

Managing Historic Automatic Identification System Data by using the right Database Management System Structure

MSc Thesis by Irene de Vreede

Supervisors: C.W. Quak & P.J.M. van Oosterom

Co-reader: J.W. Konings

Delegate: M.E.A. Haffner

Content

Topic Introduction
AIS data storage Requirements
Problem description
Research questions
Methodology
Implementation
 Database Selection
 Database Organization
 Database Performance
Results
Conclusion
Recommendations future work

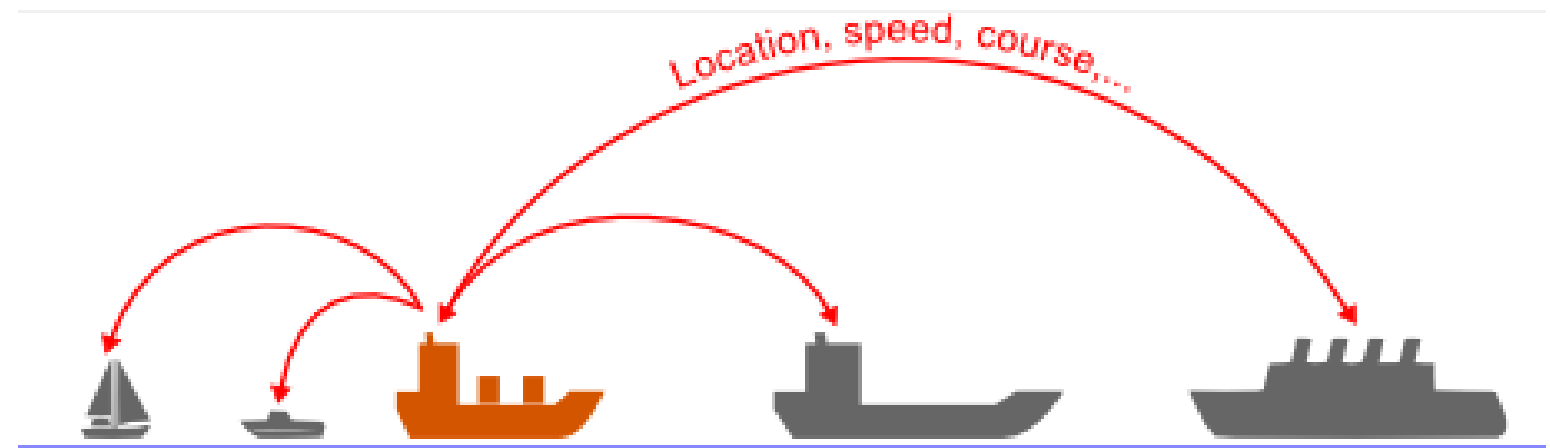
Topic Introduction

AIS data

- NMEA-0183 encrypted
- 27 different messages
- Different Update rates

AIS data usage

- Vessels
- Vessel Traffic Services



Topic Introduction

Historic AIS data possibilities

- Spatial-Temporal analyses
 - Movement Prediction
 - Anomalies detection

Historic AIS data Storage

AIS data storage Requirements

AIS Data Features

- Fast Writes
- Semi-Structured data
- Large data Volume

Spatial-Temporal Analyses

- Fast Reads
- Complex Queries

AIS data storage Requirements

Rijkswaterstaat's Spatial-Temporal Analyses (use cases)

- Location
- Trajectory
- Bounding Box

Location

Where was a certain vessel at a specific time?

Trajectory

Where has a certain vessel been at a specified time-interval?

Bounding Box

Which vessels were in a geographical area at a specified time-interval?

Problem Description

Historic AIS data shortcomings

- No data management solution for AIS data to support such spatial-temporal analyses

Research Questions

Research Question

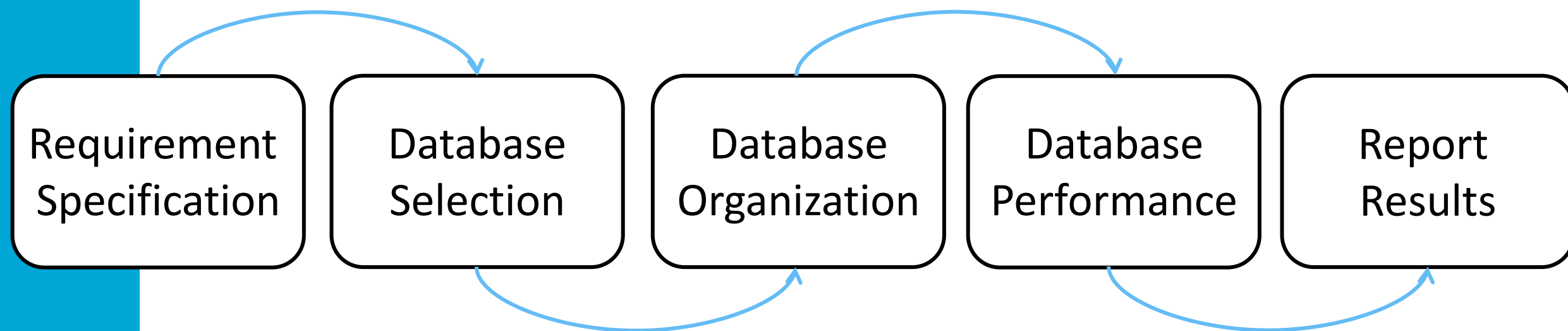
How can historic AIS data be managed, stored, and structured, to support spatial-temporal data analyses?

Sub Questions

- What is AIS data, what are its features?
- What kind of spatial-temporal analyses with historic AIS data is interesting (for Rijkswaterstaat)?
- What database should store AIS data?
- How should this database store the historic AIS data?
- What indexing technique is suitable to provide efficient historic spatial-temporal data requests?

Methodology

How can historic AIS data be managed, stored, and structured, to support spatial-temporal data analyses?



Based on [Klein et al, 2015]

