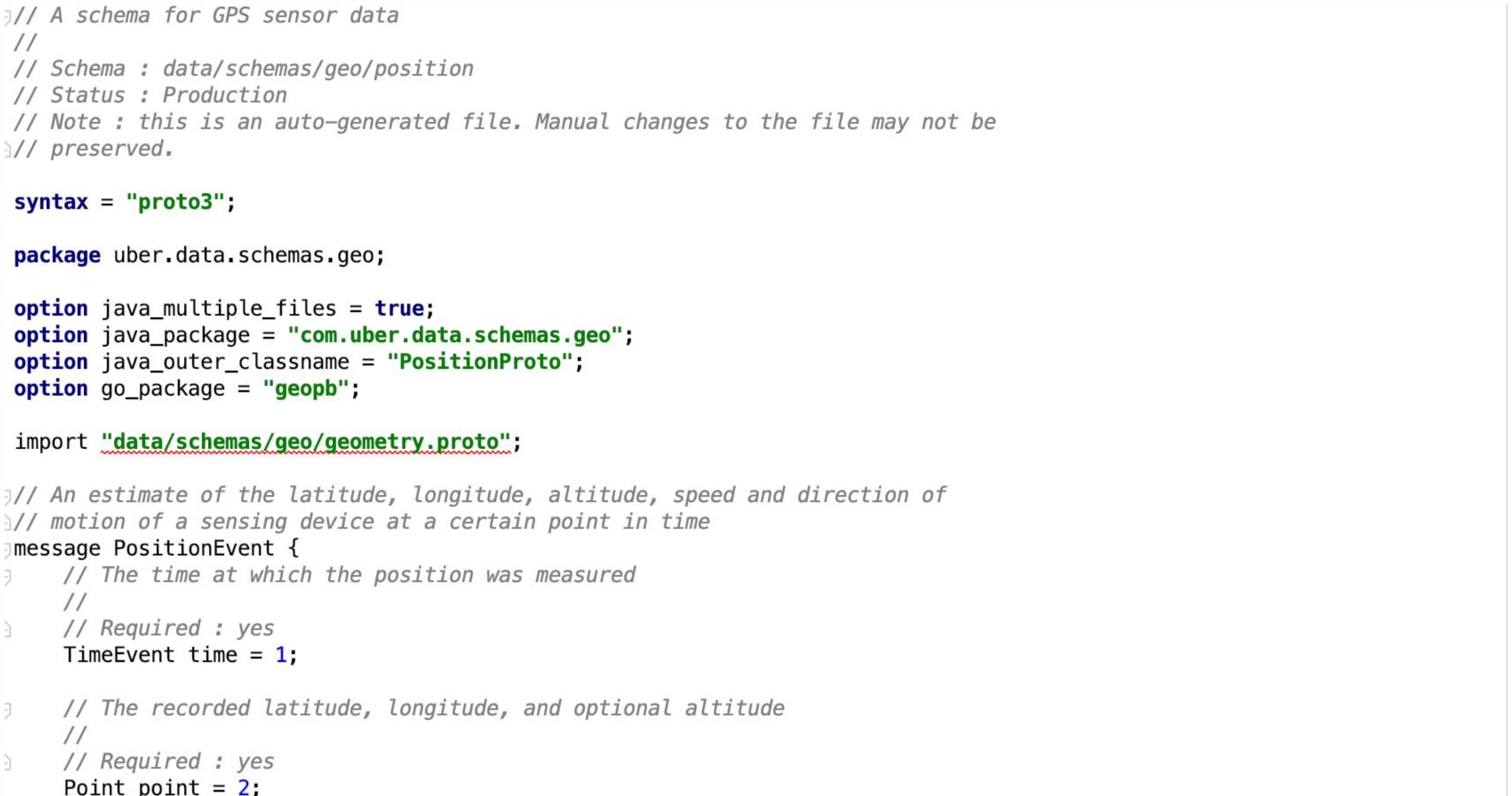

Position Event
6
point
0
Point
3
point
0
polygon
0
polyline
0
Polyline
0
Polygon
0
time
0
location provider
0
Location Provider
1
Time Event
2
shell
0
Loop
0
holes
0

