Using MobilityDB and Grafana for Aviation Trajectory Analysis

What will we cover?

Overview of MobilityDB and Grafana

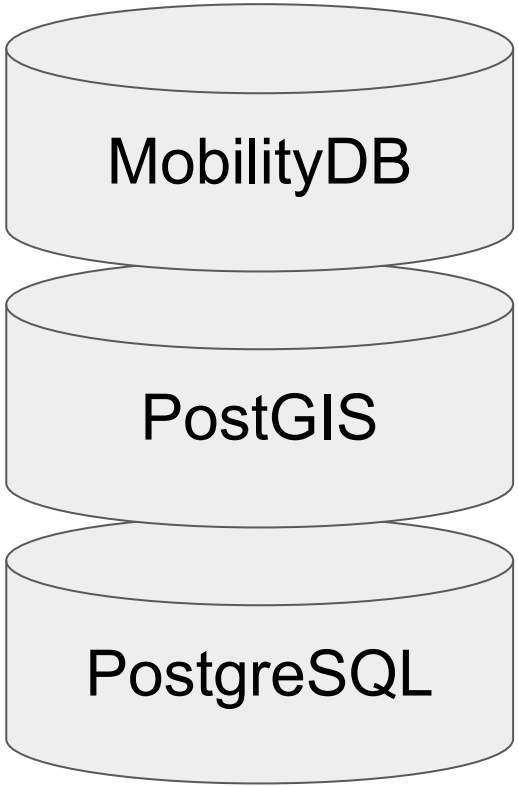
How the technologies work well together

Steps for transformations and analysis implementation

Specific SQL query examples

Conclusions of our work

MobilityDB Overview



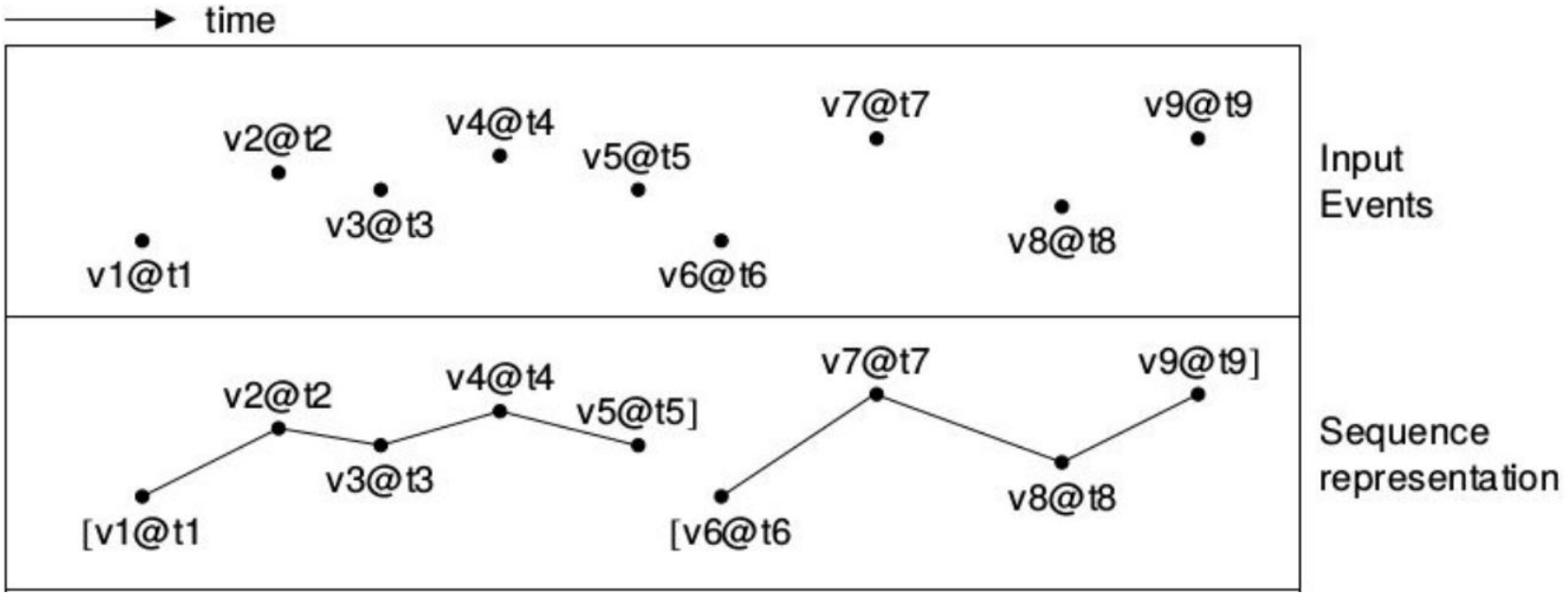
tgeompoint, tgeogpoint,
tint, tfloat, ttext, tbool