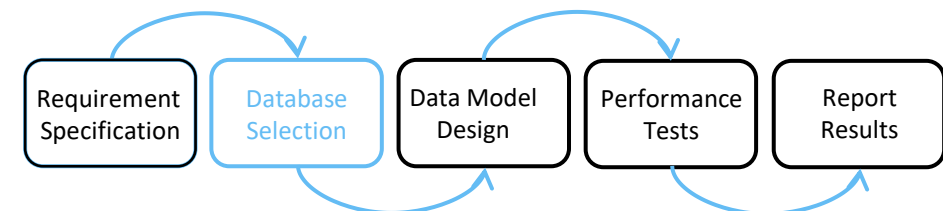
Implementation: Database Selection

MongoDB

Open Source easy to use NoSQL document based database

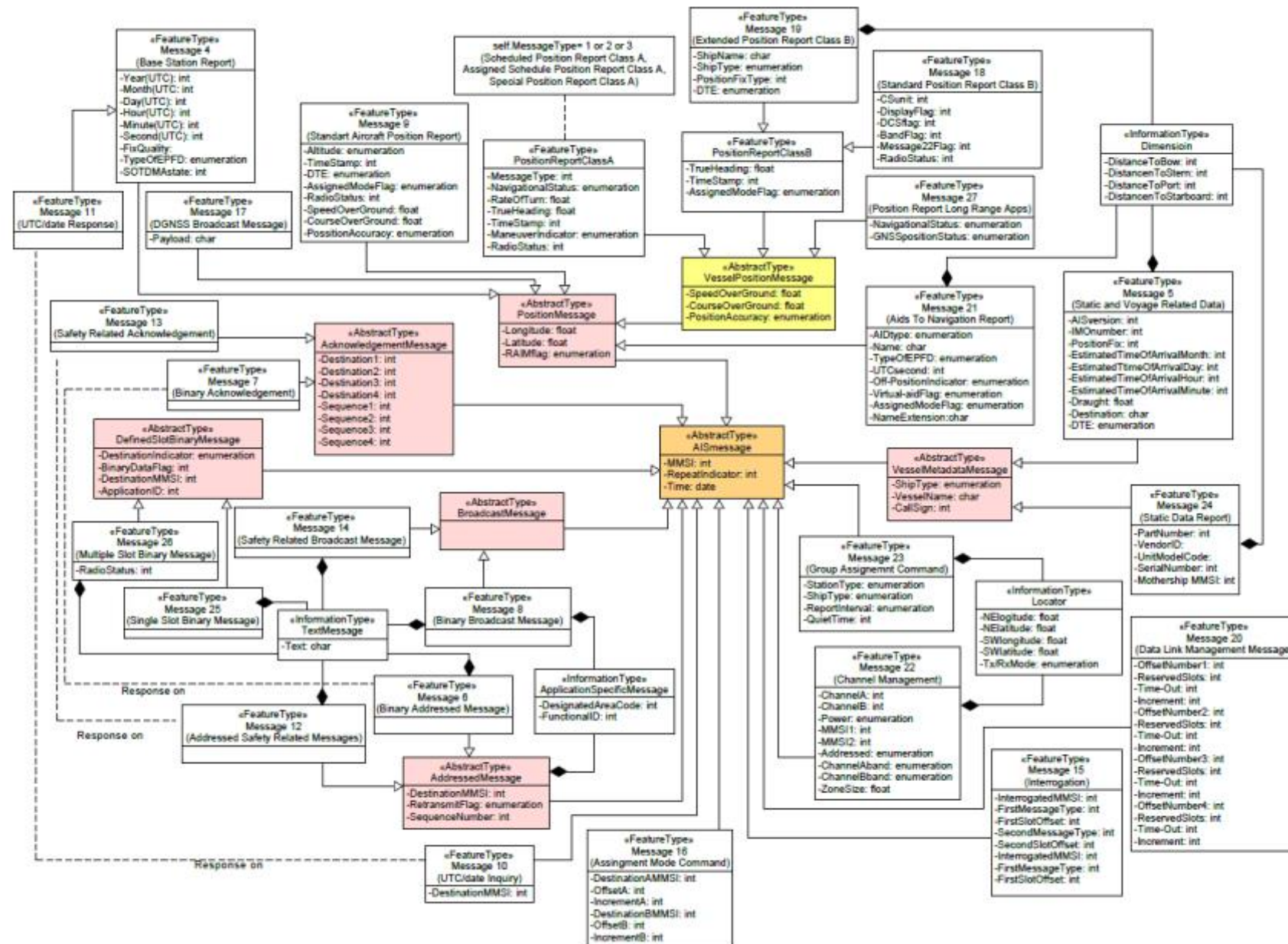
- Expressive Query Language → complex query handling
- Strong consistency → Great analysis functions
- Flexible data model → Schemeless
- Horizontal Scalability → Large data volume handling
- Easy to combine with different programming languages

Usable for complex data and performing analyses



Implementation: Database Organization

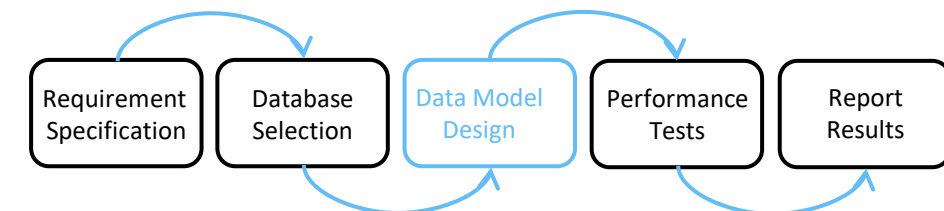
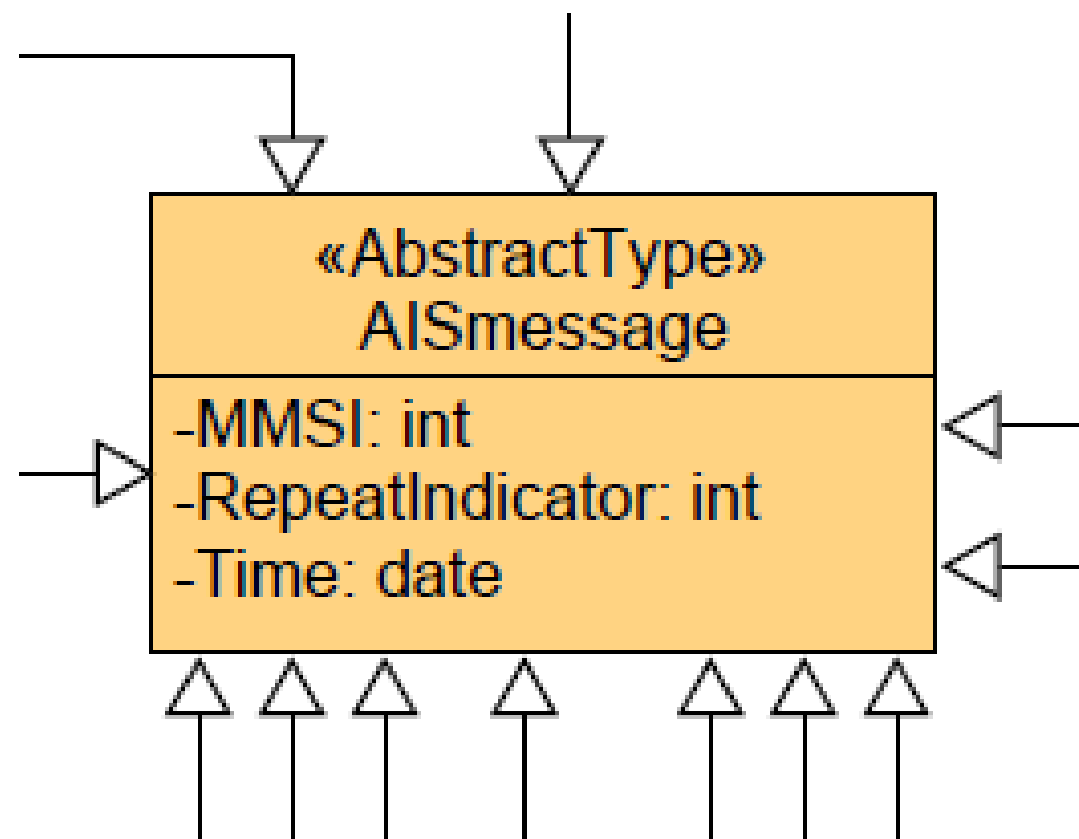
Data Model