YAML



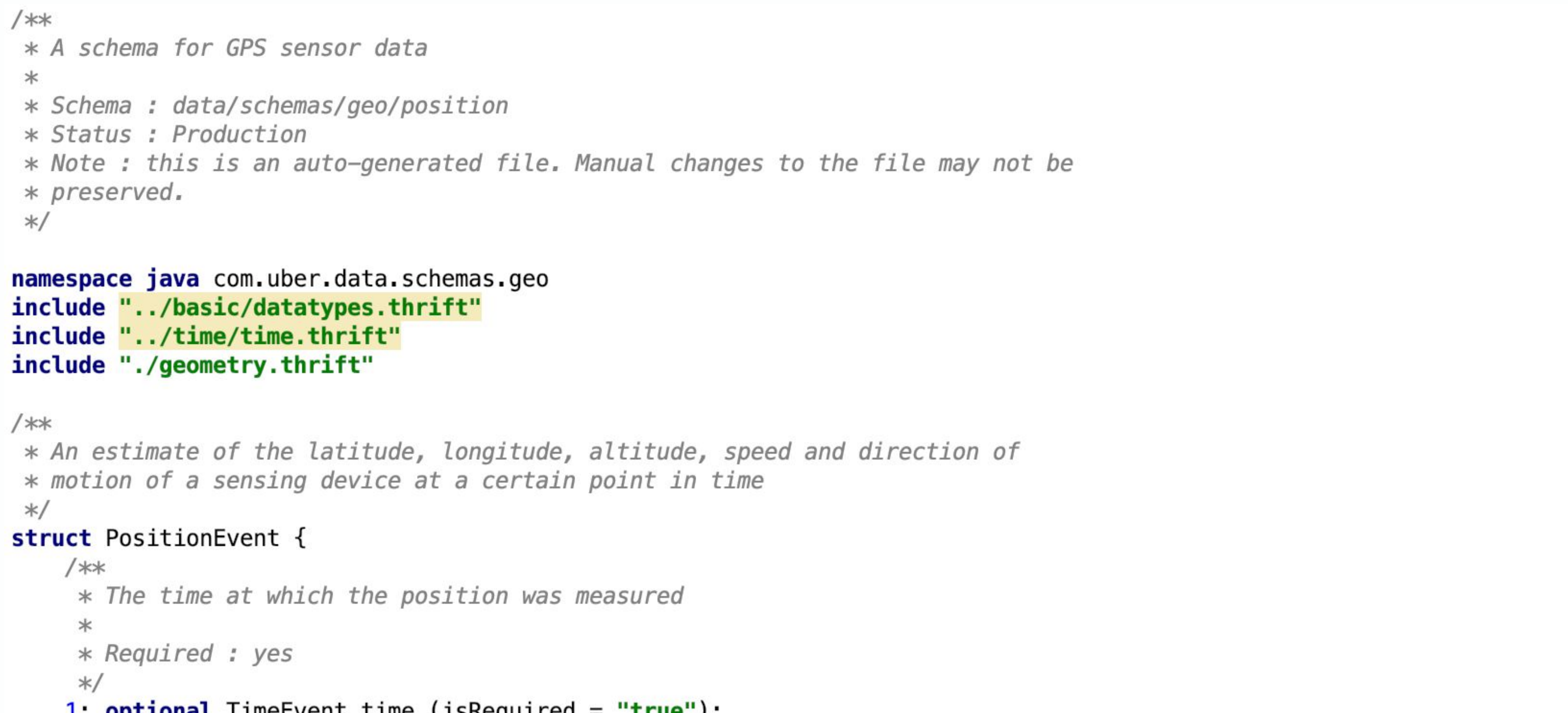
```yaml
name: position
description: "A schema for GPS sensor data"
status: Production

includes:
  - ../basic/datatypes
  - ../time/time
  - geometry

types:

  - name: PositionEvent
    description: >
      An estimate of the latitude, longitude, altitude, speed and direction of motion of a sensing device at a certain p
    properties:
      - name: time
        description: "The time at which the position was measured"
        type: TimeEvent
        required: true
        index: 1
      - name: point
        description: "The recorded latitude, longitude, and optional altitude"
        type: Point
        required: true
        index: 2
      - name: horizontalAccuracy
        description: >
```

