Real-Time Vessel Monitoring with Open-Source Tools: Leveraging PostgreSQL, PostGIS, and Python for Geospatial Analysis and Visualisation

FOSS4G:UK

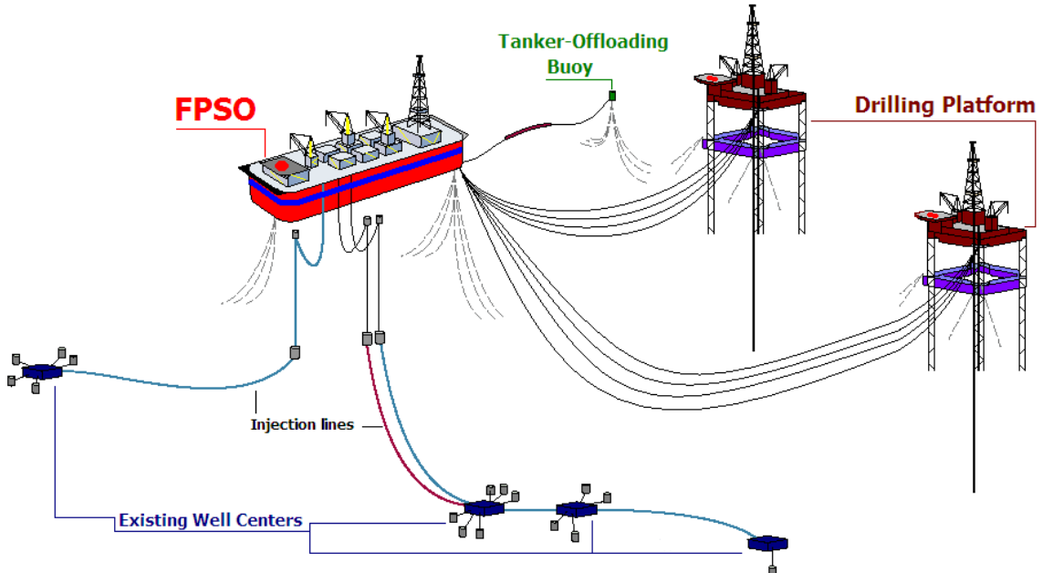
12th November 2024

Rob Burgess

Introduction

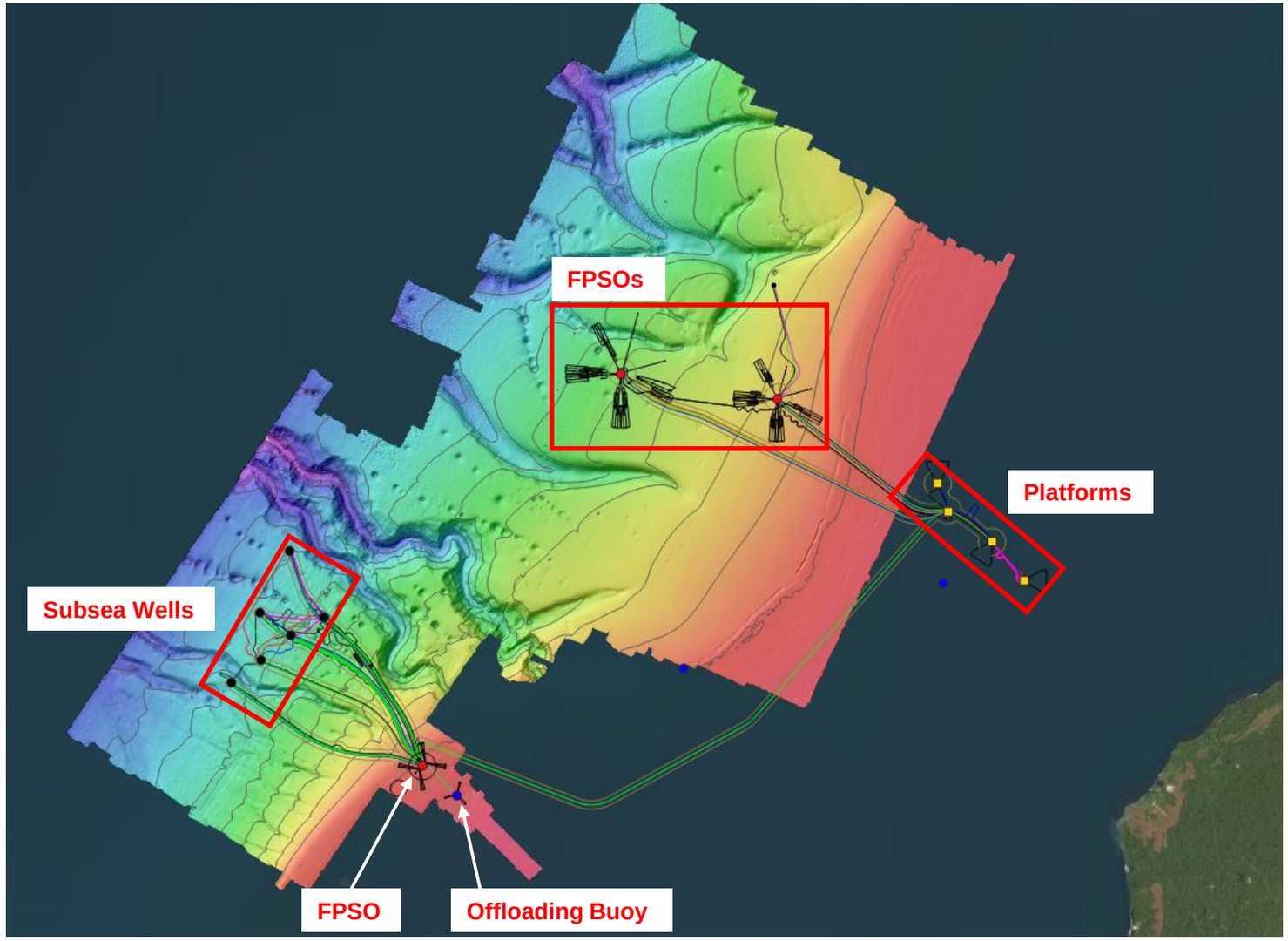
- Offshore energy project
- Facilities include platforms, FPSOs, subsea well clusters, and pipelines.