Implementation: Database Organization

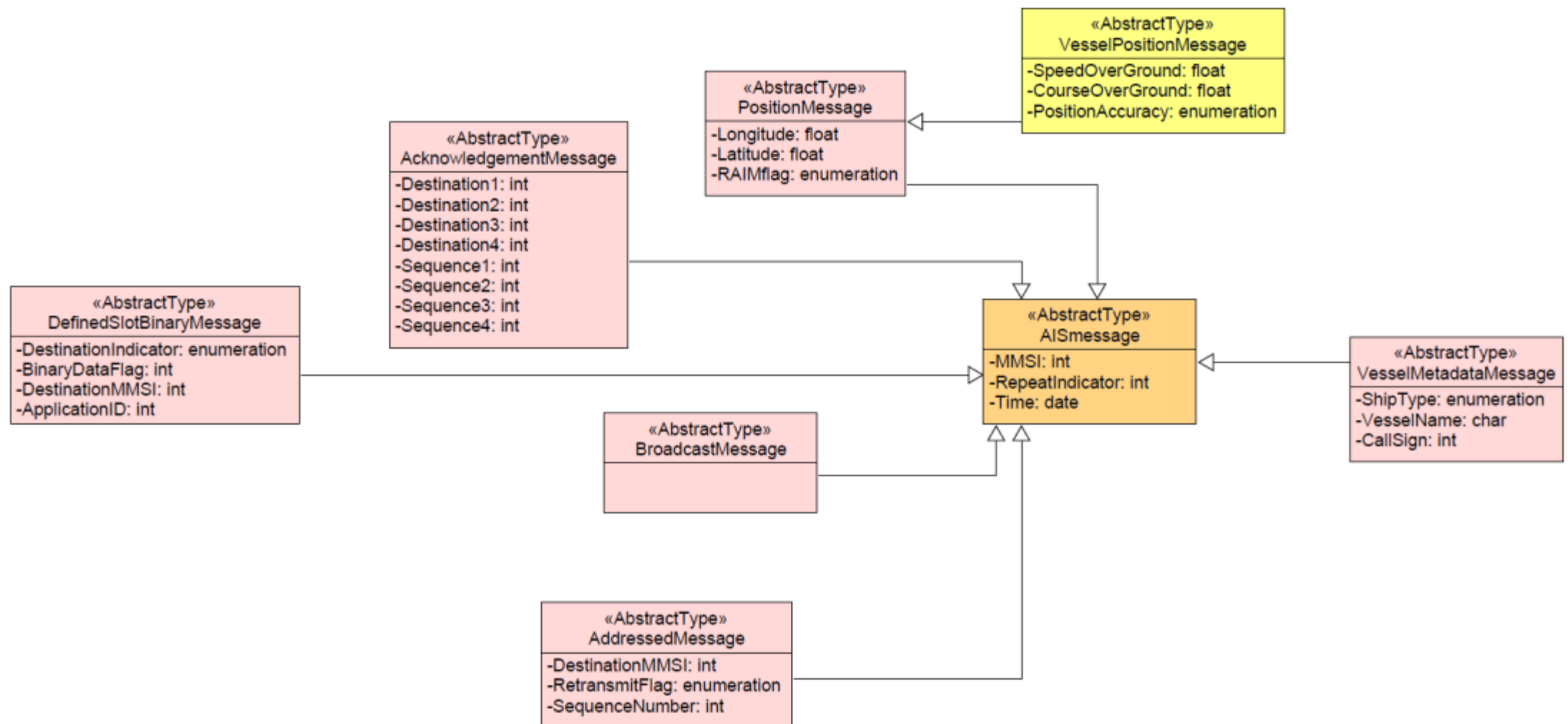
1 Central message

All messages contain these three attributes



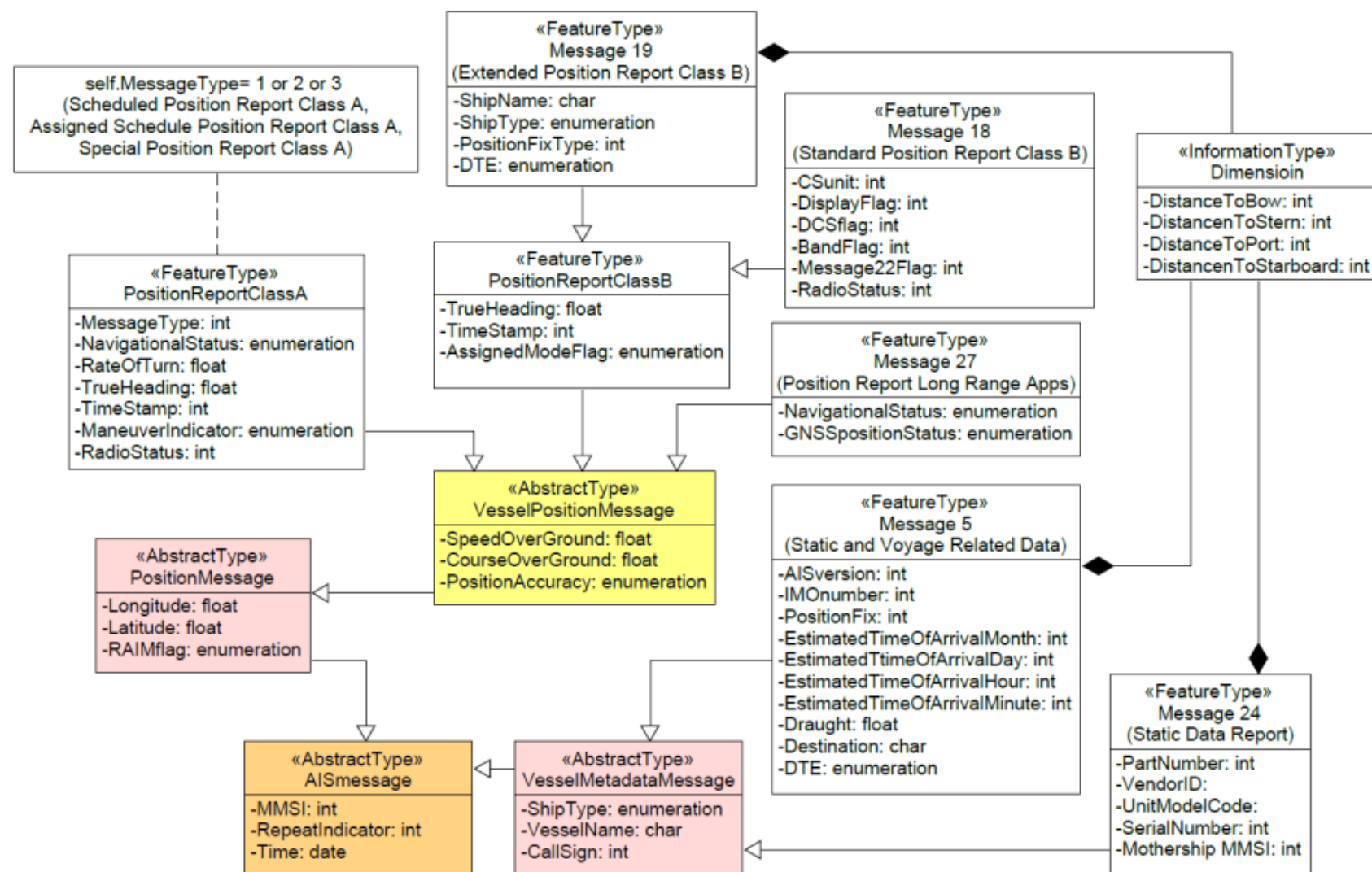
Implementation: Database Organization

6 AIS message types



Implementation: Database Organization

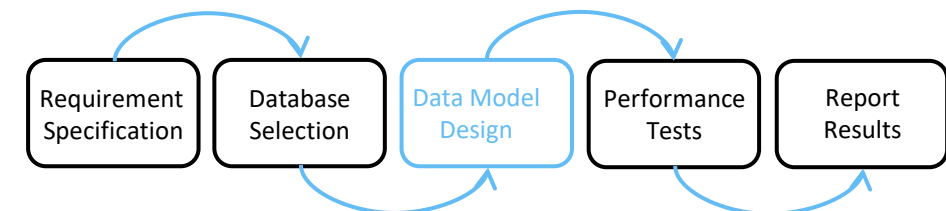
Data Model vs Use Cases → Two necessary message types
→ Eight messages → Six attributes to answer use cases



Implementation: Database Organization

Data Model vs Use Cases → Two data(base) Organizations

1. Use case focused Approach
 - MMSI, date-time, Latitude, Longitude, original AIS message
2. Spatial-Temporal Analyses focused Approach
 - All data (decoded)

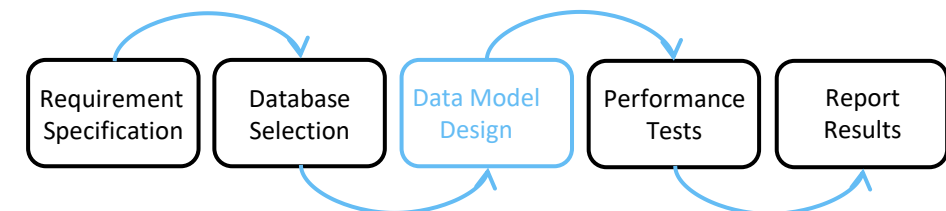


Implementation: Database Organization

MongoDB organization Based on Necessary Message Types

- Division in three database collections
 - Vessel Positions
 - Metadata
 - Other

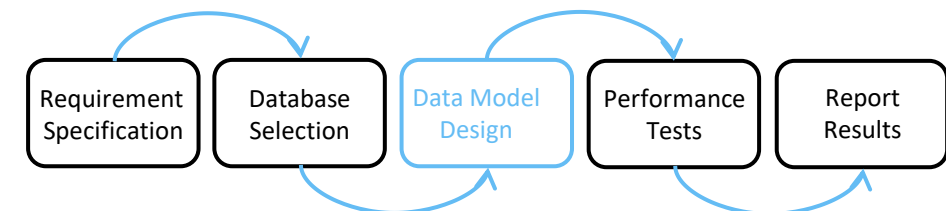
One AIS message per document



Implementation: Database Organization

MongoDB organization

- Data pre-processing
 - Decoding
 - Necessary data extraction (based on approach)
 - Format transformation → JSON
- Data loading