Protocol Buffers



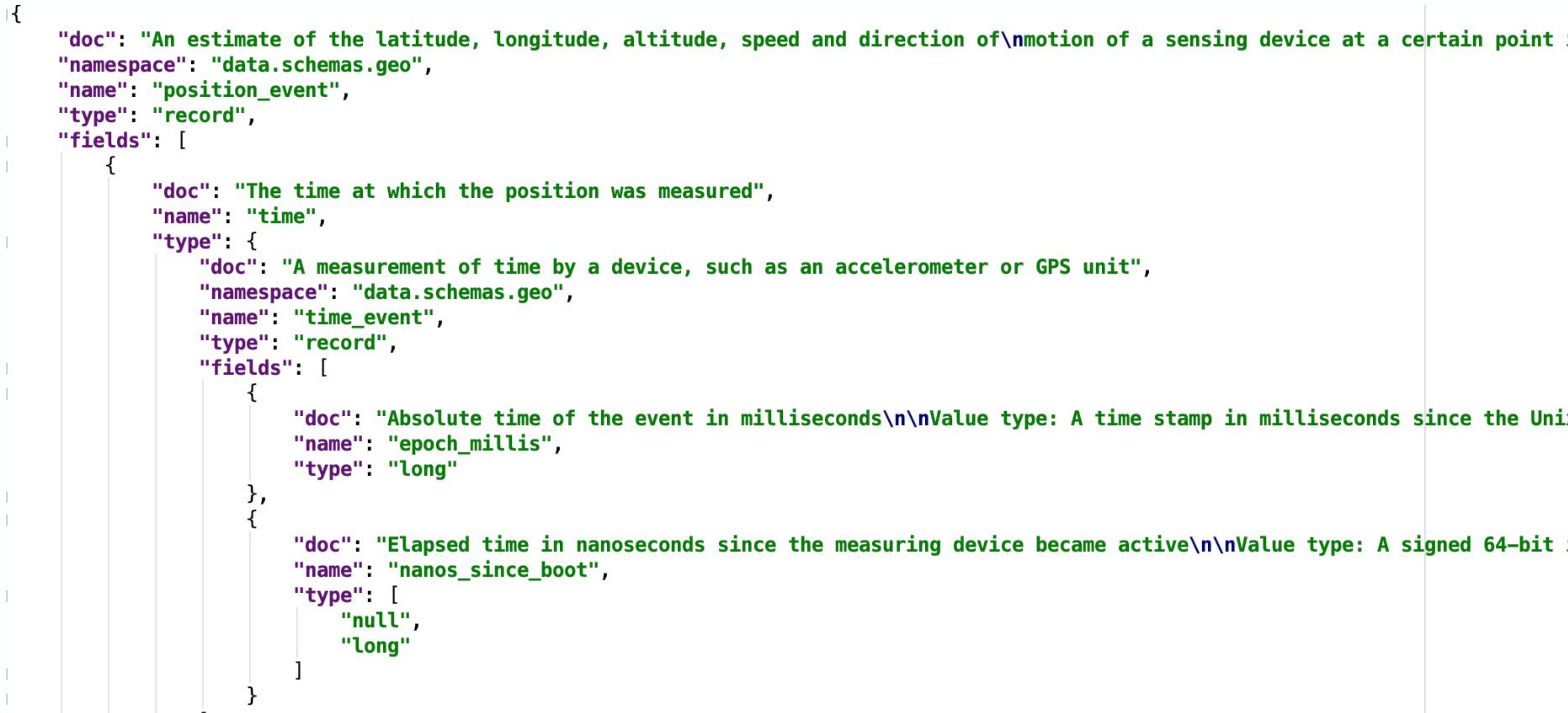
```
// A schema for GPS sensor data
//
// Schema : data/schemas/geo/position
// Status : Production
// Note : this is an auto-generated file. Manual changes to the file may not be
// preserved.

syntax = "proto3";

package uber.data.schemas.geo;

option java_multiple_files = true;
option java_package = "com.uber.data.schemas.geo";
option java_outer_classname = "PositionProto";
option go_package = "geopb";

import "data/schemas/geo/geometry.proto";

// An estimate of the latitude, longitude, altitude, speed and direction of
// motion of a sensing device at a certain point in time
message PositionEvent {
    // The time at which the position was measured
    //
    // Required : yes
    TimeEvent time = 1;

    // The recorded latitude, longitude, and optional altitude
    //
    // Required : yes
    Point point = 2;
```

