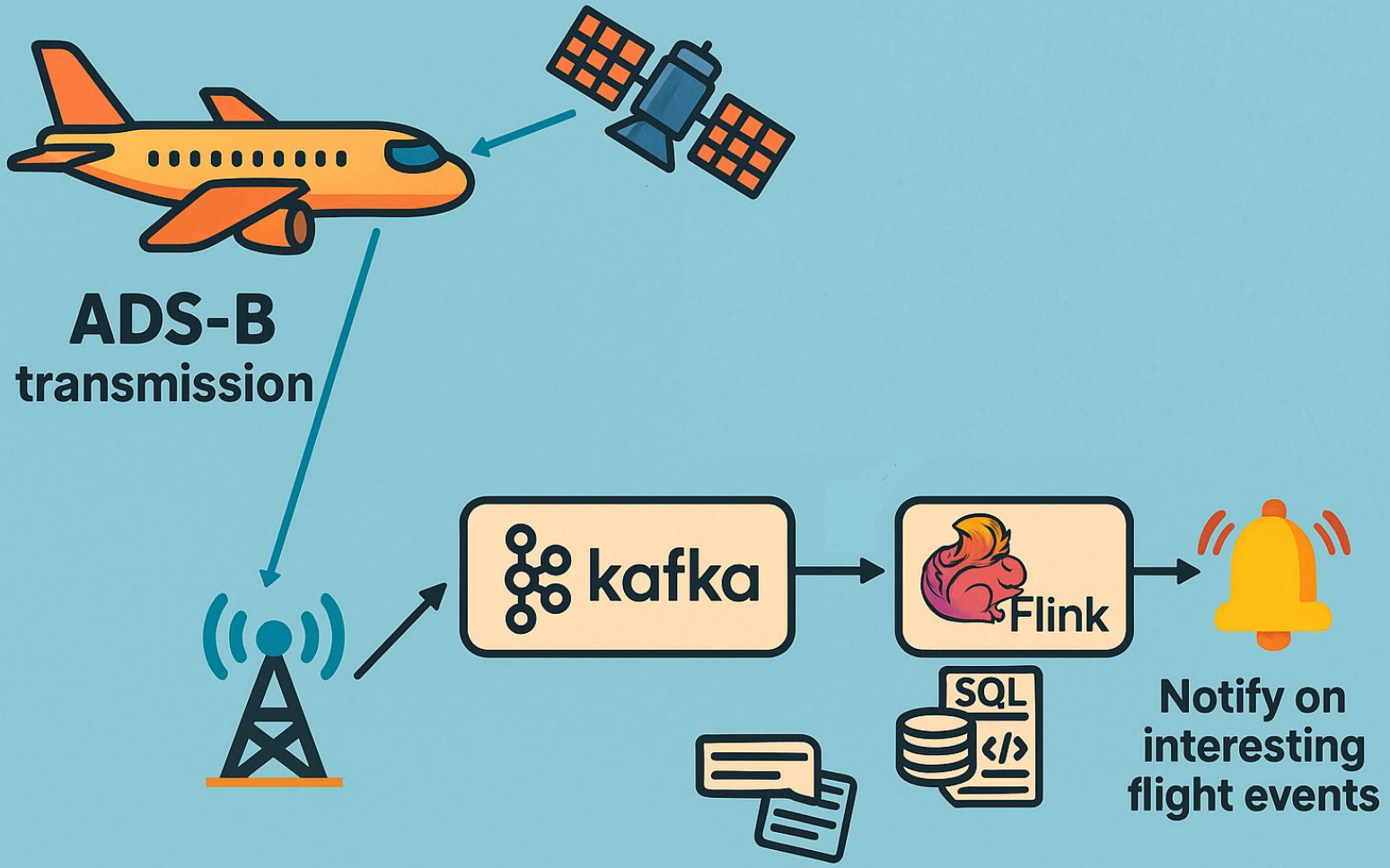
Complex event processing of ADS-B aviation data with Apache Flink

