RTL-SDRで作るAIS受信機とAIS情報表示サイト

OSC2023北海道
2023年6月17日
有限会社サンビットシステム
佐々木伸幸

Sunbit System

那覇クルーズターミナル
2017/6/14

1964年生まれ　　twitter @n_sasaki　facebook:本名

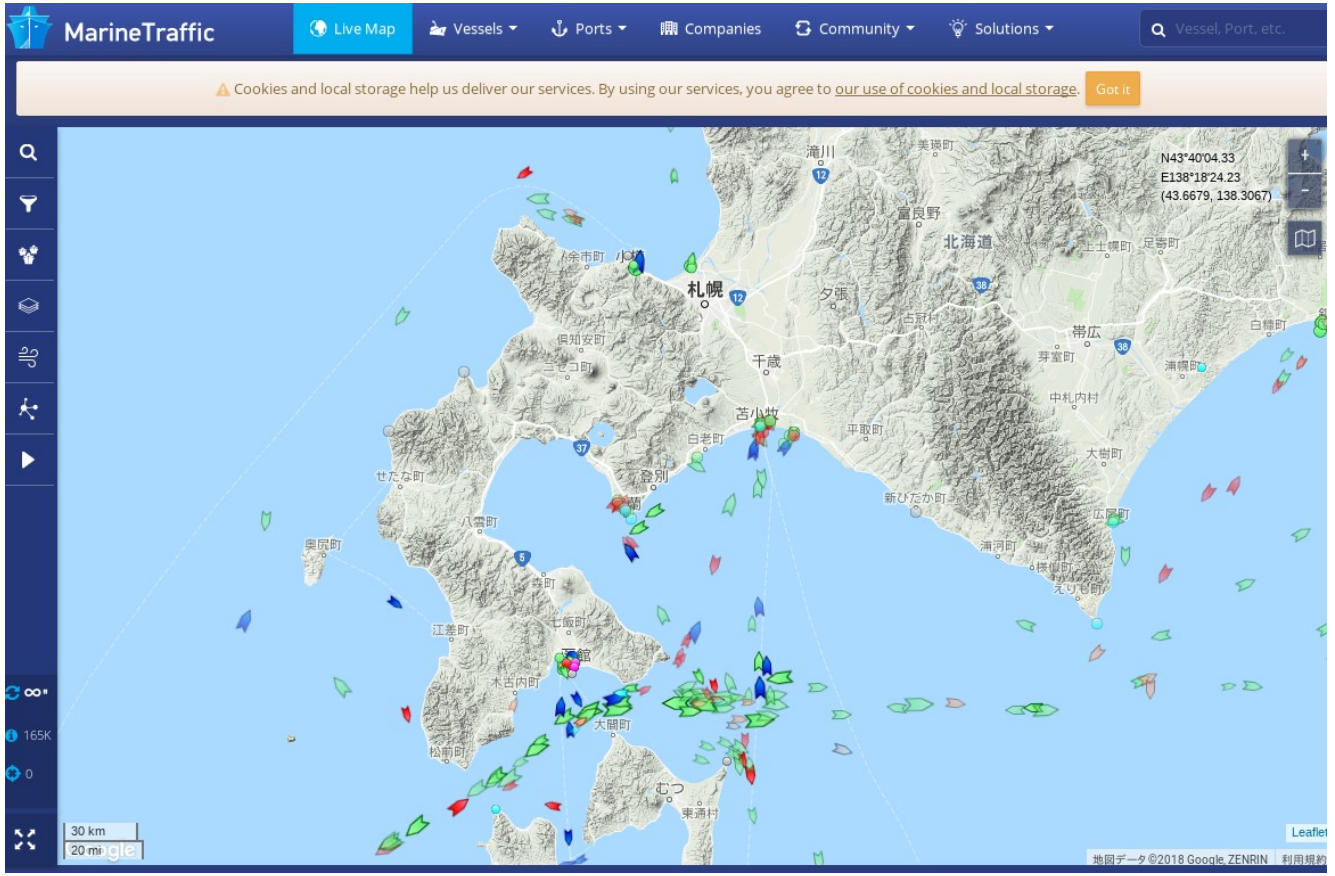
- 1988年からこの業界
- 1990年くらいからUNIXいぢってます(Xenix,SysV,SunOS,BSD)
- 1998年独立して現職。OSS関連中心にいろいろやってます
  OSC北海道には最初から関わってます。