Apache Thrift



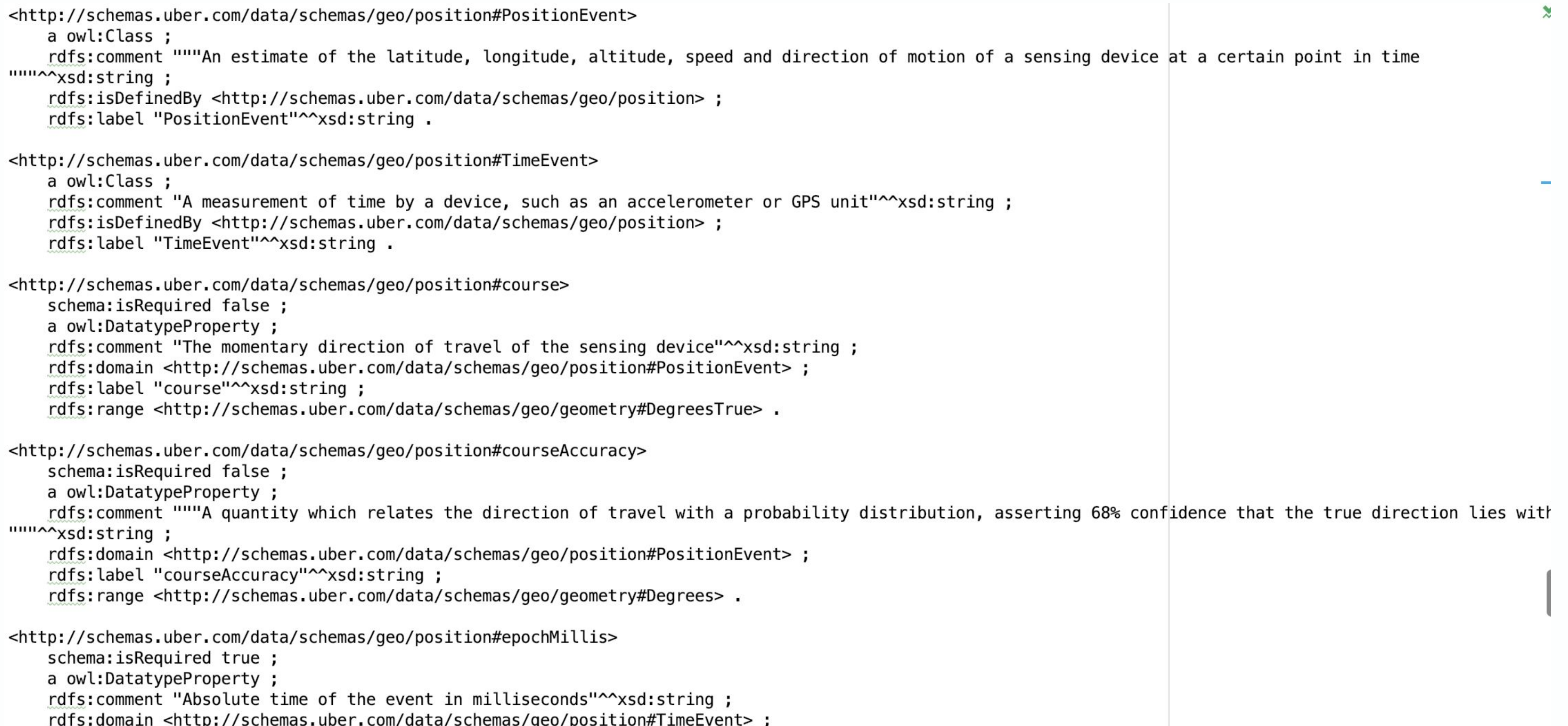
```
/**
 * A schema for GPS sensor data
 *
 * Schema : data/schemas/geo/position
 * Status : Production
 * Note : this is an auto-generated file. Manual changes to the file may not be
 * preserved.
 */

namespace java com.uber.data.schemas.geo
include "../basic/datatypes.thrift"
include "../time/time.thrift"
include "./geometry.thrift"

/**
 * An estimate of the latitude, longitude, altitude, speed and direction of
 * motion of a sensing device at a certain point in time
 */
struct PositionEvent {
    /**
     * The time at which the position was measured
     *
     * Required : yes
     */
    1: optional TimeEvent time (isRequired = "true");
```