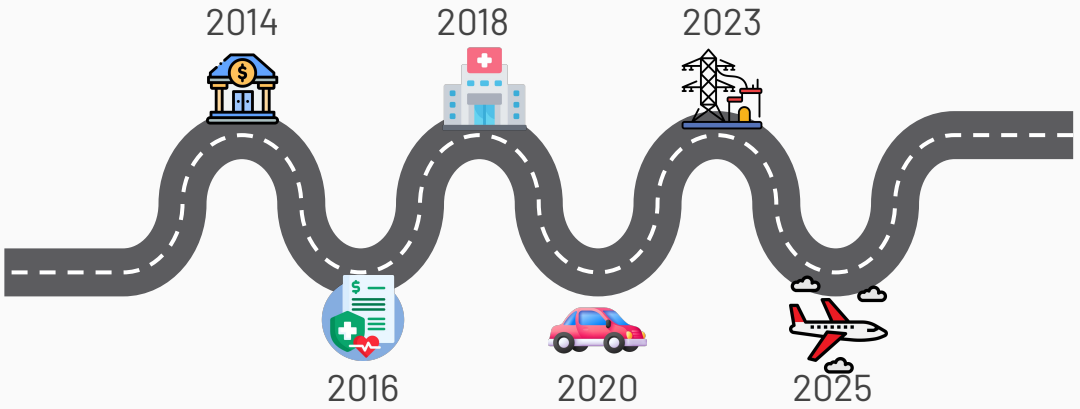
Apache Kafka x Apache Flink Meetup - July 2025

@SimonAubury


ADS-B
transmission
kafka
Flink
SQL
Notify on
interesting
flight events

Simon Aubury

Data Geek



@SimonAubury

Can I find

1. Missed approach (or go-around) during aircraft landing

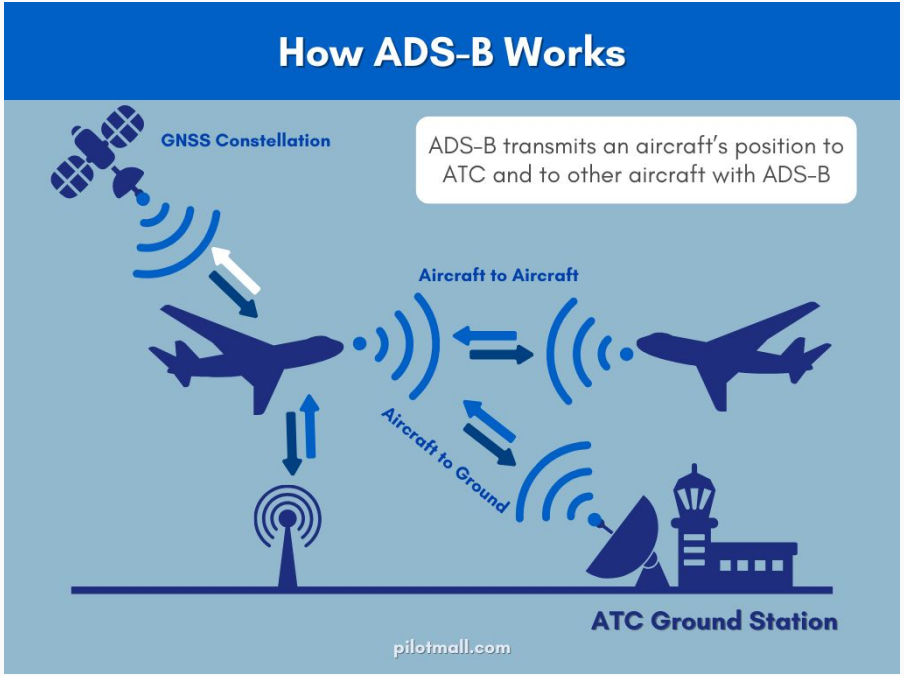
2. Paired flight paths where the same aircraft land (or takeoff) on parallel runways.

Tracking planes
ADS-B

ADS-B

Automatic Dependent Surveillance Broadcast, or ADS-B provides an updated way for pilots to view each other's positions and for ATC to manage air traffic.