Floating production storage and offloading



Introduction

- Collection of energy facilities 25km off the coast of Equatorial Guinea, spanning 450 sq km.



Introduction

- Mooring Masters oversee vessel mooring and coordination.
- Tracking is crucial for efficient operations and safety.
- Need to monitor several key aspects of vessel movement:
  - The approach of vessels to the region and their safe arrival
  - The proximity of vessels to platforms and FPSOs, especially given presence of exclusion zones
  - Has a vessel strayed into a restricted zone?
  - Has a vessel spent too much time within a zone?
  - Has a vessel safely returned to port?
  - Is the vessel stationary or on the move?

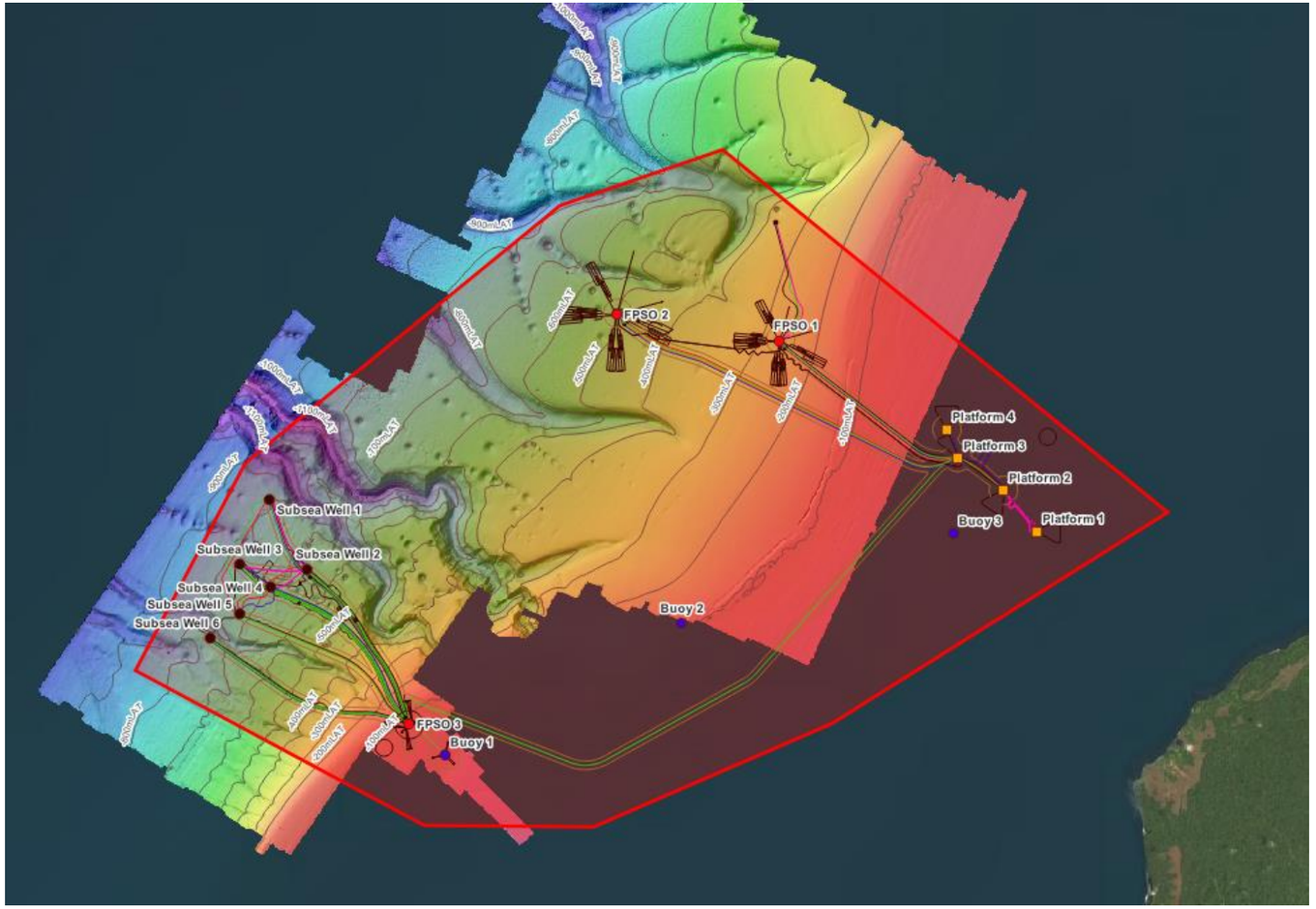
AIS

- AIS (Automatic Identification System) for real-time vessel tracking
- Providers include MarineTraffic, Spire, and platform-based systems.
- AIS data typically updates every few minutes (variable).
- Key vessels tracked: Red Snapper, Red Fox, Siem Day
- Mooring Masters can use this data to track vessel movements