Sunbit System

みたことありますよね？

Sunbit System

みたことありますよね？



Sunbit System

これはSDRで電波を受信して
船舶位置表示サイトを作る話です

Sunbit System

山口県下関港
2015/10/02

AIS=船舶位置自動識別装置

船の位置をGPSで取得し位置情報をVHF帯で受発信する装置

- 30-40Kmは到達
- 自船周辺の他船舶の位置把握と衝突回避が目的
- 電波を発信するには海上特殊無線技師免許が必要
- 受信して位置表示するだけの簡易型もある

2002年以降

- 国際航海に従事する300t以上の客船
- 500t以上のすべての船舶

に搭載義務がある

SDR: Software Defined Radio

チューナーで受信した信号をA/D変換し、復調から先の工程をソフトウエアで行う無線受信機。

変調までソフトウエアで行いD/A変換したものを電波として送信することも可能。
（ちゃんと送信を行うには資格や設備が必要）

RTL-SDR

RTL2382Uを使ったUSBワンセグチューナが生I/Qデータをを USBに流せることを利用して、復調をソフトウエアで行うソフトウエアラジオ。
同調用チップによって受信周波数範囲は変わるが、
500Khz – 1.7GHzと超広帯域受信機になる。

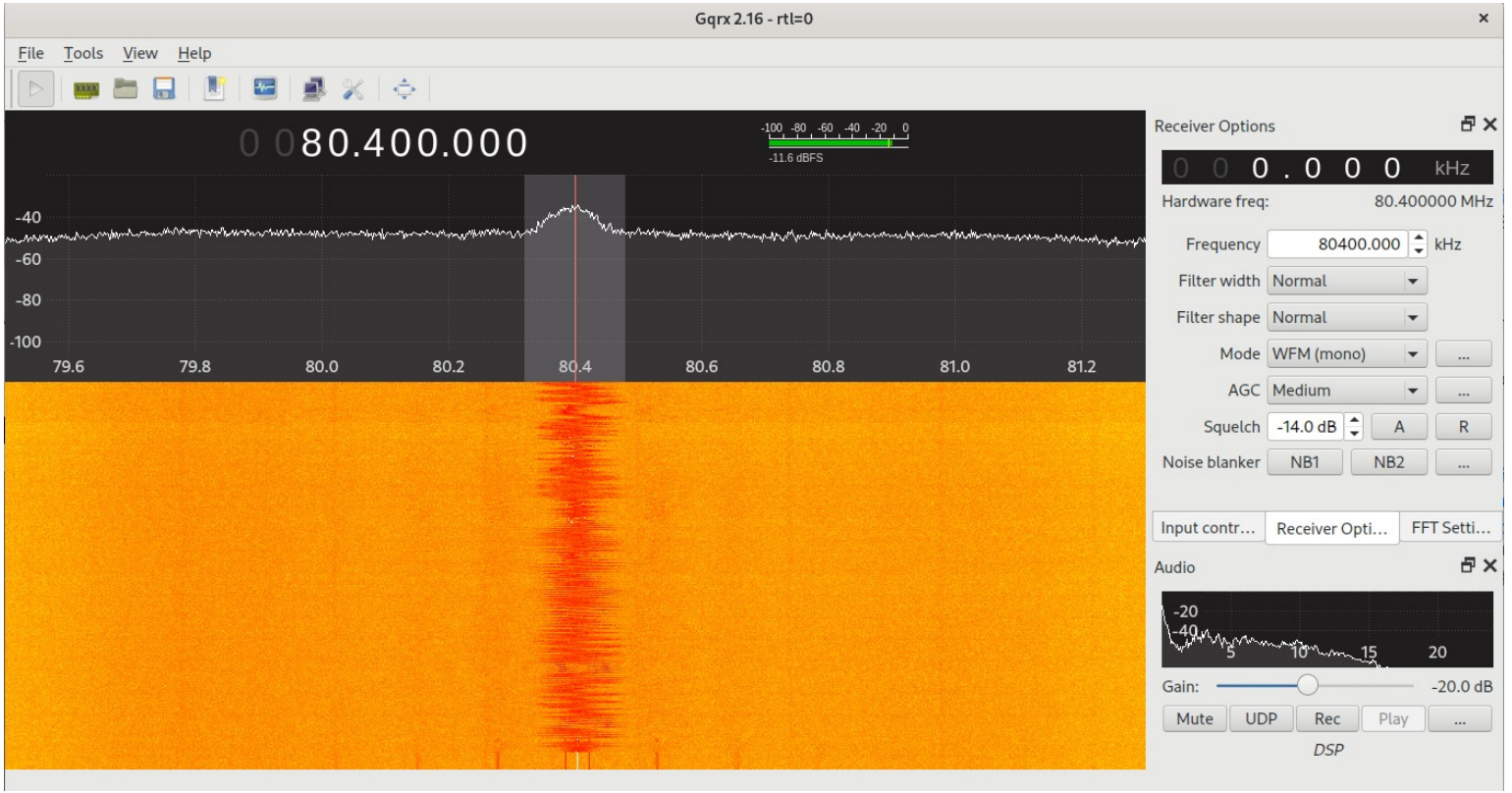
BSD,Linux, Windows, MacOSなどで動作。

情報の取得元
http://osmocom.org/projects/sdr/wiki/rtl-sdr
https://www.rtl-sdr.com/about-rtl-sdr/

RTL-SDRのUSBドングル

http://livedoor.blogimg.jp/bh5ea20tb/imgs/b/9/b9a45cf4.jpg

SDRを使ったアプリケーション(Gqrx)