- Position (GPS)
- Altitude
- Velocity
- Heading



https://www.pilotmall.com/

ADS-B capture using a Raspberry Pi

To capture the aircraft transmissions

- Raspberry Pi
- RTL2832U USB dongle
- dump1090
- A small antennae

ADS-B .. with curl

adsb.fi - a community-driven flight tracker project with a free real-time API

```
curl https://opendata.adsb.fi/api/v2/lat/-33.94/lon/151.17/dist/5
```

Flights → AVRO → Kafka

`monitor_opendata.py --airport SYD`

Poll adsb.fi every 10s

Serialise any flights found as AVRO into the **`flights`** topic

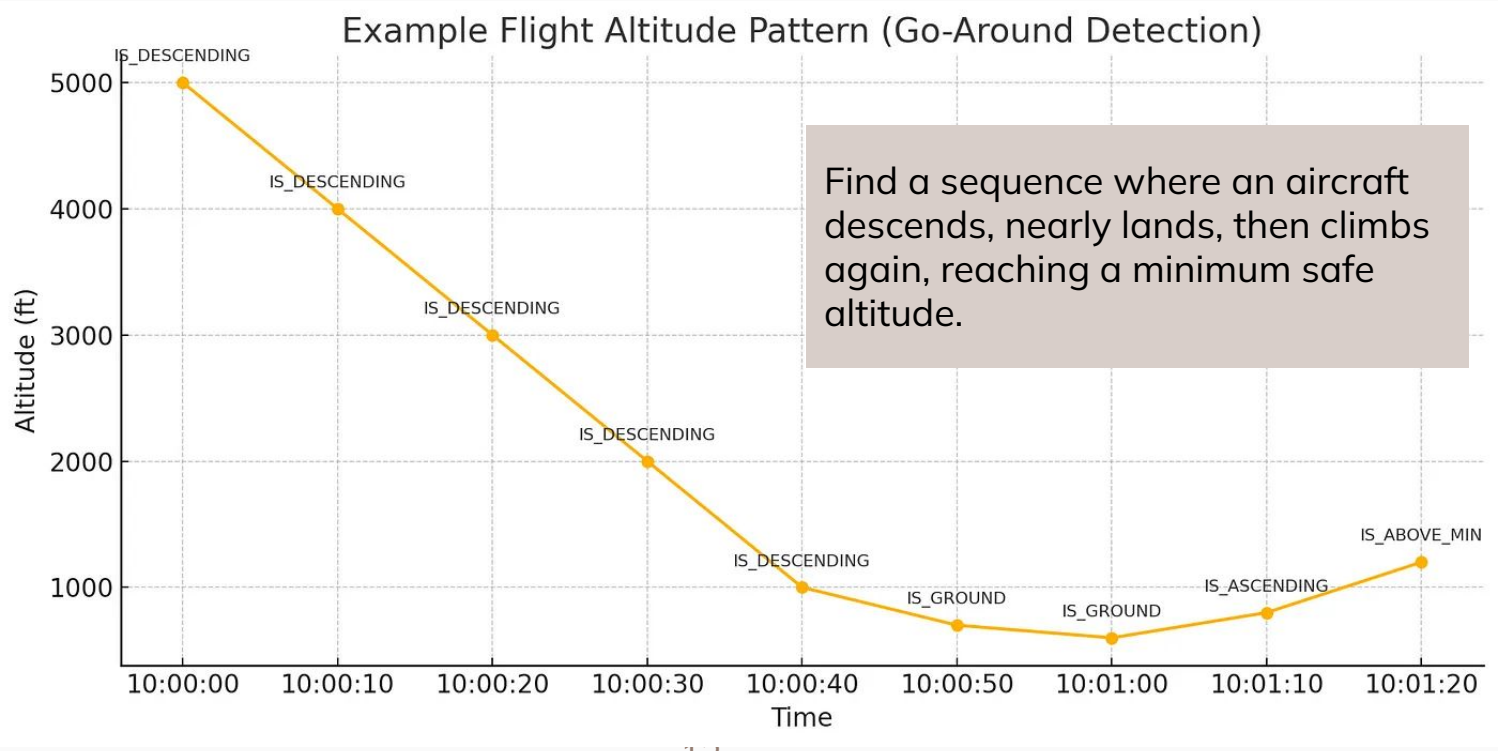
```python
def get_flights(point_lat, point_lon, dist):
    url = f'https://opendata.adsb.fi/api/v2/lat/{point_lat}/lon/{point_lon}/dist/{dist}'
    response = requests.get(url)
    response.raise_for_status()
    return response.json().get('aircraft', [])

def process_flight(file_name, flight, prd):
    # Ignore anything with a ground speed below 50 knots
    if (flight.get('gs') is None or flight.get('gs') < 50):
        return

    formatted_timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    data = f'{formatted_timestamp},{flight.get('hex')},{flight.get('flight')},{flight.get(
    with open(file_name, 'a') as f:
        f.write(f'{data}\n')

    if prd is not None:
        flight = Flight(callsign=flight.get('flight').rstrip(),
                        latitude=flight.get('lat'),
                        longtitude=flight.get('lon'),
                        altitude=flight.get('alt_geom'),
                        icao=flight.get('hex'),
                        speed=flight.get('gs'),
                        airport='' ,
                        track=flight.get('track'),
                        sqwark=flight.get('squawk'),
                        emergency=flight.get('emergency'),
                        )
        prd.do_produce(flight)
```