Geofences

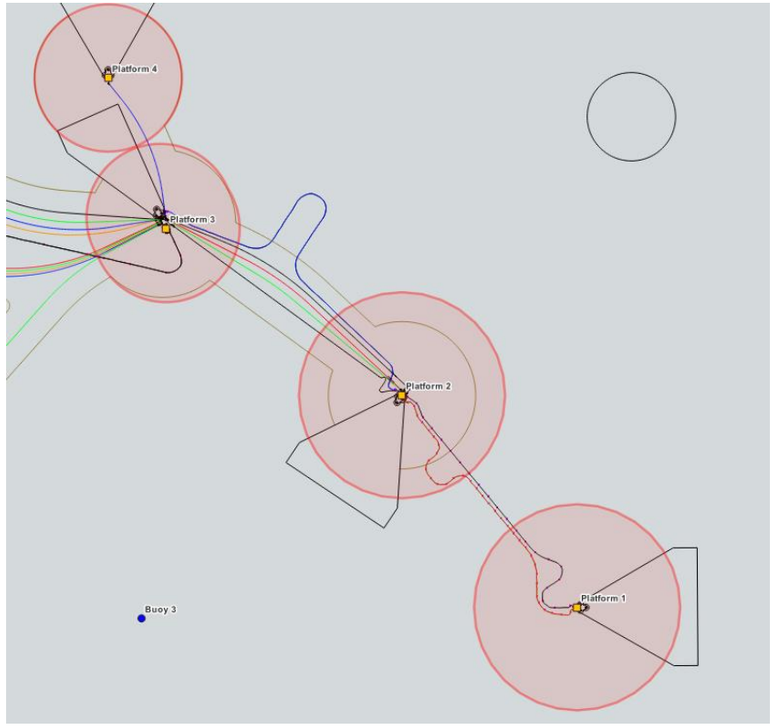
- Create geofences around field, facilities and ports