SDRのより詳しい解説

今年のSECCON2022電脳会議で江草さんが発表した資料がわかりやすいです。

https://speakerdeck.com/chibiegg/wireless-analyzing-using-sdr-at-seccon-2022

AISで聞こえてる情報

type 1,3　　クラスA位置情報(搭載義務船舶用)
type 18,19　クラスB位置情報(簡易型：漁船等)
　船舶や基地局のGPS的な現在位置情報

type 4　　基地局情報
　基地局の名前など

type 5　　船舶情報
　船舶の名前、大きさなど

その他相互の問い合わせ、航路標識情報などがある

AISで聞こえてる情報

位置情報

MMSI:	AISで使用される個体識別番号
Lon,Lat:	GPSの緯度経度
Time:	発信時刻
Heading:	船首の向き(360度)
Speed:	速度(ノット)

船舶情報

MMSI:	AISで使用される個体識別番号
Callsign	無線局的なコールサイン（なくてもよい）
Name:	船舶名（なくてもよい）
ToBow	GPS位置から船首までの長さ
ToStern	GPS位置から船尾までの長さ
ToPort	GPS位置から左舷の長さ
ToStarBoard	GPS位置から右舷の長さ

まめちしき　port, starboard

starboard (side)
古ノルド語(バイキングの言葉)のstýri(操作する)とborð(舷)
steering board がなまって star board
船の操舵は舵櫂(舵取り用の櫂)によって行われていた。舵櫂は船尾にいる漕ぎ手によって操作されるが、左利きよりも右利きの人の方が多いため、右利きの漕ぎ手が操作しやすいように舵櫂は**右舷**に設置された。

port (side)
接岸時に舵櫂の邪魔にならないように**左舷**に桟橋や岸壁を着けたことに由来する。そのような船の乗降はportで行っていたが、今は左右対象のため乗降はどちらでもよい。