https://github.com/saubury/plane_track/blob/main/monitor_opendata.py

Missed approach finding with Flink

A missed approach (go-around) is a procedure where an aircraft, during an approach to landing, discontinues the approach and climbs away from the runway.

Missed approach



Find a sequence where an aircraft descends, nearly lands, then climbs again, reaching a minimum safe altitude.

Flink SQL

Flink SQL query with **`MATCH_RECOGNIZE`**,

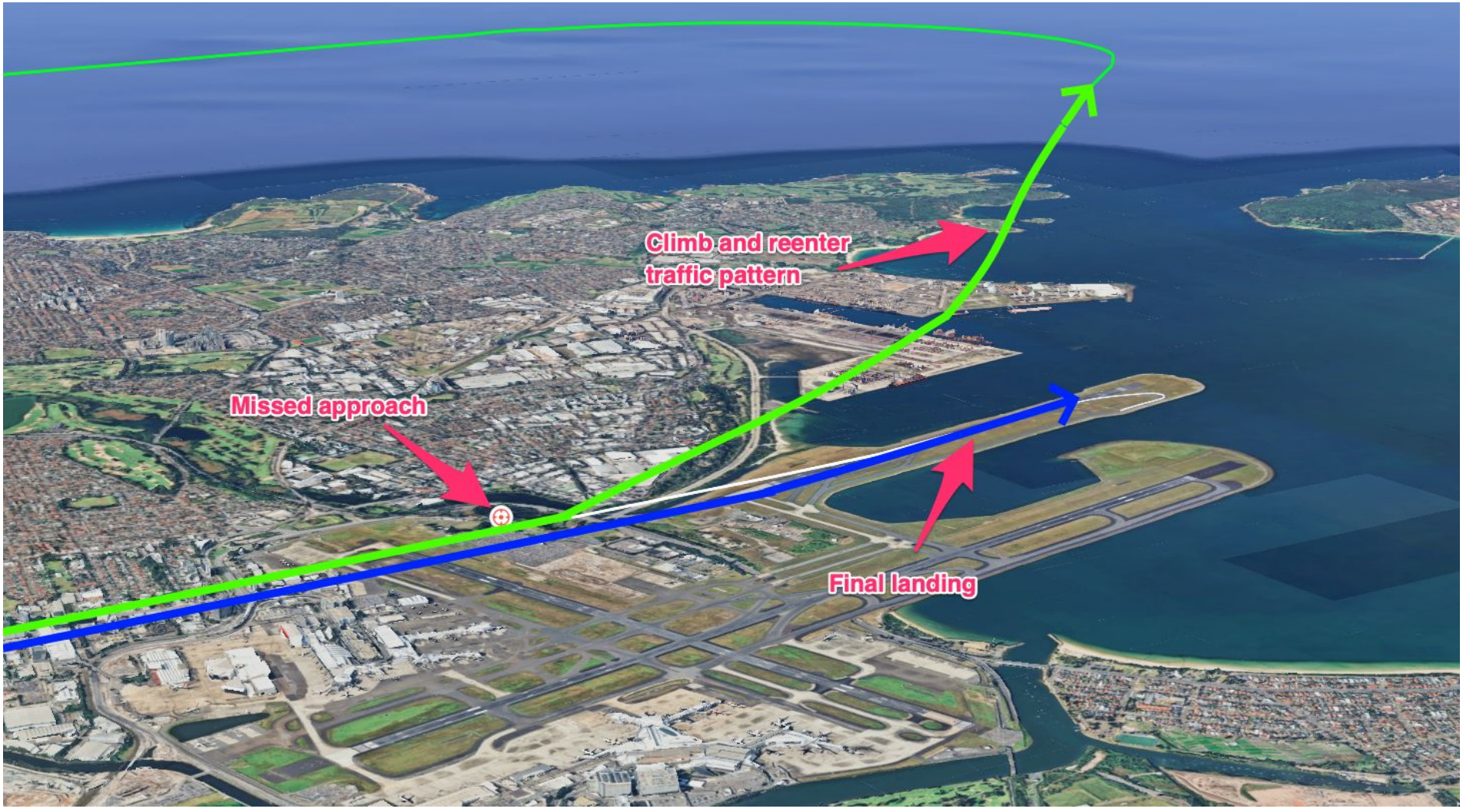
Identifies specific flight altitude patterns in a stream of aircraft data, partitioned by callsign

```
SELECT *
FROM flight
MATCH_RECOGNIZE(
    PARTITION BY callsign
    ORDER BY proc_time
    MEASURES
        IS_DESCENDING.flightts as desc_UTC,
        IS_GROUND.flightts as ground_UTC,
        IS_ASCENDING.flightts AS asc_UTC,
        IS_ABOVE_MIN.flightts AS abvm_UTC,
        IS_GROUND.altitude AS grd_altitude,
        IS_ASCENDING.altitude AS asc_altitude
    ONE ROW PER MATCH
    AFTER MATCH SKIP TO LAST IS_ASCENDING
    PATTERN (IS_DESCENDING{5,} IS_GROUND{1,} IS_ASCENDING IS_ABOVE_MIN)
    DEFINE
        IS_DESCENDING AS (LAST(altitude, 1) IS NULL
           AND altitude >= 1000) OR altitude < LAST(altitude, 1),
        IS_GROUND AS altitude <= 800,
        IS_ASCENDING AS altitude > last(altitude,1),
        IS_ABOVE_MIN AS altitude > 1000
) AS T
where TIMESTAMPDIFF(second, desc_UTC, asc_UTC) between 0 and 1000;
```

SQL Query Result (Table)
Table program finished. Page: Last of 1 Updated: 03:49:28.074