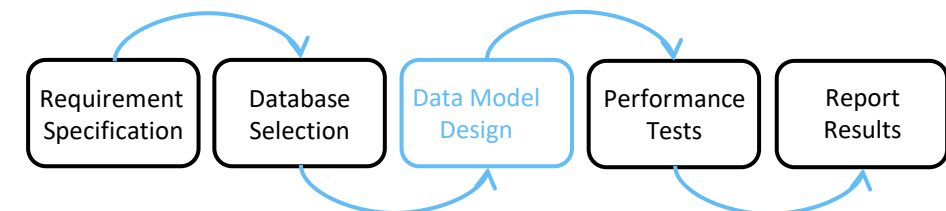


Implementation: Database Organization

MongoDB organization

```
{
  "_id" : ObjectId("57c98a6842861dd63038280c"),
  "y" : "51.871704",
  "x" : "4.312366",
  "mmsi" : "211535300",
  "message" : " \\t[ 'ABUDM,1,1,8,A,139g5i0P00PCgHpMcUCUdwnR15n,0*66\\r\\n",
  "DateTime" : "2015-11-01T00:00:00Z "
}
```

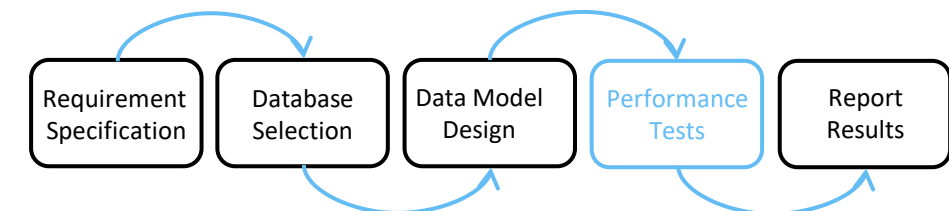
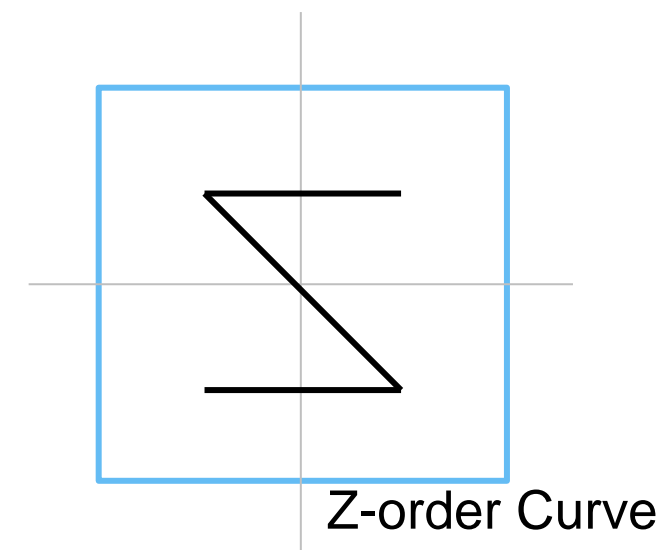
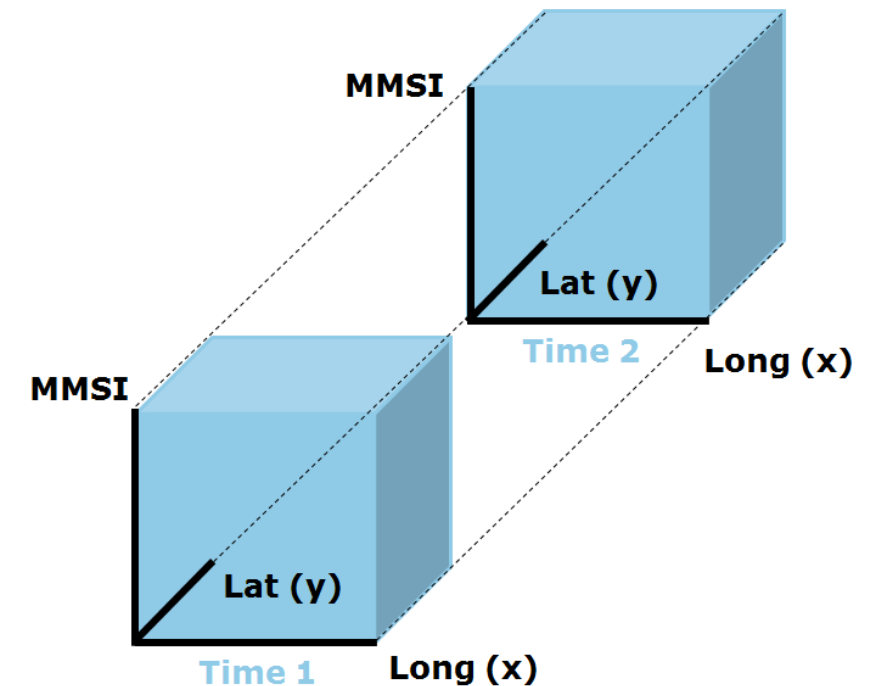
```
{
  "_id" : ObjectId("57bae5263e8595ee00c4ched"),
  "slot_timeout" : " 1L",
  "sog" : " 14.5",
  "repeat_indicator" : " 0L",
  "DateTime" : "2015-11-01T00:00:00Z ",
  "utc_min" : " 59L",
  "rot_over_range" : " False",
  "id" : " 1L",
  "sync_state" : " 0L",
  "rot" : " 0.0",
  "true_heading" : " 285L",
  "special_manoeuvre" : " 0L",
  "timestamp" : " 59L",
  "mmsi" : " 245207000L",
  "rain" : " False",
  "spare" : " 0L",
  "utc_spare" : " 0L",
  "nav_status" : " 0L",
  "utc_hour" : " 23L",
  "cog" : " 285.0",
  "y" : " 51.83150100708008",
  "x" : " 4.520533561706543",
  "position_accuracy" : " 1L"
}
```



Implementation: Database Performance

Index

- 4D Morton
 - Longitude (x)
 - Latitude (y)
 - MMSI
 - date-time
- Bit-interleaving :
 - Integrated space time
- Scaling
 - Latitude, Longitude → Meters
 - Time → Seconds

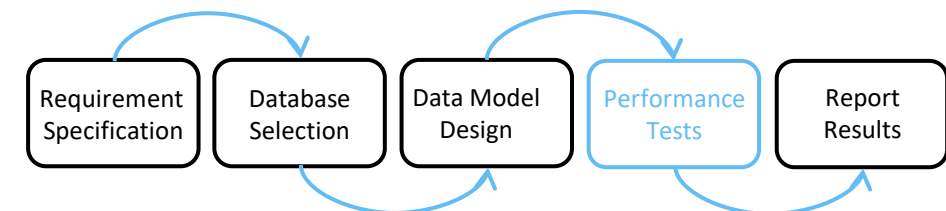


Implementation: Database Performance

Index

- Implementation
 - Morton code per Vessel Position
 - Extra Key-Value pair
 - Unique Index → `_id`
 - B-tree Index on Morton code

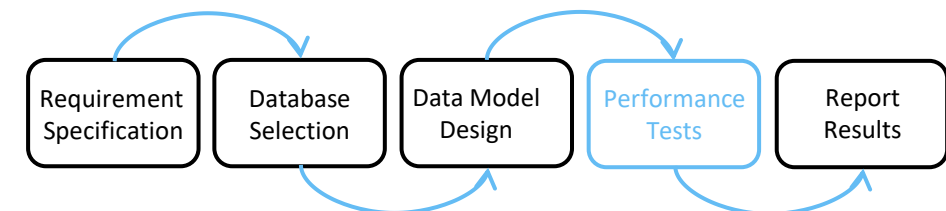
```
{  
  "_id": "10676913965458926673960704305355367441",  
  "mmsi": "245207000",  
  "DateTime": "2015-11-01T00:00:00Z ",  
  "y": "51.831501",  
  "x": "4.520533",  
  "message": " \t[!ABVDM,1,1,1,A,13an?n002APDdH0Mb85;8`sn06sd,0*76\\r\\n']\n",  
}
```



Implementation: Database Performance

Index

- B-tree index on MMSI
- B-tree index on date-time
- B-tree index on latitude
- B-tree index on longitude



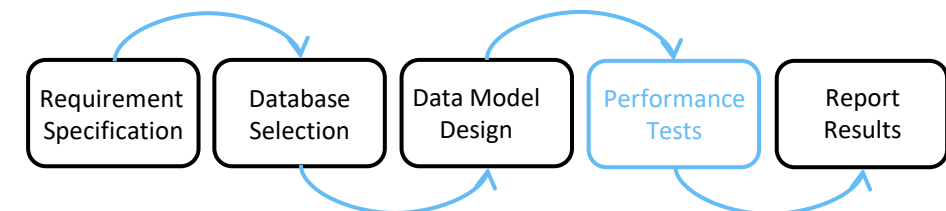
Implementation: Database Performance

Test Plan

- 2 data(base) organizations
- 5 indexes
- 3 queries (Based on the three use cases) **x5**
- 2 data volumes (day & week)