In-Field Geofence – Vessels would be classified as `in_field`

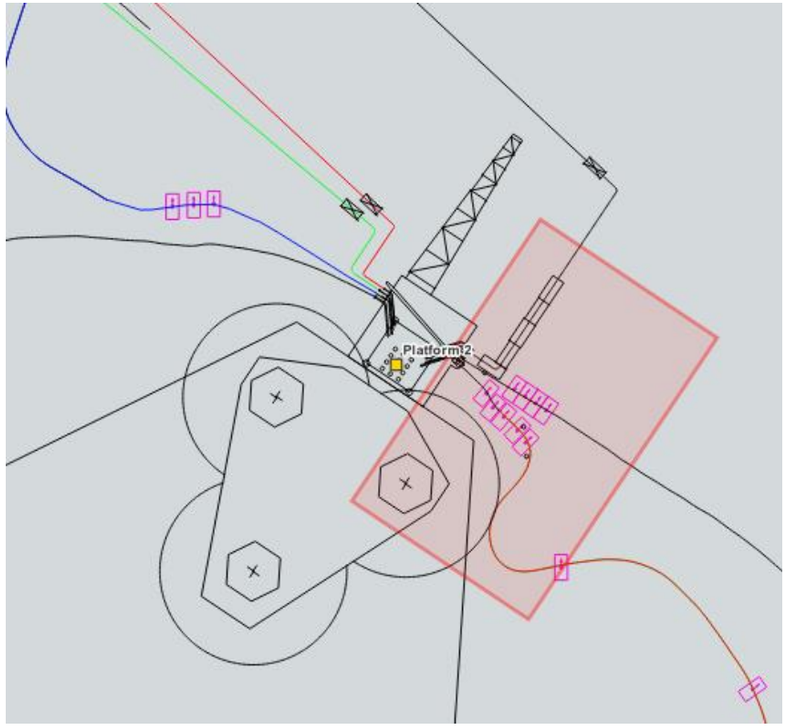


Geofences

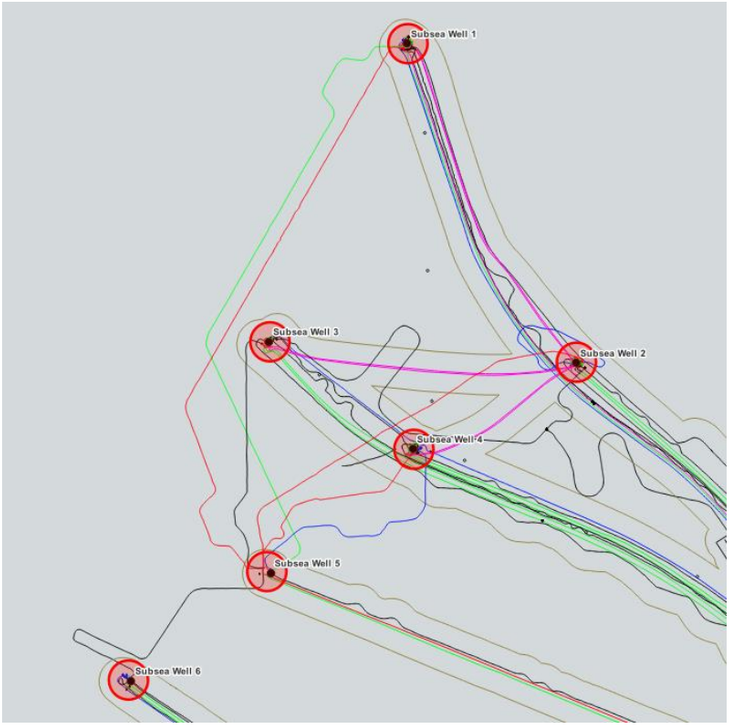
Platforms



Platform Loading / Unloading Areas



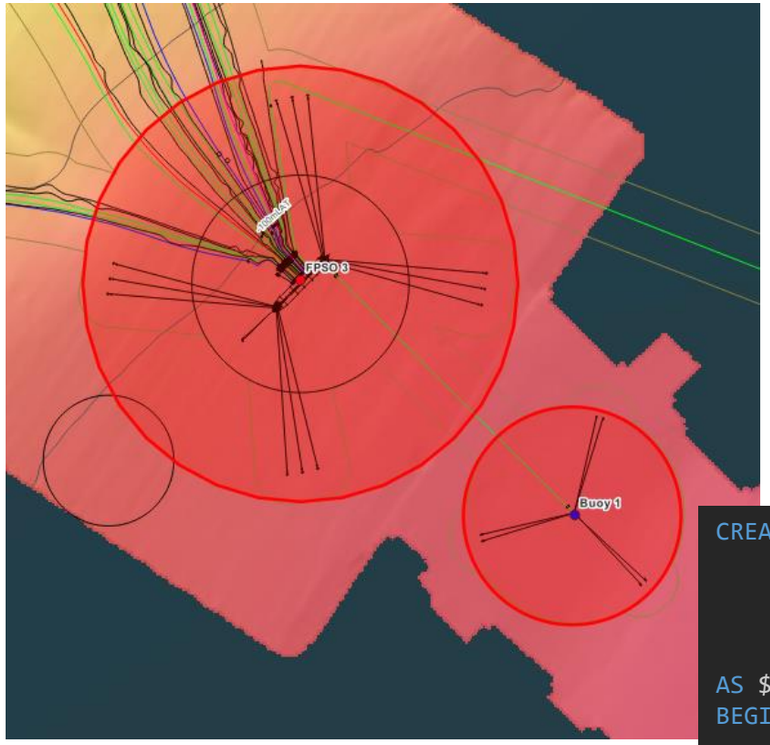
Subsea Well Clusters



Vessels would be classified as being within a `project` geofence

Geofences

Dynamic Geofences - Vessels would be classified as being within a `project` geofence




```sql
CREATE OR REPLACE FUNCTION public.update_geofence_geometry()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN


  IF NEW.mmsi = 566983000 THEN
    -- Create 700m buffer
    UPDATE geofences
    SET the_geom = ST_Transform(ST_Buffer(ST_Transform(NEW.the_geom, 32630), 700), 4326)
    WHERE location = 'Noble Venturer';

  END IF;

  RETURN NEW;
END;
$BODY$;
```

Geofences

Port Geofences – Vessels would be classified as `in_port`

Postgres LISTEN/NOTIFY

- Enables real-time, event driven communication between the database and external applications.
- LISTEN: we can set up a listener for incoming data
- NOTIFY: send a message to trigger an action in the external application

```
CREATE OR REPLACE FUNCTION public.notify_new_position()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
DECLARE
    vname TEXT;
    vtimestamp TEXT;
    vgeom BYTEA;
    vspeed TEXT;
    json_message TEXT;
BEGIN
    SELECT vessel_name INTO vname
    FROM geofence_vessels
    WHERE mmsi = NEW.mmsi AND enabled = true;

1   -- Get timestamp, the_geom in WKB format, and speed
    SELECT NEW.timestamp::text, ST_AsEWKB(NEW.the_geom), NEW.attrs ->> 'speed' INTO vtimestamp, vgeom, vspeed;

2   IF vname IS NOT NULL THEN
        json_message := json_build_object(
            'vessel_name', vname,
            'mmsi', NEW.mmsi,
            'timestamp', vtimestamp,
            'geom', encode(vgeom, 'hex'),
            'speed', vspeed
        )::text;

3       PERFORM pg_notify('vessel_event_channel', json_message);
    END IF;
    RETURN NEW;
END;
$BODY$;
```

Example Payload

```
{
  "vessel_name": "Topaz Commander",
  "mmsi": "538006537",
  "timestamp": "2024-10-10T09:33:30Z",
  "geom": "0101000020E61000005839B4C87…",
  "speed": "0.7"
}
```

Python Listener

- Create a listener

Setting up the listener

```
def listen_for_geofencing_notifications():
    with get_db_connection() as conn:
        cursor = conn.cursor()
        cursor.execute("LISTEN vessel_event_channel;")
        print("Waiting for notifications on channel 'vessel_event_channel'")
```

Handling notifications

```
    conn.poll()
    while conn.notifies:
        notify = conn.notifies.pop(0)
        payload = notify.payload

        try:
            data = json.loads(payload)
            vessel_name = data.get('vessel_name')
            mmsi = data.get('mmsi')
            timestamp = data.get('timestamp')
            geom = data.get('geom')
            speed = data.get('speed')
            print(f"Got NOTIFY for MMSI: {mmsi}, Vessel Name: {vessel_name}, Timestamp: {timestamp}, Speed: {speed}")
```

```
Got NOTIFY for MMSI: 538006537, Vessel Name: Topaz Commander, Timestamp: 2024-10-10 09:33:30+00, Speed: 0.7
```

Process vessel position

```
process_vessel_geofencing(session, mmsi, vessel_name, timestamp, geom, speed)
```