callsign	desc_UTC
ANZ101	27 Apr 2025

https://github.com/saubury/plane_track

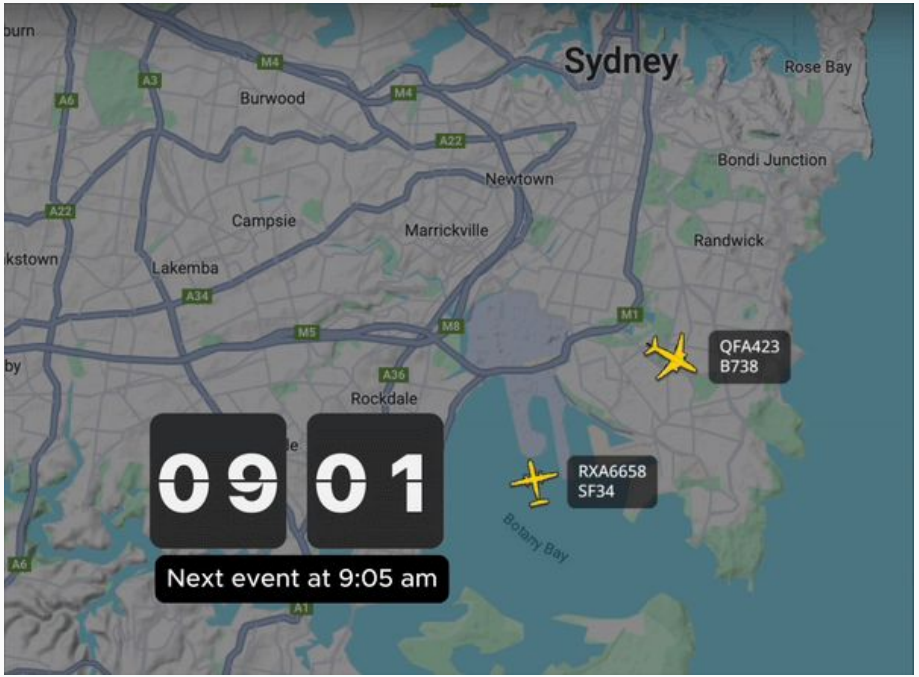

Climb and reenter
traffic pattern
Missed approach
Final landing


Google Earth
Airbus, Vexcel Imaging US, Inc

Finding paired flights with Flink

Twin flying

Paired flight landings occur when aircraft land on parallel runways.



Flink UDF's

To measure the distance between aircraft using a Flink user-defined function (UDF)

By adding a distance scalar Java function I could calculate distance between two aircraft.

```java
public class Distance extends ScalarFunction {
    public static final String NAME = "calculateDistanceKm";

    // Equirectangular approximation to calculate distance in km between two points
    public float eval(float lat1, float lon1, float lat2, float lon2) {
        float EARTH_RADIUS = 6371;
        float lat1Rad = (float) Math.toRadians(lat1);
        float lat2Rad = (float) Math.toRadians(lat2);
        float lon1Rad = (float) Math.toRadians(lon1);
        float lon2Rad = (float) Math.toRadians(lon2);

        float x = (float) ((lon2Rad - lon1Rad) * Math.cos((lat1Rad + lat2Rad) / 2));
        float y = (lat2Rad - lat1Rad);
        float distance = (float) (Math.sqrt(x * x + y * y) * EARTH_RADIUS);

        return distance;
    }
}
```

```
add jar '/target-jars/udf_example-1.0.jar';

CREATE FUNCTION distancekm  AS 'com.example.my.Distance';

SHOW USER FUNCTIONS;

-- Straight line distance (km) between Sydney and Melbourne
SELECT distancekm(-33.9401302, 151.175371, -37.840935, 144.946457)
as dist_km;
```

https://github.com/saubury/plane_track

Paired

Find pairs of flights that were geographically close (within 1.5 km) of each other during overlapping or near-overlapping times (within 35 seconds)

```
SELECT f1.callsign AS f1,
f2.callsign AS f2,
distancekm(f1.latitude , f1.longtitude, f2.latitude, f2.longtitude) as km
FROM flight f1, flight f2
WHERE f1.flightts BETWEEN f2.flightts - interval '35' SECOND AND f2.flightts
AND f1.callsign < f2.callsign
AND distancekm(f1.latitude , f1.longtitude, f2.latitude, f2.longtitude) < 1.5;
```

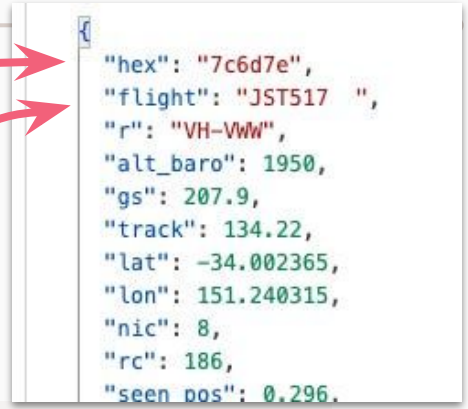
SQL Query Result (Table)
efresh: 1 s Page: Last of 12 Updated: 1

f1	f2	km
QFA749	TGW13	1.3
QFA749	TGW13	1.1
ANZ109	JST724	1.3
ANZ109	JST613	1.3

https://github.com/saubury/plane_track

What about …
the type of plane?

Annotating



ICAO codes

Aircraft ICAO codes such as 7C6D7E to airframes such as Airbus A321-231 which is 16 years old

Flight codes

Flight code such as JST517 departs Sydney at 12:40 PM and arrives in Melbourne at 2:15 PM operating daily.