Comparing effectiveness of indexes & data(base) organizations

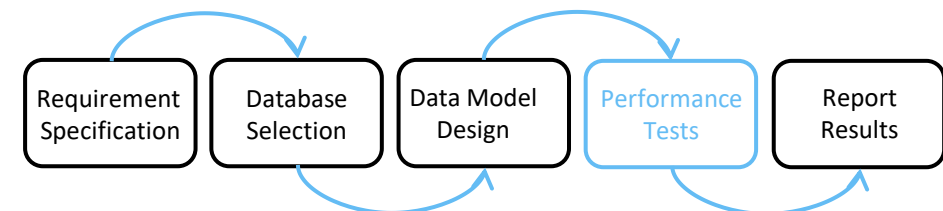
▪



Implementation: Database Performance

Effectiveness measured

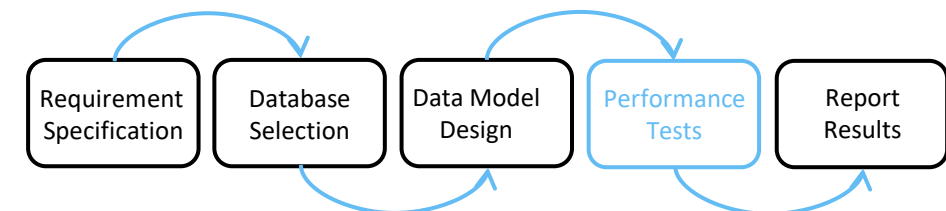
- Response Time
- Examined Documents
- Index Volume
- Database Volume



Implementation: Database Performance

Query design (Connection between python and MongoDB)

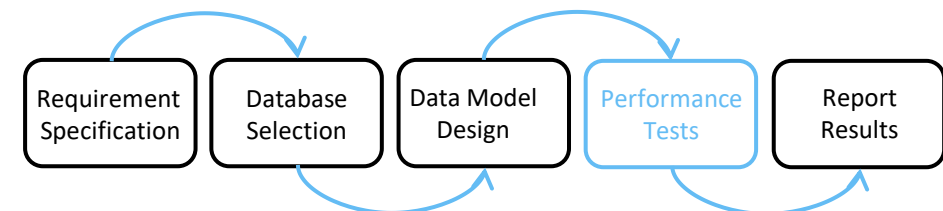
- Location
 - Last known location of vessel 244130275 at 2015-11-01/00:13:57
- Trajectory
 - All historic positions of vessel 244130275
- Bounding Box
 - All positions of vessels situated around Dordrecht at 2015-11-01/00:56:20



Implementation: Database Performance

Different Query necessary to Use index on Morton code

- Associated Morton codes are to be calculated
- These Morton codes are then queried



Implementation: Database Performance



★ location of the Vessel

0 250 500 m

Implementation: Database Performance

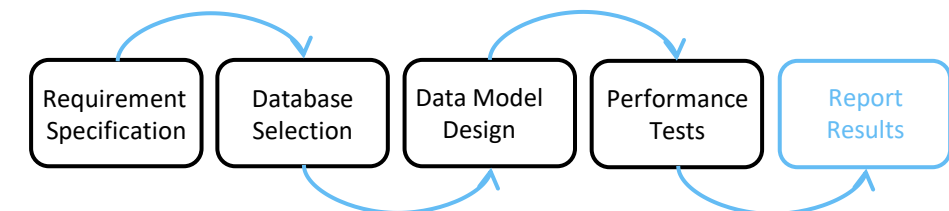
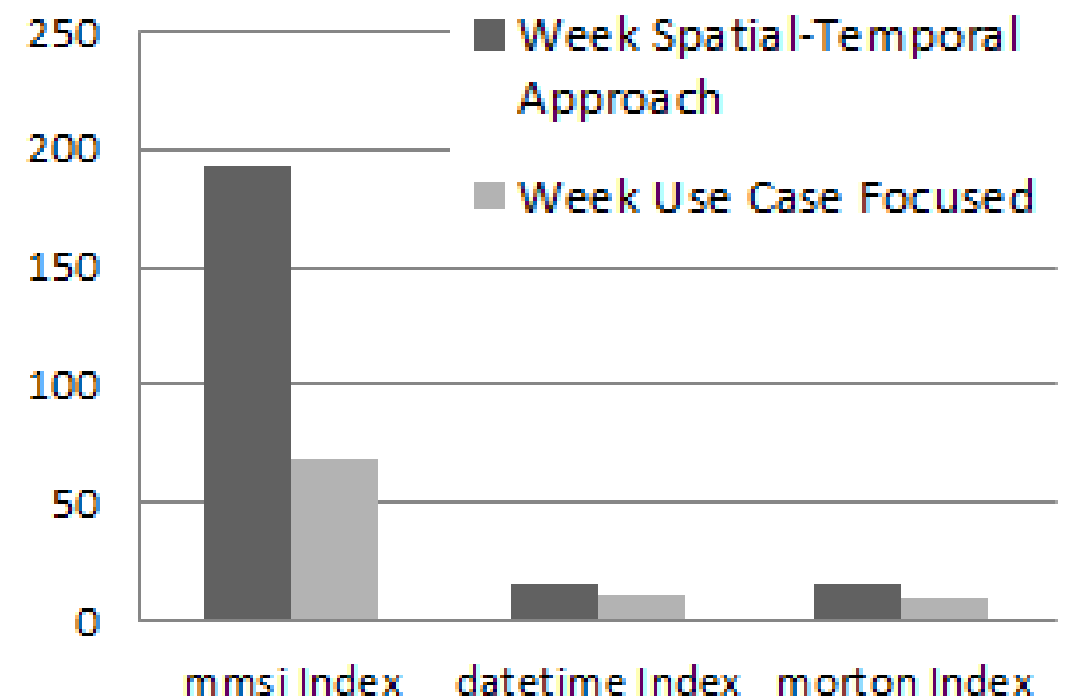
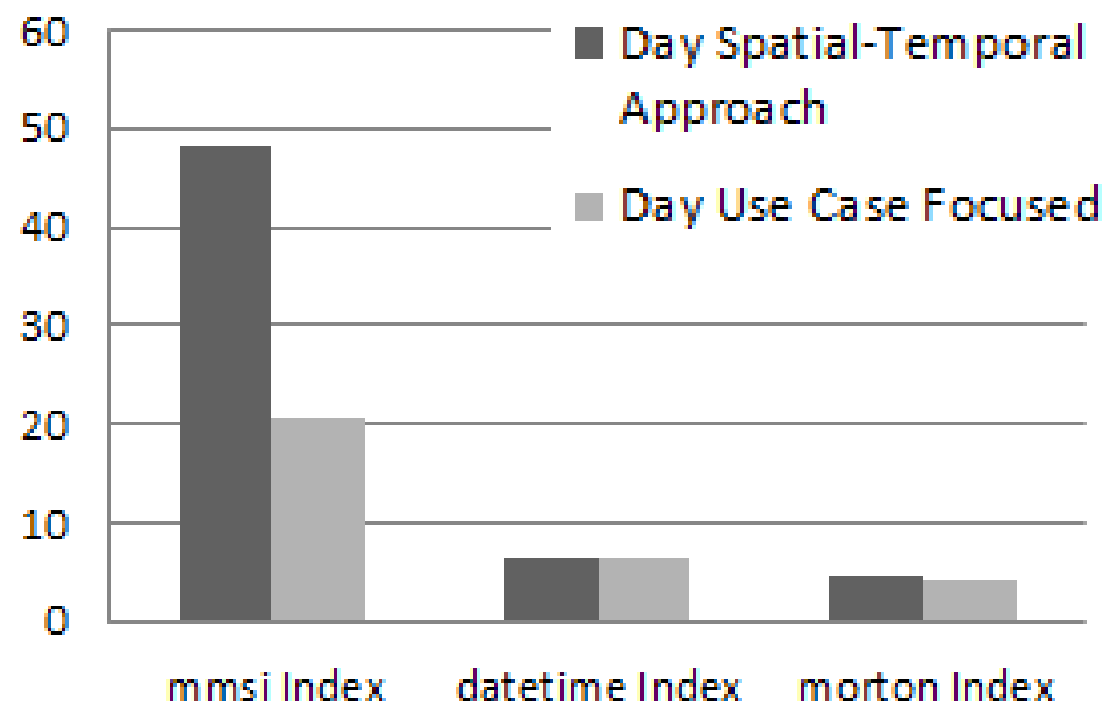