Identifying Geofence Intersections

- Check for geofence intersections

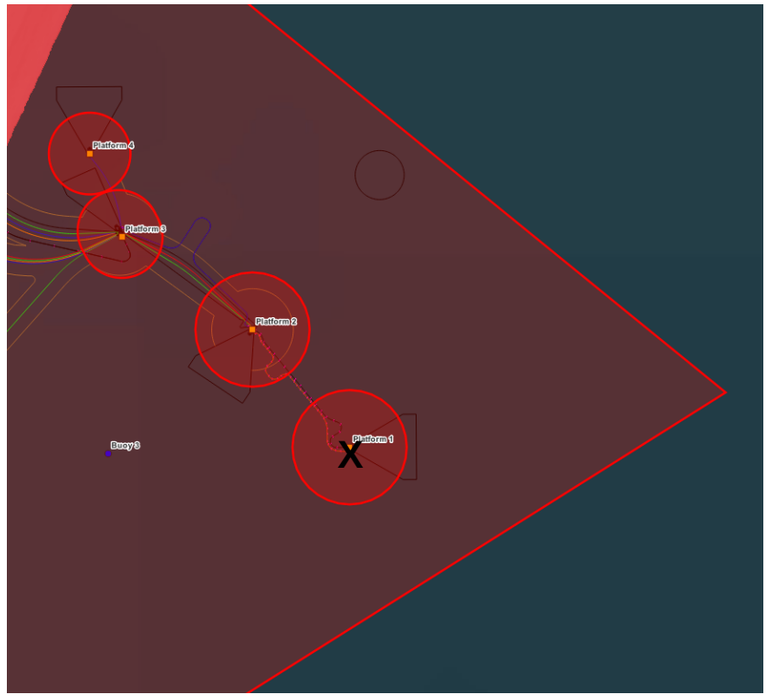
Initialising Geofence Checks

```
geofences = get_geofences(session)

vessel_within_geofence = False
geofence_id = None
geofence_type = None

 # Initialise a list to store geofences that contain the vessel point
containing_geofences = []

for idx, geofence_row in geofences.iterrows():
    geofence_geom_raw = geofence_row['the_geom']
    geofence_geom_bytes = bytes(geofence_geom_raw)
    geofence_geom_proc = wkb.loads(geofence_geom_bytes)
```

Checking Geofence Intersections

```
        if geofence_geom_proc.contains(Point(ves_geom_proc)):
            area = geofence_geom_proc.area  # Calculate the area of the geofence
            containing_geofences.append({
                'id': geofence_row['id'],
                'type': geofence_row['type'],
                'geom': geofence_geom_proc,
                'area': area
            })
```

Identifying Geofence Intersections



Example Data Structure

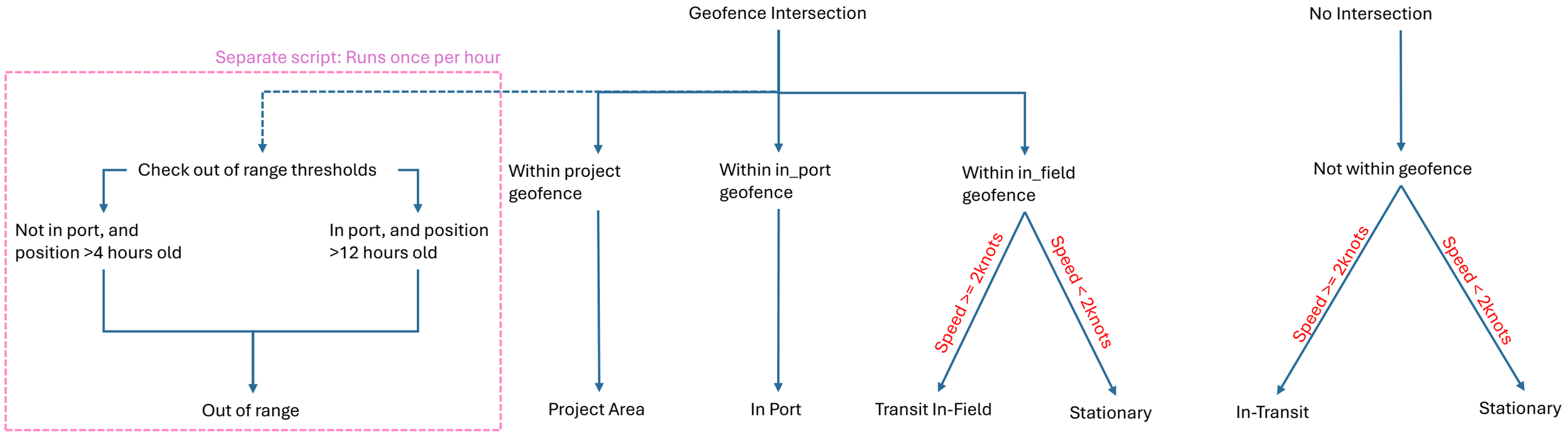
```
containing_geofences = [
    {
        'id': 1,
        'type': ‘project',
        'geom': <shapely.geometry.Polygon object at 0x7f8b0c3eafd0>
        'area': 1500.5
    },
    {
        'id': 2,
        'type': ‘in_field',
        'geom': <shapely.geometry.Polygon object at 0x7f8b0c3ea320>
        'area’: 350000
    }
]
```

Select Smallest Geofence

```
if containing_geofences:
    # Sort the geofences by area (ascending order)
    containing_geofences.sort(key=lambda x: x['area'])
    # Select the geofence with the smallest area
    selected_geofence = containing_geofences[0]
    vessel_within_geofence = True
    geofence_id = selected_geofence['id']
    geofence_type = selected_geofence['type’]
  …
```

Determining Current State

- Evaluates vessel state based on location within a geofence and vessel's speed.
- Returns specific states such as 'in_port,' 'transit_in_field,' 'stationary,' or 'in_transit' depending on geofence type and movement.