geometry, geography

numeric, monetary, character,
data/time, boolean, enum,
arrays, range,
XML, JSON, ...

MobilityDB Temporal Types

Abstract data model



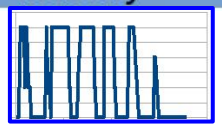
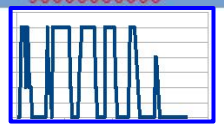
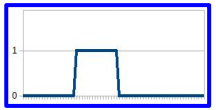
Temporal Types

time	icao24	lat	lon	velocity	vertrate	callsign	onground	alert	spi	geoaltitude
1590969910	7380c4	37.2278594970703	35.2675157912234	221.358025998638	-0.32512	ELY076	False	False	False	12207.24
1590969920	7380c4	37.2064105534958	35.233154296875	221.675226897561	-0.32512	ELY076	False	False	False	12207.24
1590969930	7380c4	37.1942596435547	35.2137529089096	221.675226897561	-0.65024	ELY076	False	False	False	12207.24
1590969940	7380c4	37.1815516585011	35.193567690642	221.675226897561	-0.32512	ELY076	False	False	False	12207.24
1590969950	7380c4	37.1695411811441	35.1743416164232	221.358025998638	-0.32512	ELY076	False	False	False	12207.24
1590969960	7380c4	37.1570651814089	35.1543393342391	220.952949389884	0	ELY076	False	False	False	12199.62
1590969970	7380c4	37.1451873779297	35.1353876641456	220.635909596049	0	ELY076	False	False	False	12199.62
1590969980	7380c4	37.1331939697266	35.116278465758	220.635909596049	0	ELY076	False	False	False	12199.62
1590969990	7380c4	37.1205682269597	35.0961237368376	220.952949389884	0	ELY076	False	False	False	12207.24
1590970000	7380c4	37.1080456749868	35.0761811629586	220.952949389884	-0.32512	ELY076	False	False	False	12207.24
1590970010	7380c4	37.0961608886719	35.0572561710439	220.952949389884	-0.32512	ELY076	False	False	False	12207.24
1590970020	7380c4	37.0836988546081	35.037490181301	221.270730983924	-0.32512	ELY076	False	False	False	12207.24
1590970030	7380c4	37.0714090638241	35.0179655655571	221.270730983924	-0.32512	ELY076	False	False	False	12199.62
1590970040	7380c4	37.0591192730402	34.9984409498132	221.270730983924	0	ELY076	False	False	False	12199.62
1590970050	7380c4	37.0469512939453	34.9790662400266	221.675226897561	0.32512	ELY076	False	False	False	12199.62
1590970060	7380c4	37.0344931392346	34.9593320100204	221.675226897561	0.32512	ELY076	False	False	False	12207.24
1590970070	7380c4	37.0220031738281	34.939562209109	221.993166722344	-0.32512	ELY076	False	False	False	12207.24
1590970080	7380c4	37.0095411397643	34.9198648203974	221.993166722344	-0.32512	ELY076	False	False	False	12207.24
1590970090	7380c4	36.9973297119141	34.9004672436004	221.993166722344	-0.32512	ELY076	False	False	False	12199.62

For flights entity (table)