Implementation: Database Performance



Results

Indexes Compared: Response Time (seconds) Location Query



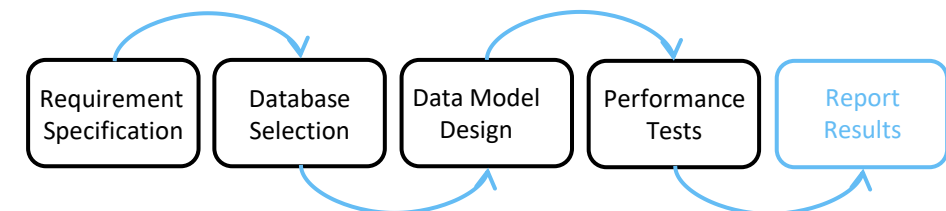
Results

Indexes Compared: Examined documents Location Query

MMSI	Date-Time	Morton
37535	328153	427

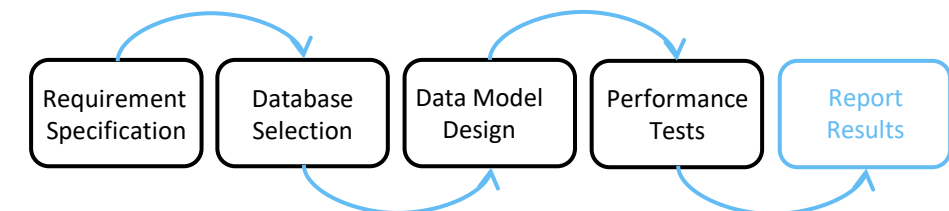
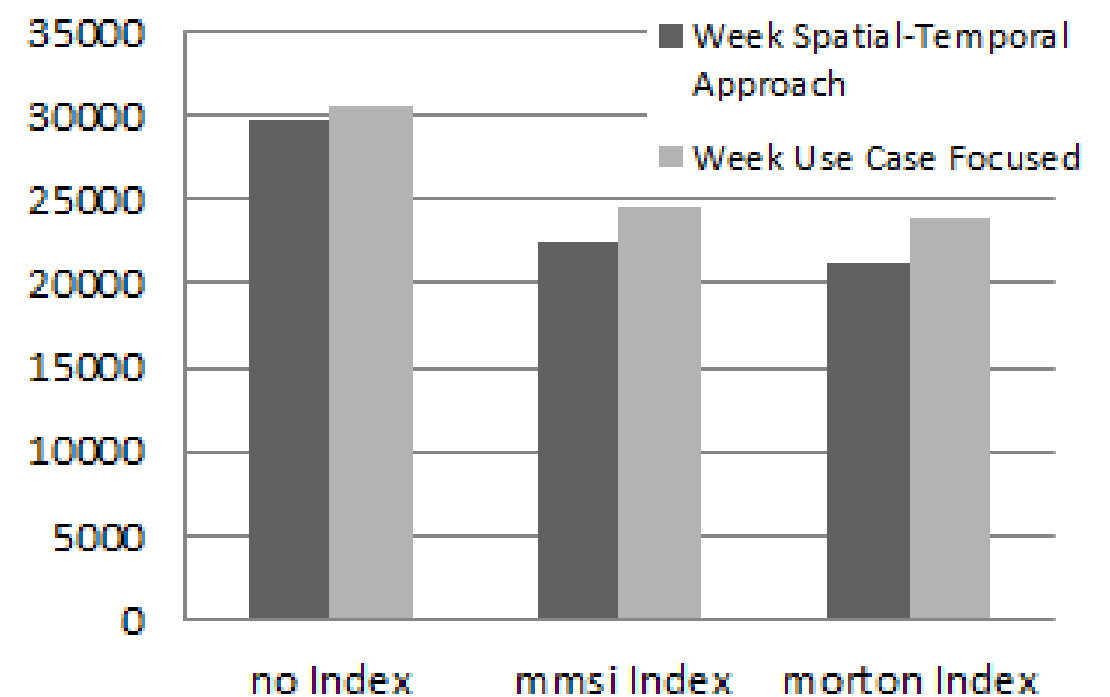
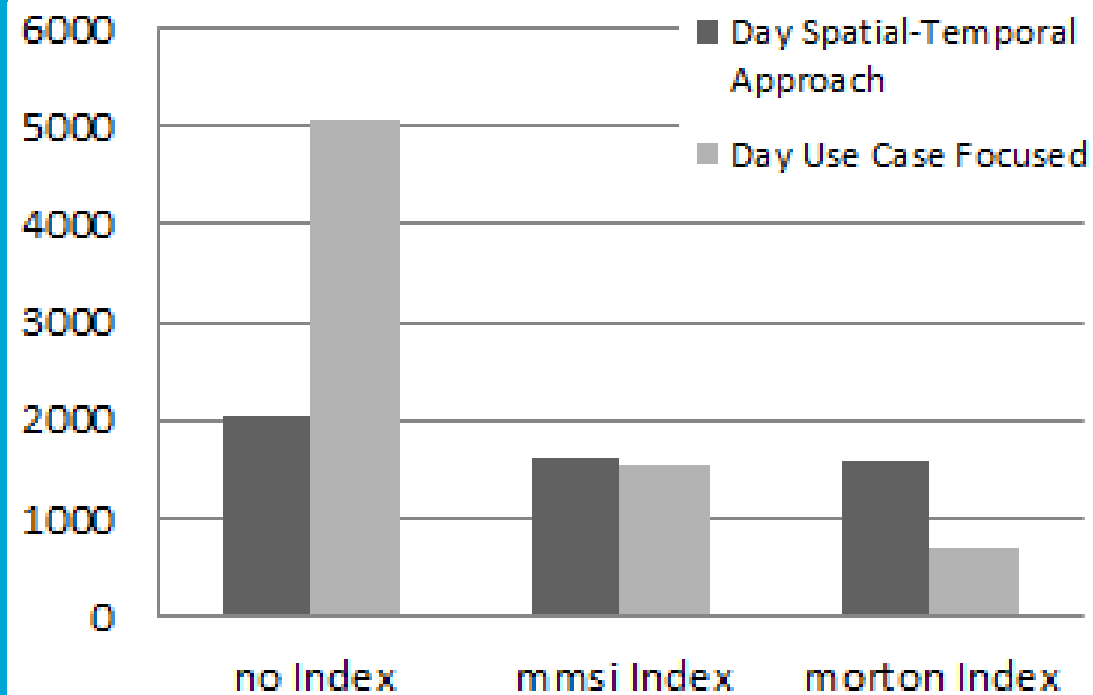
Indexes Compared: Index Volume (KB) Location Query