Apache Avro



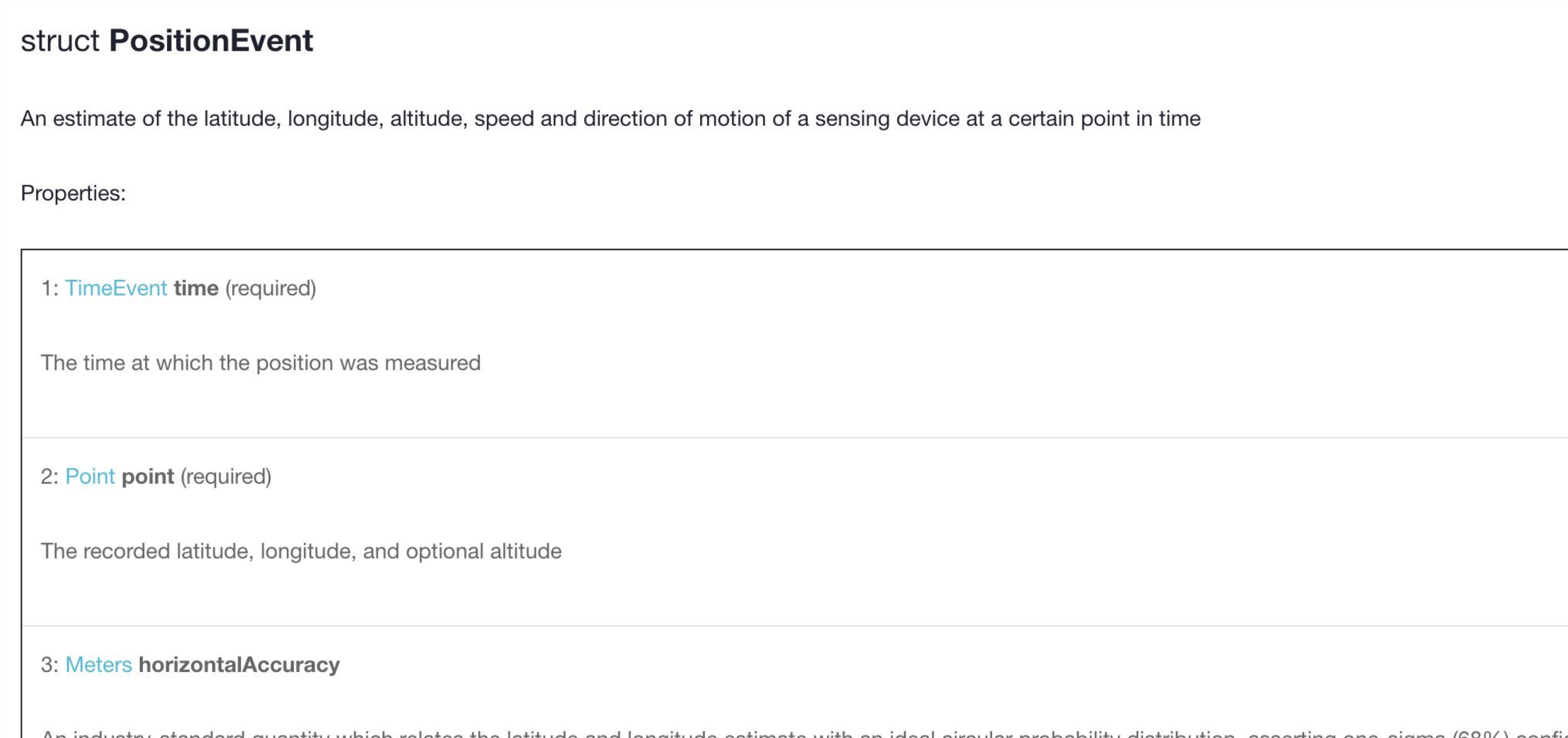
```
{
    "doc": "An estimate of the latitude, longitude, altitude, speed and direction of\nmotion of a sensing device at a certain point
    "namespace": "data.schemas.geo",
    "name": "position_event",
    "type": "record",
    "fields": [
        {
            "doc": "The time at which the position was measured",
            "name": "time",
            "type": {
                "doc": "A measurement of time by a device, such as an accelerometer or GPS unit",
                "namespace": "data.schemas.geo",
                "name": "time_event",
                "type": "record",
                "fields": [
                    {
                        "doc": "Absolute time of the event in milliseconds\n\nValue type: A time stamp in milliseconds since the Uni
                        "name": "epoch_millis",
                        "type": "long"
                    },
                    {
                        "doc": "Elapsed time in nanoseconds since the measuring device became active\n\nValue type: A signed 64-bit
                        "name": "nanos_since_boot",
                        "type": [
                            "null",
                            "long"
                        ]
                    }
```