Before: size 2.5 GB

```
CREATE TABLE flight_traj (icao24, callsign, tonground, taltitude, tgeom) AS
SELECT icao24, callsign,
       tbool_seq(array_agg (tbool_inst (onground, et_ts) ORDER BY et_ts)
                    FILTER (WHERE onground IS NOT NULL)),
       tfloat_seq(array_agg (tfloat_inst (geoaltitude, et_ts) ORDER BY et_ts)
                    FILTER (WHERE geoaltitude IS NOT NULL)),
       tgeompoint_seq(array_agg (tgeompoint_inst (geom, et_ts) ORDER BY et_ts)
                    FILTER (WHERE geom IS NOT NULL))
  FROM flights
  GROUP BY icao24, callsign;
```

icao24	velocity	vertrate	callsign	onground	tgeometry
7380c4			ELY076		

After: 870 MB

Grafana Overview

- Open source dashboard
- Supports multiple sources
- Advanced spatiotemporal visualizations
- Variables to filter query through GUI
- Alert notification feature
- Plugins, community, data exploration

Grafana + MobilityDB Together

Strengths

Immediate visualizations of spatial queries (improvement on QGIS)

Multiple queries easily combined (simplifies SQL)

Final filters and data visualization processing in Grafana through GUI

Challenges

Trajectory visualization

Data explosion on un-controlled queries

Implementation overview

1.	Clean and Transform Data	OpenSky → MobilityDB
2.	Create Queries and Visualizations	MobilityDB ↔ Grafana
3.	Deploy and Respond to End User	Grafana → MobilityDB → Grafana

Data Source

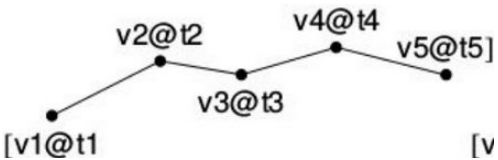
Backend

Frontend

1: Clean and Transform

```
WITH icao24_with_null_lat AS (
      SELECT icao24, COUNT(lat)
      FROM flights
      GROUP BY icao24
      HAVING COUNT(lat) = 0)
DELETE
FROM flights
WHERE icao24 IN (SELECT icao24 FROM
icao24_with_null_lat);
```

icao24	lat	lon	velocity	vertrate	callsign	onground
89900	42.3144014002913	-75.5166342092115	235.301196933626	0	EVA031	False
89900	42.3314666748047	-75.5302567915483	234.596053220677		EVA031	False
89900		-75.5457375266335			EVA031	
89900	42.3680295782574		233.448769545272	0.32512	EVA031	False
89900	42.3868103027344	-75.5745766379616	233.448769545272		EVA031	False
89900		-75.5893228220385	233.448769545272	-0.32512	EVA031	
89900	42.4231567382813	-75.6037278608842	232.743699027094	0	EVA031	False



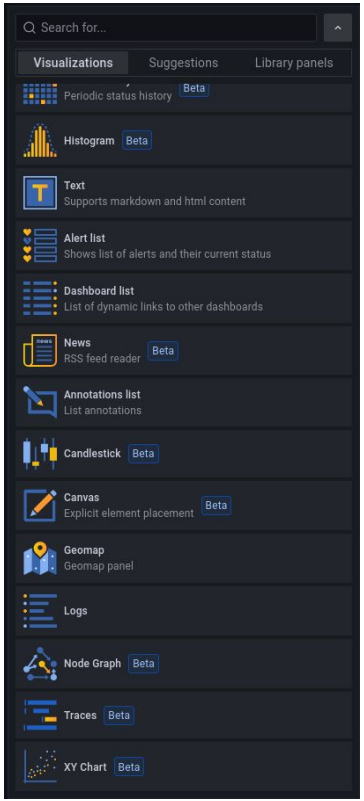

```
CREATE TABLE flight_traj (icao24, callsign, tonground, taltitude, tgeom) AS
SELECT icao24, callsign,
      tbool_seq(array_agg (tbool_inst (onground, et_ts) ORDER BY et_ts)
                  FILTER (WHERE onground IS NOT NULL)),
      tfloat_seq(array_agg (tfloat_inst (geoaltitude, et_ts) ORDER BY et_ts)
                  FILTER (WHERE geoaltitude IS NOT NULL)),
      tgeompoint_seq(array_agg (tgeompoint_inst (geom, et_ts) ORDER BY et_ts)
                  FILTER (WHERE geom IS NOT NULL))
  FROM flights
  GROUP BY icao24, callsign;
```