MMSI	Date-Time	Morton
84840448	62828544	328761344



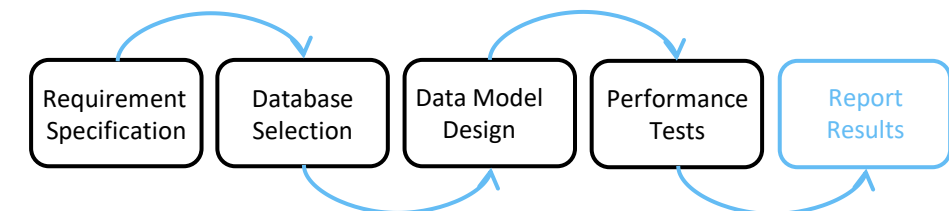
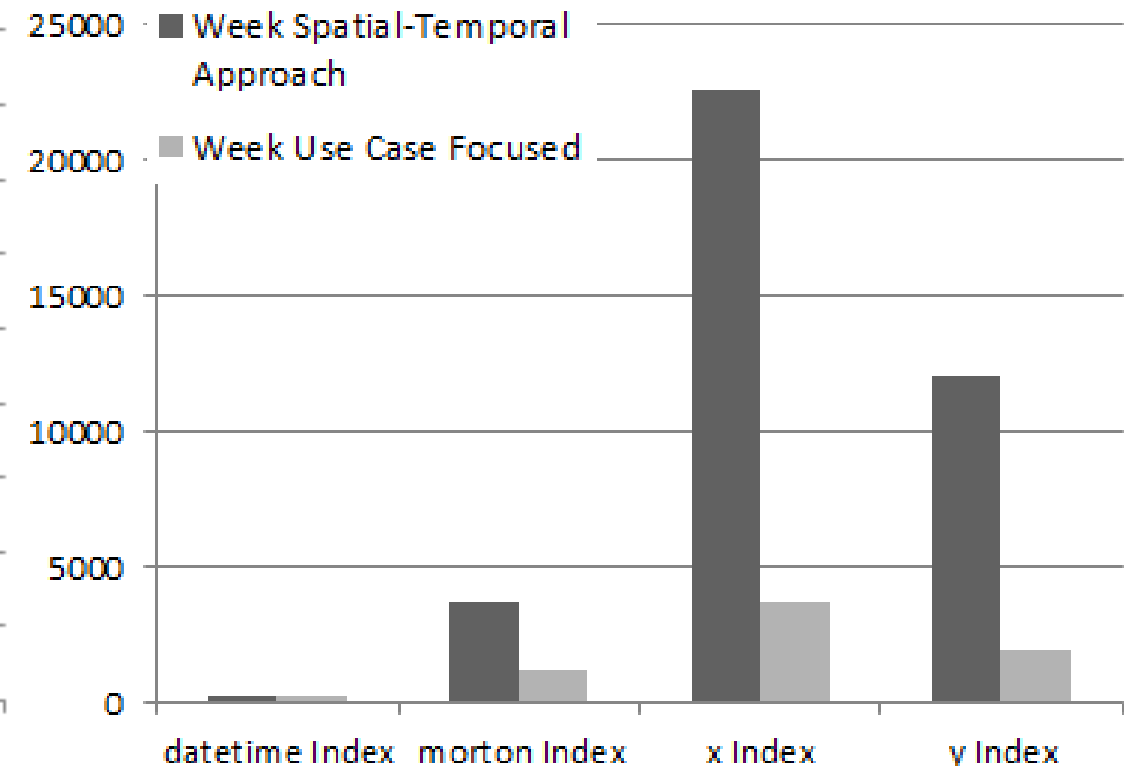
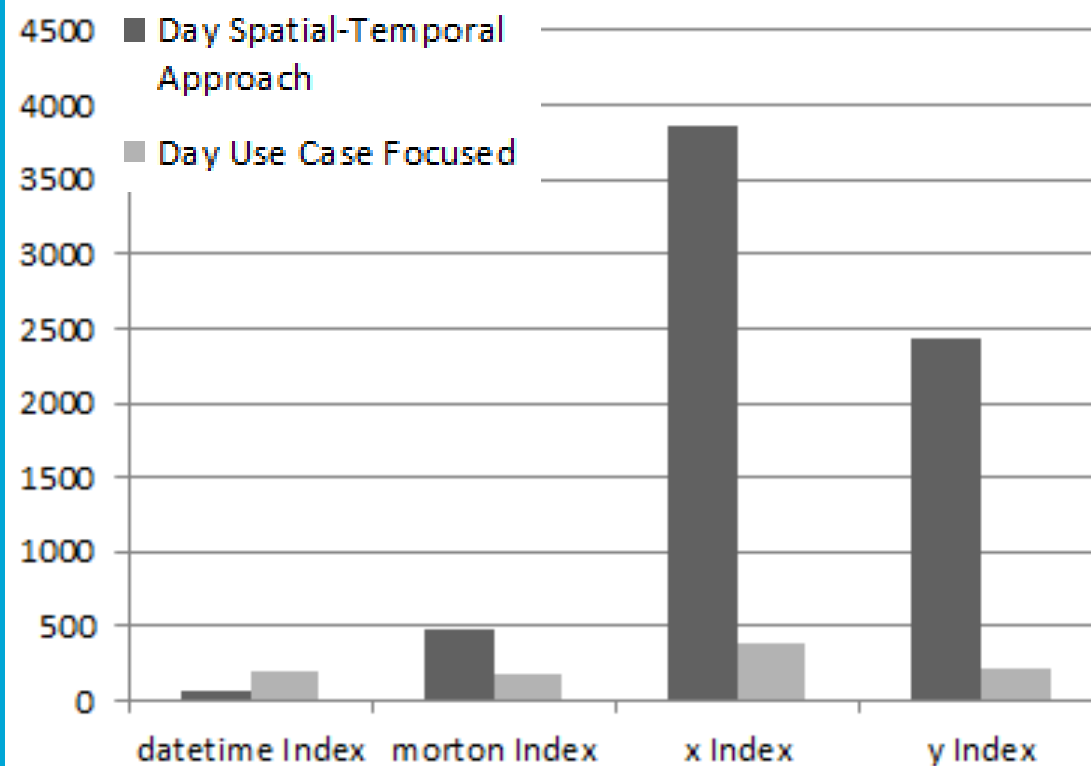
Results

Indexes Compared: Response Time (seconds) Trajectory Query



Results

Indexes Compared Response Time (seconds) B-Box Query



Results

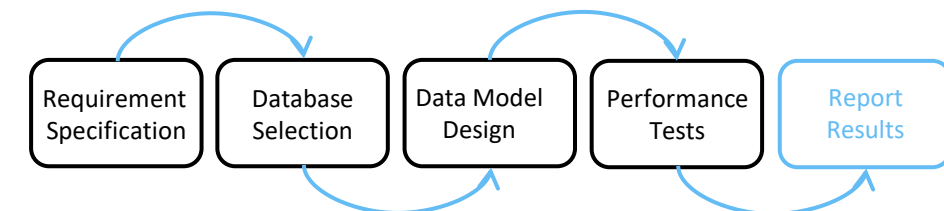
Indexes Compared: Examined documents B-Box Query

Date-Time	Morton	X	Y
3896	171	15361707	1207958

Indexes Compared: Index Volume (KB) B-Box Query

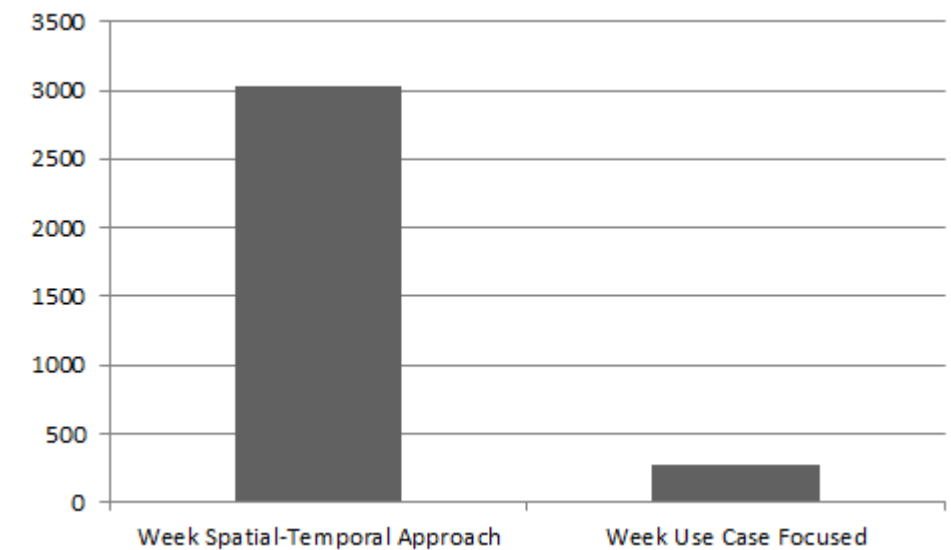
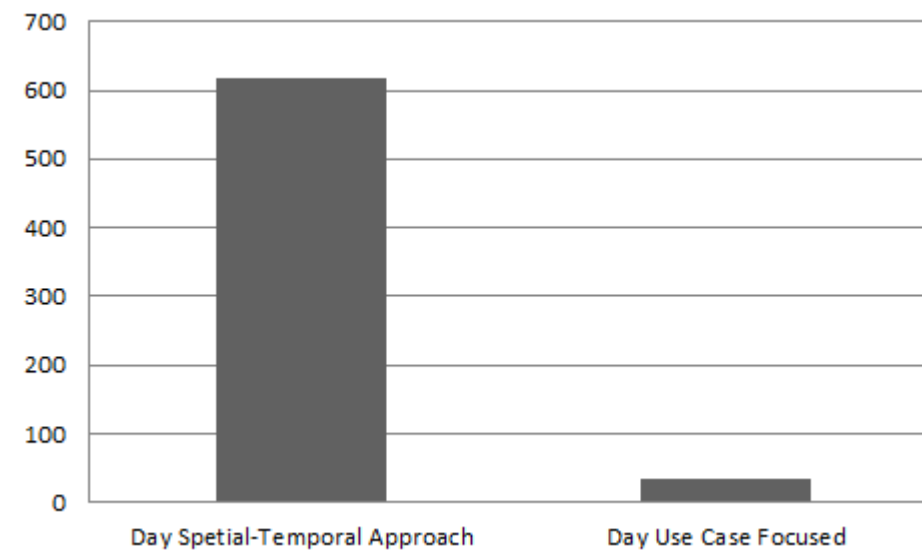
Date-Time	Morton	X	Y
62828544	328761344	112398336	97120331