Visualising Data

- See demo

mmsi [pk]	timestamp [pk]	geofence_id	state
538006537	2024-10-04 16:00:00+00	38	project
538006537	2024-10-04 15:50:45+00	577	transit_in_field
538006537	2024-10-04 15:37:38.28+00	53	project
538006537	2024-10-04 15:35:46+00	577	transit_in_field
538006537	2024-10-04 15:30:00+00	43	project
538006537	2024-10-04 15:26:45+00	577	transit_in_field
538006537	2024-10-04 15:08:08.635+00	38	project
538006537	2024-10-04 14:43:40.765+00	577	transit_in_field
538006537	2024-10-04 06:02:02.752+00	24	project
538006537	2024-10-04 01:55:48.291+00		out_of_range
538006537	2024-10-03 21:55:48.291+00	577	stationary
538006537	2024-10-03 21:45:47.086+00	577	transit_in_field
538006537	2024-10-03 15:30:18.806+00		in_transit
538006537	2024-10-03 15:20:07.391+00	12	in_port
538006537	2024-10-03 15:19:09.615+00		in_transit

Conclusions

- Shown how we can use AIS data in combination with open source tools to track vessels around offshore energy facilities
- We sourced AIS data from multiple sources, identified vessels of interest, and created geofences
- Created a workflow to process incoming positions and determine their state

- We can visualise the data we collect using charting / dashboards

Thank you!