2: Create Queries and Visualizations

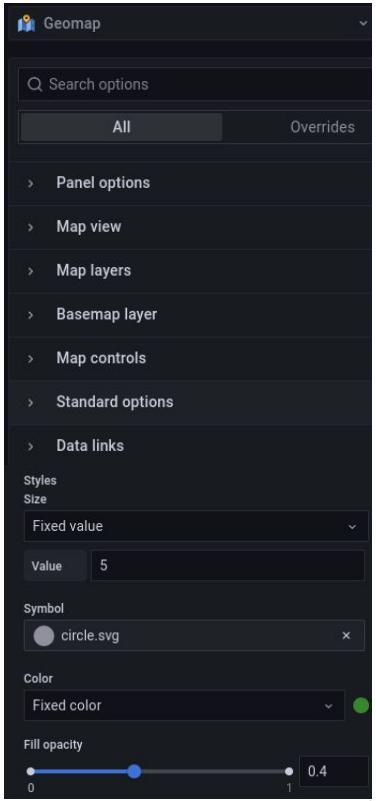
SQL Queries designed to be used with Grafana variables

Push query processing to MobilityDB. Design to limit data transfer into Grafana

Visualizations and data representation (marker colours, sizes, limits etc.) set in Grafana



Visualization Style



Panel Specific Settings

3: Deploy and Respond to End-User

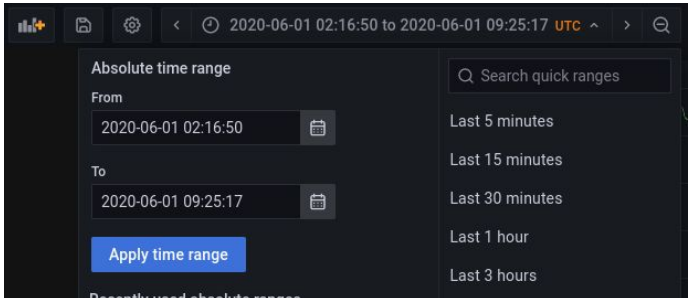
GUI is used with predefined filters (no SQL knowledge needed)

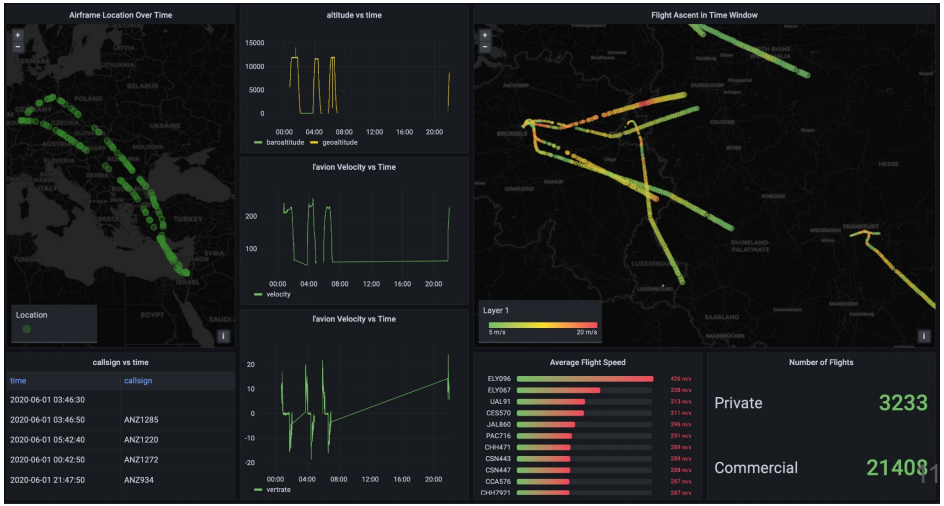
Interactive visualizations with dynamically changing information