Opensky data archive

600k airframes

```
7c8155,Australia,Eagle R & D,Helicycle,Private,VH-ZTZ,UHEL
7c8160,Australia,Beech Aircraft Corp.,76 Duchess,Air Gold
7c8162,Australia,North American,T-28D Trojan,Private,VH-ZU
7c8163,Australia,Piper,PA-44-180 Seminole,Airflite Pty Lto
7c8164,Australia,Robinson,R22 Beta II,Aeroair Pty Ltd,VH-Z
7c8165,Australia,Robinson,R44 Raven II,Private,VH-ZUF,R44
7c8166,Australia,Robinson,R44 Raven II,Private,VH-ZUG,R44
7c8167,Australia,Sikorsky,S-92 A,Bristow Helicopters Austr
7c8168,Australia,Piper
7c8169,Australia,Sikor
7c816a,Australia,North
7c816c,Australia,Piper
7c816d,Australia,Bell,
7c816e,Australia,Sikor
7c8170,Australia,Sikor
7c8171,Australia,Van''
7c8173,Australia,Robin
7c8174,Australia,,,,VH
7c8175,Australia,Sikor
7c8176,Australia,Sikor
7c8178,Australia,Bell,
7c8184,Australia,Eurocc
7c8185,Australia,Eurocc
7c8186,Australia,Airbu
```

```
CREATE TABLE aircraft_lookup (
    icao24  varchar(100) not null,
    country  varchar(100),
    manufacturerName varchar(100),
    model varchar(100),
    owner varchar(100),
    registration varchar(100),
    typecode varchar(100)
) WITH (
    'connector' = 'filesystem',
    'path' = '/data_csv/aircraft_lookup_aus.csv',
    'format' = 'csv'
);
```

420k route

```
VOZ631,CBR-SYD
VOZ632,SYD-CBR
VOZ633,CBR-SYD
VOZ634,SYD-CBR
VOZ635,CBR-SYD
VOZ636,SYD-CBR
VOZ637,CBR-
VOZ638,SYD-
VOZ640,SYD-
VOZ641,CBR-
VOZ642,SYD-
VOZ643,CBR-
VOZ644,SYD-
VOZ645,CBR-
VOZ646,SYD-CBR
VOZ647,CBR-SYD
```

```
CREATE TABLE route_lookup (
    flight  varchar(100) not null,
    route  varchar(100)
) WITH (
    'connector' = 'filesystem',
    'path' = '/data_csv/flight_route_syd.csv',
    'format' = 'csv'
);
```

Same plane types

```
CREATE OR REPLACE view flight_decorated
as
select f.*, a.model, a.owner, a.typecode, r.route
from flight f
left join aircraft_lookup a on (f.icao = a.icao24)
left join route_lookup r on (f.callsign = r.flight)
;
```

```
SELECT f1.flightts,
f1.callsign || ' ('  || COALESCE(f1.route, '-') ||')' || ' ' || f1.typecode
CAST(ROUND(DISTANCEKM(f1.latitude , f1.longtitude, f2.latitude, f2.longtitu
f2.callsign || ' ('  || COALESCE(f2.route, '-') ||')' || ' ' || f2.typecode
FROM flight_decorated f1, flight_decorated f2
WHERE f1.flightts BETWEEN f2.flightts - interval '20' SECOND AND f2.flightt
AND f1.callsign < f2.callsign
AND f1.typecode = f2.typecode
AND DISTANCEKM(f1.latitude , f1.longtitude, f2.latitude, f2.longtitude) < 1
```

https://github.com/saubury/plane_track

TLDR …

Take-away's

Stream analysis

✈️ ADS-B signals provide a rich data source that, when processed in real-time can find cool events

CEP makes sense of data.

🛠️ Apache Flink can detect patterns in event streams .. and slow to spot with batch analytics.

Open-source + curiosity

🧪 With just a basic radio receiver & streaming framework you can build meaningful analytics pipelines

Apply beyond aviation

🌍 The techniques for identifying anomalies or sequences are transferable to other industries

THANKS!

Do you have any questions?

Slides

Blog / code

@SimonAubury

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**