Turtle (OWL)



```
<http://schemas.uber.com/data/schemas/geo/position#PositionEvent>
    a owl:Class ;
    rdfs:comment """An estimate of the latitude, longitude, altitude, speed and direction of motion of a sensing device at a certain point in time
"""^^xsd:string ;
    rdfs:isDefinedBy <http://schemas.uber.com/data/schemas/geo/position> ;
    rdfs:label "PositionEvent"^^xsd:string .

<http://schemas.uber.com/data/schemas/geo/position#TimeEvent>
    a owl:Class ;
    rdfs:comment "A measurement of time by a device, such as an accelerometer or GPS unit"^^xsd:string ;
    rdfs:isDefinedBy <http://schemas.uber.com/data/schemas/geo/position> ;
    rdfs:label "TimeEvent"^^xsd:string .

<http://schemas.uber.com/data/schemas/geo/position#course>
    schema:isRequired false ;
    a owl:DatatypeProperty ;
    rdfs:comment "The momentary direction of travel of the sensing device"^^xsd:string ;
    rdfs:domain <http://schemas.uber.com/data/schemas/geo/position#PositionEvent> ;
    rdfs:label "course"^^xsd:string ;
    rdfs:range <http://schemas.uber.com/data/schemas/geo/geometry#DegreesTrue> .

<http://schemas.uber.com/data/schemas/geo/position#courseAccuracy>
    schema:isRequired false ;
    a owl:DatatypeProperty ;
    rdfs:comment """A quantity which relates the direction of travel with a probability distribution, asserting 68% confidence that the true direction lies wit
"""^^xsd:string ;
    rdfs:domain <http://schemas.uber.com/data/schemas/geo/position#PositionEvent> ;
    rdfs:label "courseAccuracy"^^xsd:string ;
    rdfs:range <http://schemas.uber.com/data/schemas/geo/geometry#Degrees> .

<http://schemas.uber.com/data/schemas/geo/position#epochMillis>
    schema:isRequired true ;
    a owl:DatatypeProperty ;
    rdfs:comment "Absolute time of the event in milliseconds"^^xsd:string ;
    rdfs:domain <http://schemas.uber.com/data/schemas/geo/position#TimeEvent> ;
```

Docs



struct **PositionEvent**

An estimate of the latitude, longitude, altitude, speed and direction of motion of a sensing device at a certain point in time

Properties:

1: TimeEvent **time** (required)

The time at which the position was measured

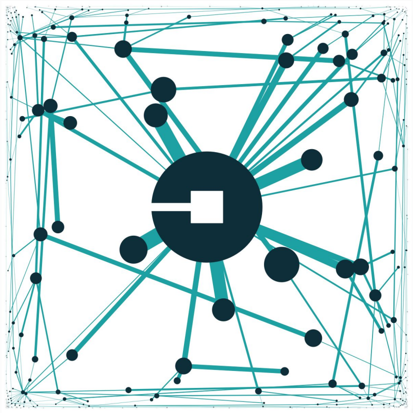
2: Point **point** (required)

The recorded latitude, longitude, and optional altitude

3: Meters **horizontalAccuracy**

An industry-standard quantity which relates the latitude and longitude estimate with an ideal circular probability distribution, asserting one-sigma (68%) confi

Thanks



joshsh@uber.com