No transformations or table creation, only data retrieval

Users can adjust panels and SQL queries if desired

Real-time monitoring and alarms available





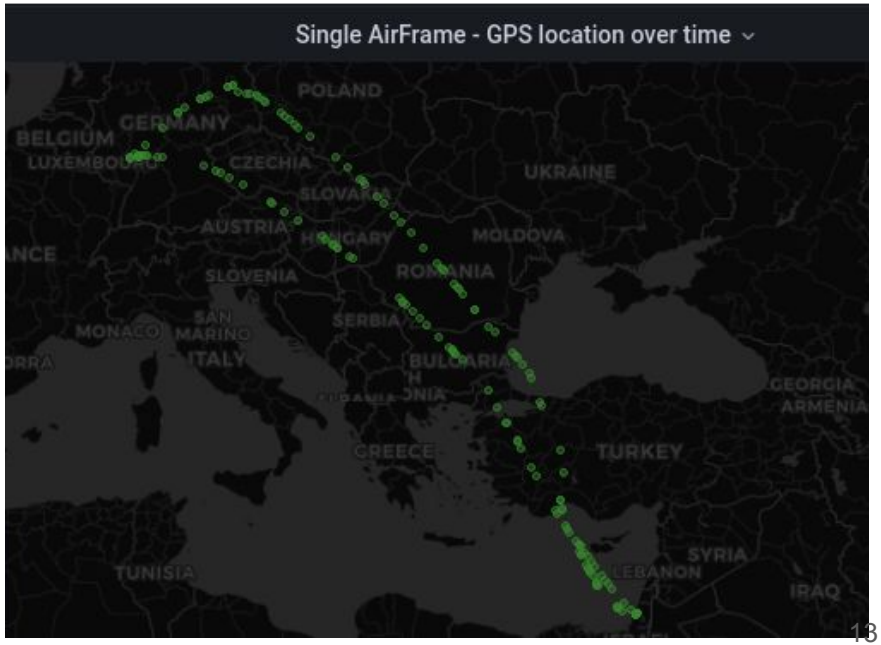
SQL Query Examples

What is the flight path of a single airframe over a user defined time period?

```
SELECT et_ts, icao24, lat, lon
FROM flights TABLESAMPLE SYSTEM (5)
WHERE icao24 IN ('738286')
  AND $__timeFilter(et_ts);
```

Trajectory visualization using discrete points and limiting data returned

User input through Grafana variable



What is the altitude and ground speed of flight TRA051?

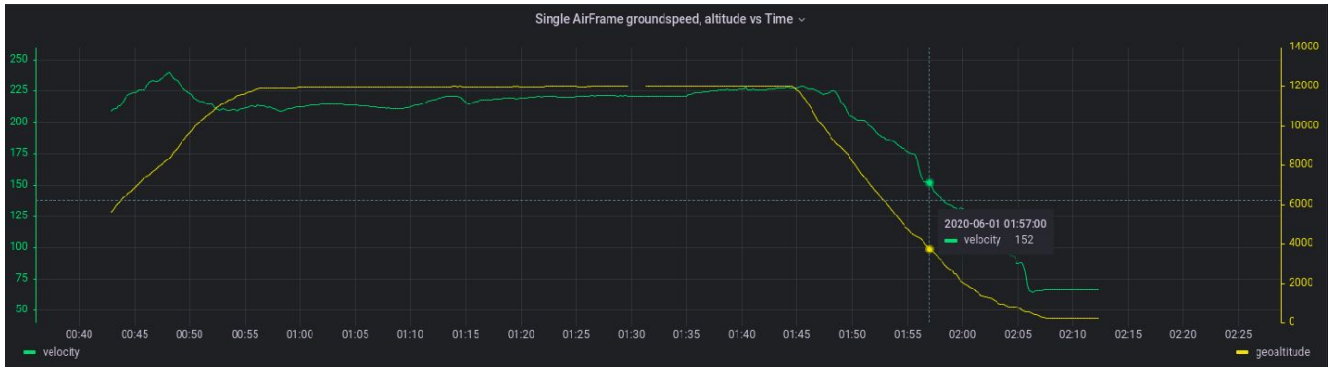
Multiple queries and sources in a single visualization

Tooltips and additional information on mouse hover

Visualization options set in Grafana GUI

```
-- Query A
SELECT et_ts AS "time", velocity
FROM flights
WHERE icao24 = 'c827a6' AND callsign='TRA051'
AND $__timeFilter(et_ts)
```

```
-- Query B
SELECT et_ts AS "time", geoaltitude
FROM flights
WHERE icao24 = 'c827a6' AND callsign='TRA051'
AND $__timeFilter(et_ts)
```



Creating Flight Trajectories

Problem: How can we isolate an airplane + flight combination?