Date-Time index used according to query optimization MongoDB

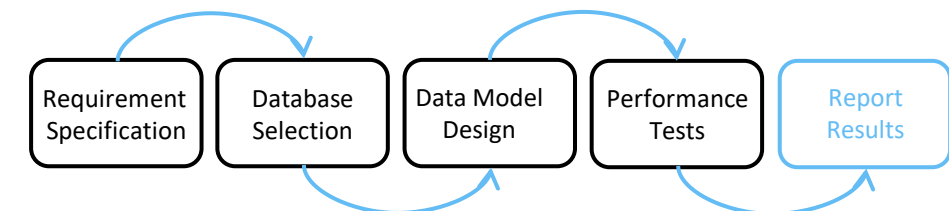
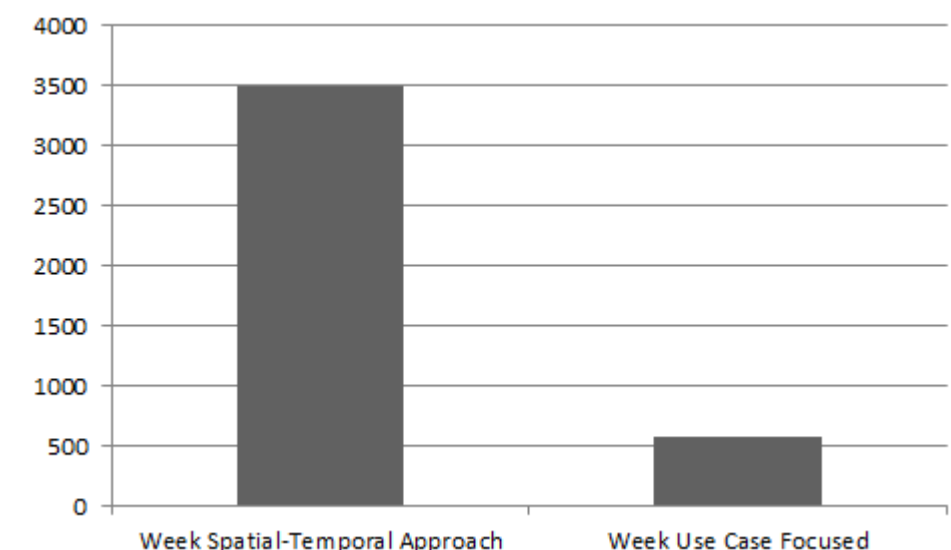
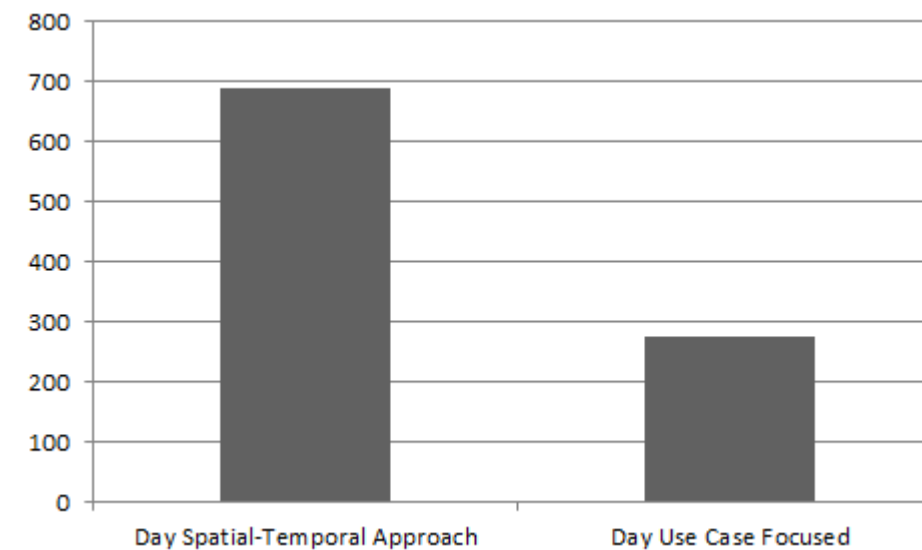


Results

Data(base) organizations Compared: Response time, Location



Data(base) organizations Compared: Response time, B-box



Results

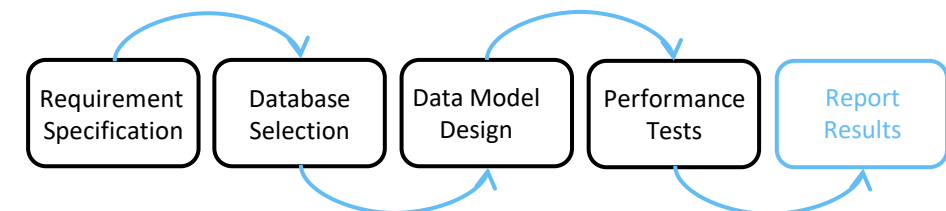
Data(base) organizations Compared: Database Volume

Day Spatial-Temporal Approach 15.311 GB

Day Use Case Focused 1.654 GB

Week Spatial-Temporal Approach 83.640 GB

Week Use Case Focused 12.195 GB

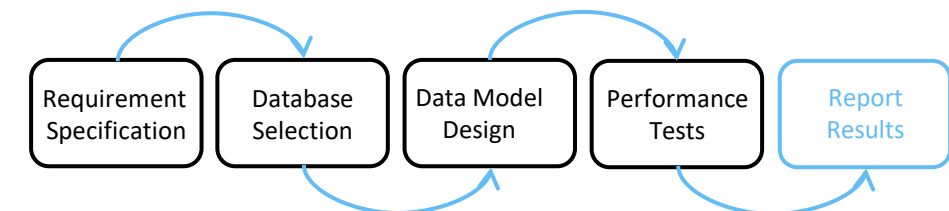


Results

Data(base) organizations Compared: Database Volume

```
{  
  "mmsi": "245207000",  
  "DateTime": "2015-11-01T00:00:00Z ",  
  "y": "51.831501",  
  "x": "4.520533",  
  "message": " \t[!ABVDM,1,1,1,A,13an?n002APDdH0Mb85;8`sn06sd,0*76\\r\\n']\n",  
}
```

```
{  
  'slot_timeout': 1L,  
  'sync_state': 0L,  
  'true_heading': 285L,  
  'utc_spare': 0L,  
  'sog': 14.5,  
  'rot': 0.0,  
  'nav_status': 0L,  
  'repeat_indicator': 0L,  
  'raim': False,  
  'id': 1L,  
  'utc_min': 59L,  
  'spare': 0L,  
  'cog': 285.0,  
  'timestamp': 59L,  
  'y': 51.83150100708008,  
  'u'x': 4.520533561706543,  
  'position_accuracy': 1L,  
  'utc_hour': 23L,  
  'rot_over_range': False,  
  'mmsi': 245207000L,  
  'special_manoeuvre': 0L  
}
```



Conclusion

- **How can historic AIS data be managed, stored, and structured, to support spatial-temporal data analyses?**
- AIS data
- Spatial-temporal analyses
- Database
- Database organization
- Indexing technique

Recommendations Future Work

- Another database by benchmark
Theoretically chosen database does not have to be the best choice in reality
- Using an Space Filling Curve cluster with reserved space on the hyper cube to ensure space for data updates

Thank you for your attention!