Select Vessel

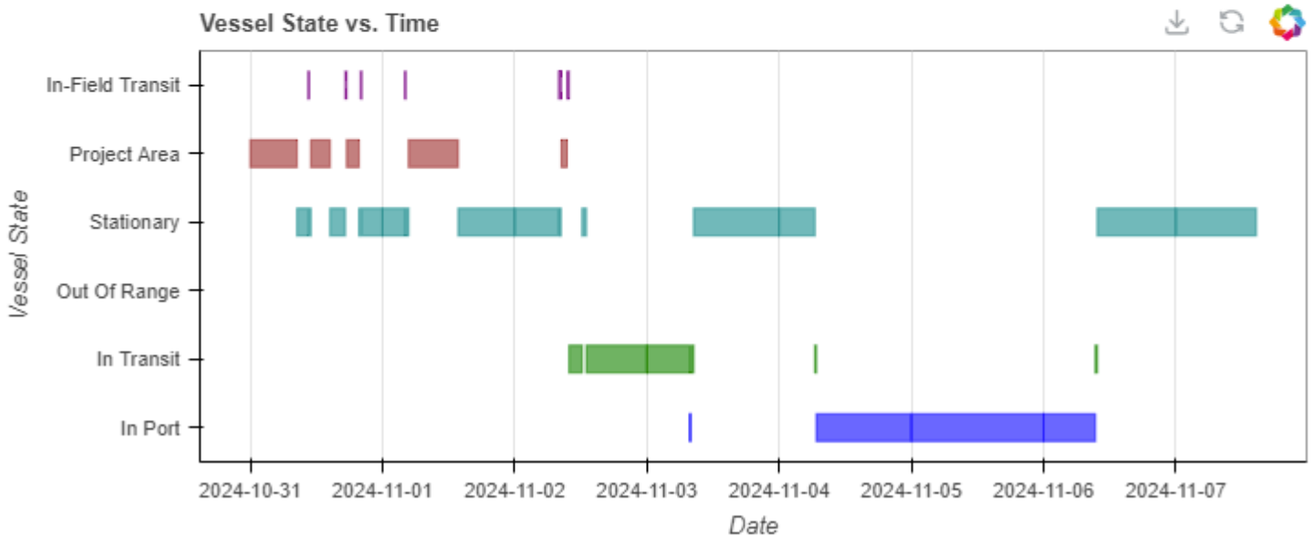
Fram Princess

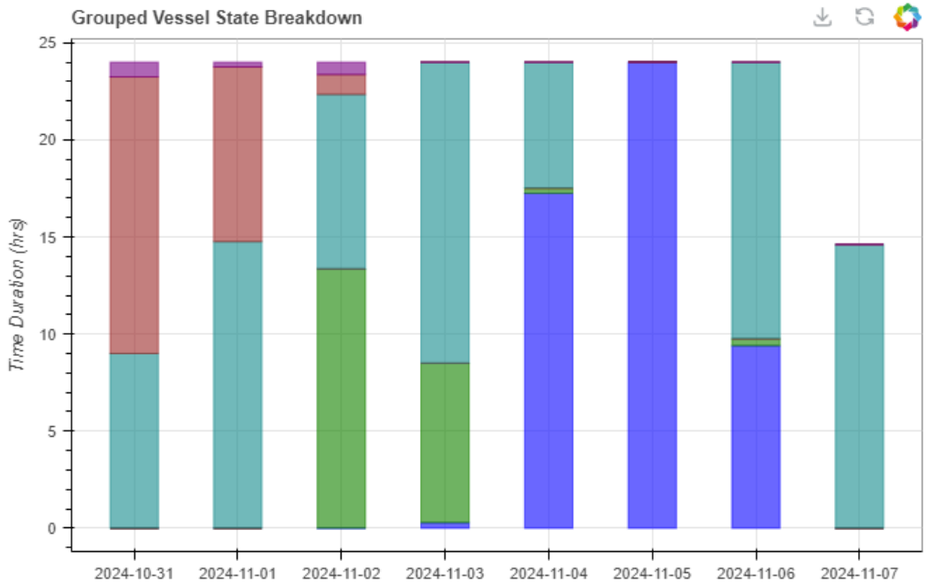
Date Range

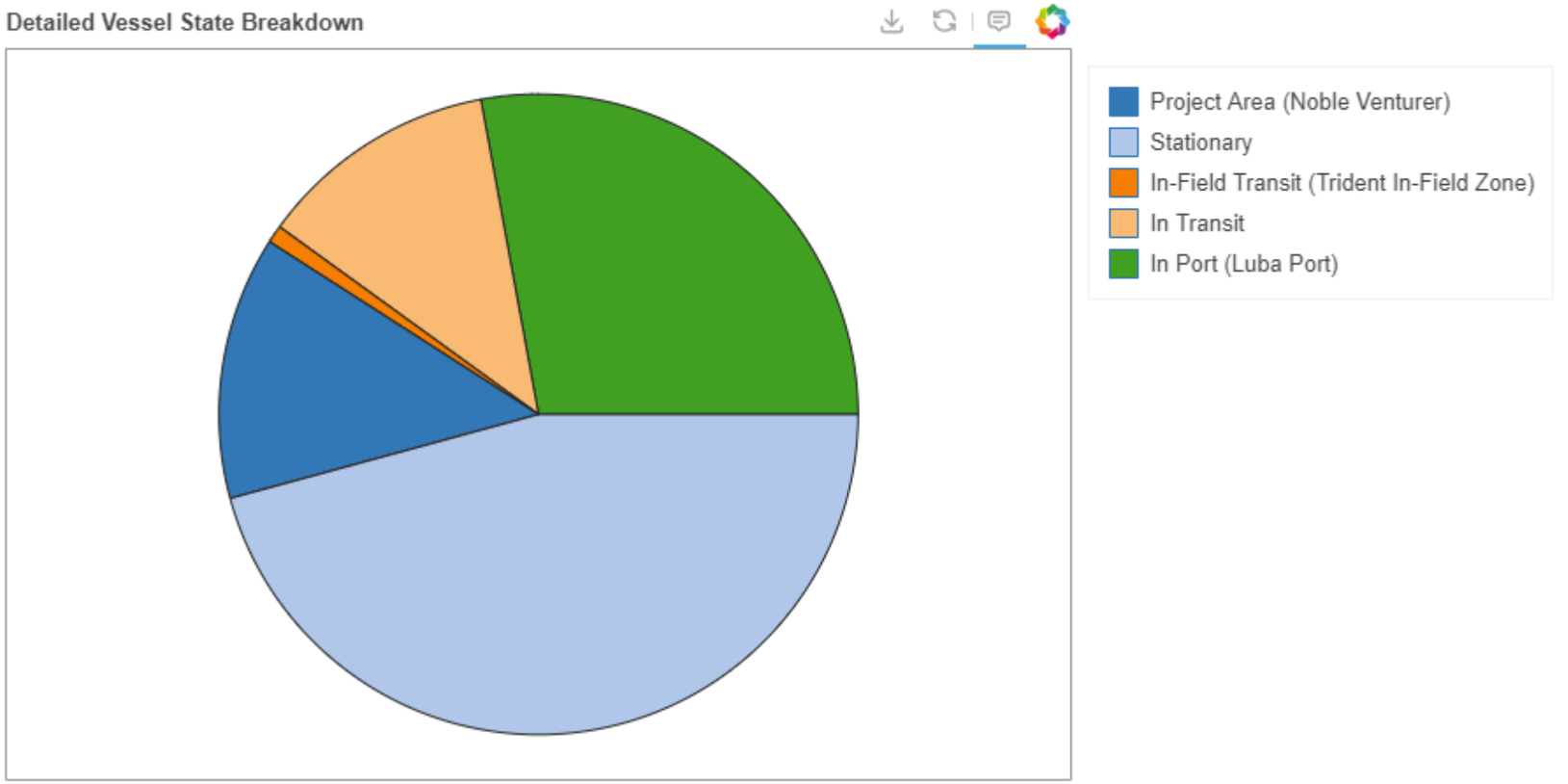
2024-10-31 to 2024-11-0

Update

Vessel State	State Duration (%)
Stationary	45.8
In Port	27.9
Project Area	13.3
In Transit	12.2
In-Field Transit	0.9






Detailed Vessel State Breakdown
Project Area (Noble Venturer)
Stationary
In-Field Transit (Trident In-Field Zone)
In Transit
In Port (Luba Port)