1. Retrieve the start and end time of each flight
2. Use those times to slice other values

```
CREATE TABLE flight_traj (icao24, callsign, flight_period, trip, vertrate) AS
WITH airframe_callsign_period AS (
    SELECT icao24, trip, vertrate,
        startValue(unnest(segments(callsign)))  AS start_callsign,
        unnest(segments(callsign))::period      AS callsign_period
    FROM airframe_traj )
SELECT icao24, callsign, start_callsign,
    callsign_period                             AS flight_period,
    atPeriod(trip, callsign_period)             AS trip,
    atPeriod(vertrate, callsign_period)         AS vertrate,
FROM airframe_callsign_period;
```

For all flights taking off at any given time, how did the vertical ascent rate change over the course of takeoff?

1. Slice trajectory to user defined time-range
2. Slice (1) into periods where 1 < vertical ascent < 20
3. Select first ascent period
4. Unnest values from array to discrete points for visualization

```
WITH
flight_traj_time (icao24, callsign, t_trp, t_alt, t_vrt) AS (
  SELECT icao24, callsign,
    atPeriod(trip, period '[$__from:date, $__to:date)'),
    atPeriod(geoaltitude, period '[$__from:date, $__to:date)'),
    atPeriod(vertrate, period '[$__from:date, $__to:date)')
  FROM flight_traj ),
flight_traj_asc(icao24, callsign, asc_trip, asc_altitude, asc_vrate) AS (
  SELECT icao24, callsign,
    atPeriod(t_trp, period(sequenceN(atRange(t_vrt,
floatrange'[1,20]'),1))),
    atPeriod(t_alt, period(sequenceN(atRange(t_vrt,
floatrange'[1,20]'),1))),
    atPeriod(t_vrt, period(sequenceN(atRange(t_vrt,
floatrange'[1,20]'),1)))
  FROM flight_traj_time ),
final_output AS (
  SELECT icao24, callsign,
    getValue(unnest(instants(asc_altitude)))    AS altitude,
    getValue(unnest(instants(asc_vrate)))       AS vertrate,
    ST_X(getValue(unnest(instants(asc_trip))))  AS lon,
    ST_Y(getValue(unnest(instants(asc_trip))))  AS lat
   FROM flight_traj_asc )
SELECT *
FROM final_output
WHERE vertrate IS NOT NULL AND altitude IS NOT NULL;
```

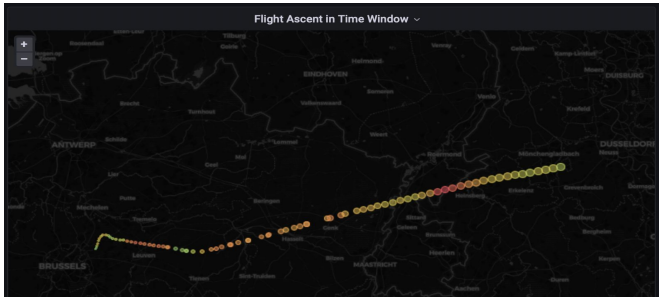
For all flights taking off at any given time, how did the vertical ascent rate change over the course of takeoff?

We unnest arrays into discrete points as we cannot use geoms in Grafana

Marker size reflects altitude

Color represent ascent rate

Visualizing 5 dimensions



How much airplane traffic is there at Amsterdam at any given time in the day?

1. Create geographic boundary for city of Amsterdam
2. Create time slice
3. Return all slices that intersect with the geographic boundary