Select Vessel

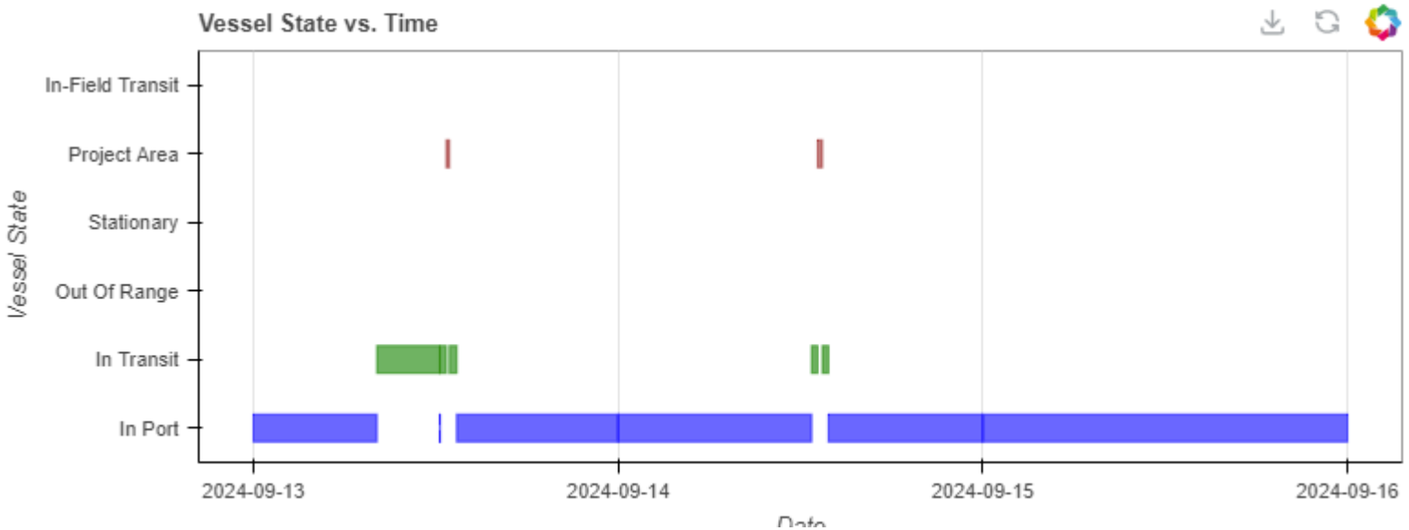
5B-DDH

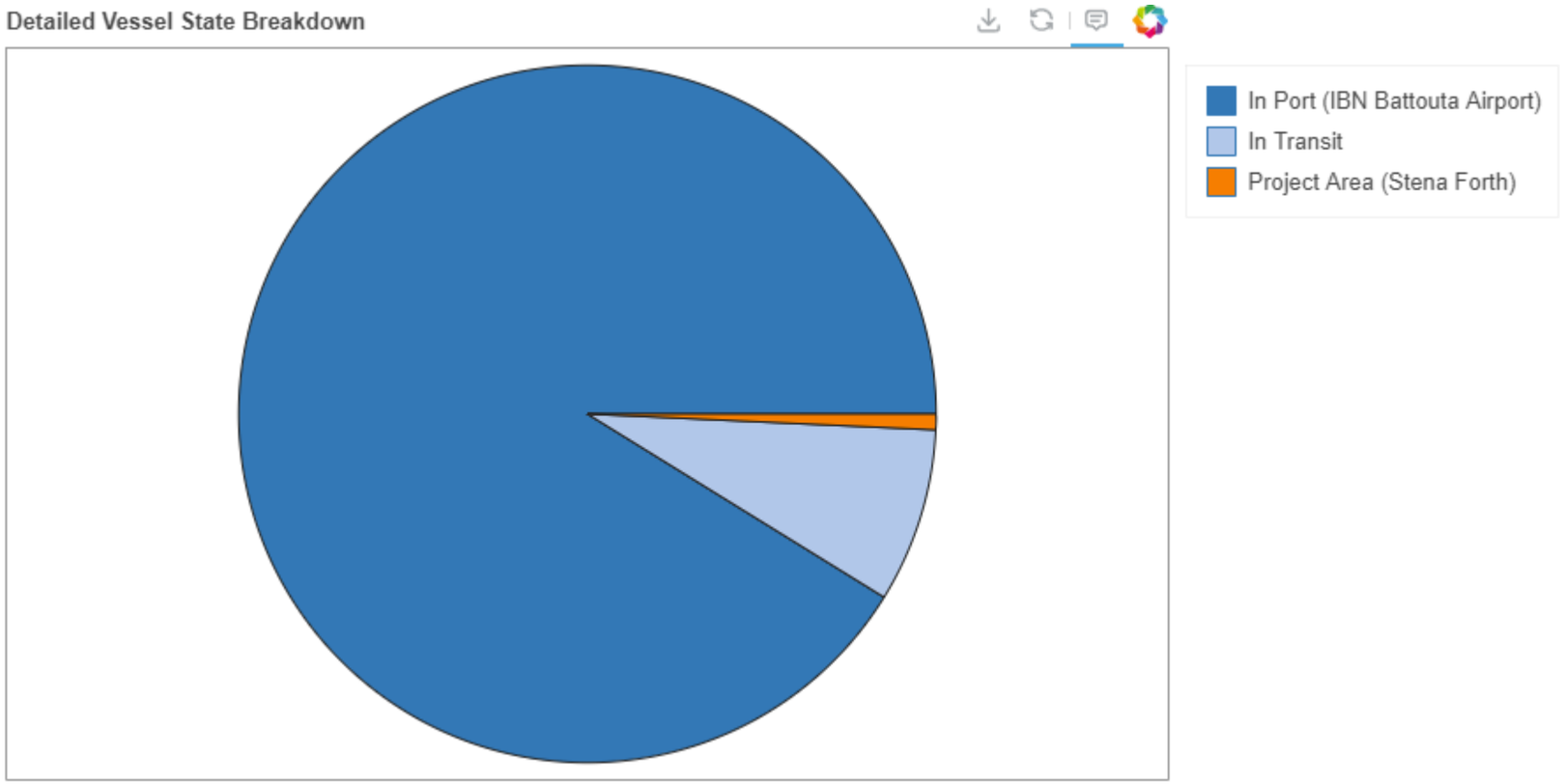
Date Range

2024-09-13 to 2024-09-1:

Update

Vessel State	State Duration (%)
In Port	91.2
In Transit	8.1
Project Area	0.7




Detailed Vessel State Breakdown
In Port (IBN Battouta Airport)
In Transit
Project Area (Stena Forth)