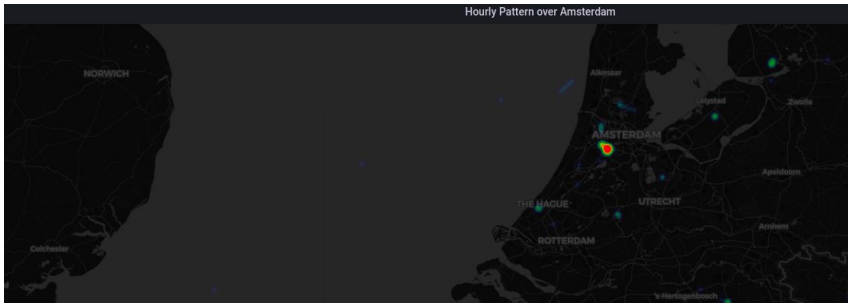
```
WITH
City(Amsterdam) AS (
  SELECT ST_MakeEnvelope(3.409884, 51.246014, 7.103755, 52.680961,
4326)),
-- Clip all temporal columns to the user-specified time range.
flight_traj_time_slice (icao24, callsign, time_slice_trip) AS (
  SELECT icao24,callsign,
    atPeriod(trip, period '[${__from:date}, ${__to:date})')
  FROM flight_traj TABLESAMPLE SYSTEM (70)),
-- Clip all the result that are outside the specified window.
clipped_flight AS (
  SELECT icao24, callsign, time_slice_trip
  FROM flight_traj_time_slice, Location
  WHERE intersects(time_slice_trip, Amsterdam ) )
SELECT
  icao24, callsign,
  getTimestamp ( unnest(instants(time_slice_trip)))  AS et,
  ST_Y(getValue( unnest(instants(time_slice_trip)))) AS lat,
  ST_X(getValue( unnest(instants(time_slice_trip)))) AS lon
FROM clipped_flight;
```

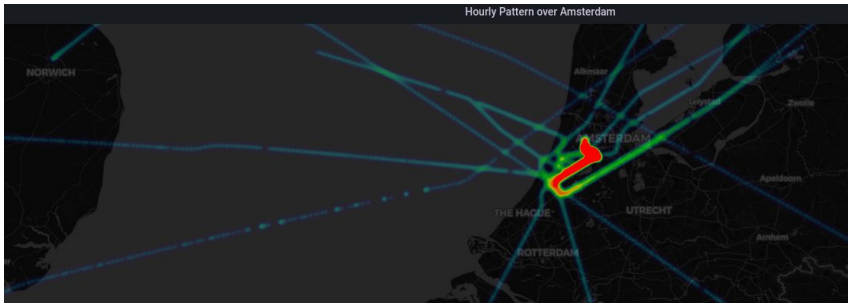
How much airplane traffic is there at Amsterdam at any given time in the day?

Heatmap visualization with 1-hour time window shows 10am and 11am

Geographic slices can be pre-configured as Grafana variables and injected directly into query

Areas can be defined as airports, functional control blocks or other geographies of interest





Demo

MEOS

```
int main() {
    set<TInstant<int>> instants = {
        TInstant<int>(2, unix_time_point(2012, 1, 1)),
        TInstant<int>(1, unix_time_point(2012, 1, 2)),
        TInstant<int>(4, unix_time_point(2012, 1, 3)),
        TInstant<int>(3, unix_time_point(2012, 1, 4)), };
    TSequence<int> tseq(instants);
    cout << tseq << endl;
    return 0;}
```

PyMEOS

```
trajectory = TGeomPointSeq({
    TGeomPointInst(GeomPoint(0, 0), datetime_utc(2012, 1, 1, 8, 0)),
    TGeomPointInst(GeomPoint(2, 0), datetime_utc(2012, 1, 1, 8, 10)),
    TGeomPointInst(GeomPoint(2, 1), datetime_utc(2012, 1, 1, 8, 15)),})
```

movingpandas

```
df = pd.DataFrame([
  {'geometry':Point(0,0), 't':datetime(2018,1,1,12,0,0)},
  {'geometry':Point(2,0), 't':datetime(2018,1,1,12,10,0)},
  {'geometry':Point(2,1), 't':datetime(2018,1,1,12,15,0)} ]).set_index('t')
gdf = GeoDataFrame(df, crs=31256)
toy_traj = mpd.Trajectory(gdf, 1)
```

Conclusions

MobilityDB

- PostgreSQL extension, open-source
- Spatiotemporal data types and functions
- 3:1 compression for OpenSky
- Generic trajectories with no flight specific functionality
- Extension - Develop functions to import common geographic envelopes

Grafana

- Dashboard tool, open-source
- Accepts multiple data sources used concurrently
- Native map visualization and time-series panels
- Cannot process trajectories directly
- Extension - Develop native trajectory visualizations and time-series animations