飛行機も船に習い starboard side, port sideというが、船より乗降口の慣例が強く、ほとんどの旅客機で使用される乗降口は port side。

AISを受信してみる

RTL-AIS

RTL-SDRを元にAIS信号のデコードまでを行う。
161.075Mhz, 162.025Mhzの2波を信号処理し、
受信したデータをNMEA形式で出力、UDPで送信できる。

NMEA形式の文字列
!AIVDM,2,2,6,B,Bp<Up8888888880,2*32
!AIVDM,1,1,,B,?0474GQa=PEPD003000,2*57
!AIVDM,1,1,,B,16Kdpj@02>:8AitHHbP86nQF0000,0*70

データフォーマットの解説
http://catb.org/gpsd/AIVDM.html

gpsd

gpsの情報を解析、中継するソフトウエア
AISのNMEAデータ(AVIDM)をフィールドデータに分解する機能を持っている
http://www.catb.org/gpsd/

gpsdには蓄積機能がない
＝受信したデータを後から利用できない。

どうやって貯めようか...

AISデータの格納方法を考える

TYPE 1,3　位置情報　　数秒 - 十数秒に一度(停泊等は3分)

TYPE 4　　基地局　　6分に一度
TYPE 5　　船舶情報　　６分に一度

type1,3 位置情報は蓄積しないと航跡をたどる等ができない
　位置情報だけでは船舶の詳しい情報はわからない

type4,5 基地局・船舶情報は最も新しいものがあればよい

位置情報で送られてくるのは地理座標系のWGS84を60000倍した数値

PostgreSQL+PostGISに格納する

AIS情報の緯度経度をPostGISのgeom形式で格納
PostGISには測地系に関する演算機能もある

geom形式
位置情報を点・線・面等に分類し、EPSGコードと共に位置情報を保持
(EPSGコードは世界中の空間座標系を示すコード)
WGS84はEPSG:4326に該当

gpsd2pgsqlを自力で書いてPostGISに貯める

gpsd2pgsql

use GPSD3ライブラリ;

gpsdにjsonモードでつなぐ
DBあける
LOOP(終了まで(SIGTERMとか))
　　jsonデータを解析してタイプから動作を判断
　　case 位置情報:
　　 INSERT MMSI,位置,格納時刻(位置は生値とgeometryで格納)
　　case 船舶or基地局:
　　 UPDATE MMSI,付属情報 (UPSERT的に)
　　default: とりあえず無視
END LOOP
(私はperlで書きましたが好きなので書けばよいと思います)

DBテーブル

```
position (
      char(10) mmsi;       # AIS識別子
      numeric lon,lat;          # 緯度経度
      geometry geom;            # 緯度経度を空間座標系にしたもの
      timestamp  createat  # 格納時刻
      # その他速度、進行方向、動静状態なども定義);

vessel (
      char(10) mmsi;       # AIS識別子
      text name;           # 船舶名
      int type;               # 貨物船、客船などの種別
      int to_bow;              # GPS位置から船首までの長さ(m)
      int to_stern;           # GPS位置から船尾までの長さ
      int to_port;             # GPS位置から左舷までの長さ
      int to_starboard;       # GPS位置から右舷までの長さ);
```

電波を受けてからDBまで

HAM用
144MHz 1/4λアンテナ

USBドングル

RTL-SDR

RTL-AIS

gpsd3

gpsd2pgsql.pl

postgreSQL

gpsd3までは
Rasberry Pi2
でも大丈夫

DB側はそれなり
に性能必要

Sunbit System

データ蓄積と表示

せっかく蓄積した情報は地図上で視覚的に見たい。

位置情報には船舶の位置・方向・速度はあるが、
船舶の名前や大きさ情報はない。

船舶の大きさを含めて表示する場合、両方を一度に読み出す必要がある。

mmsiをキーにVIEWを作る。

船舶位置のview

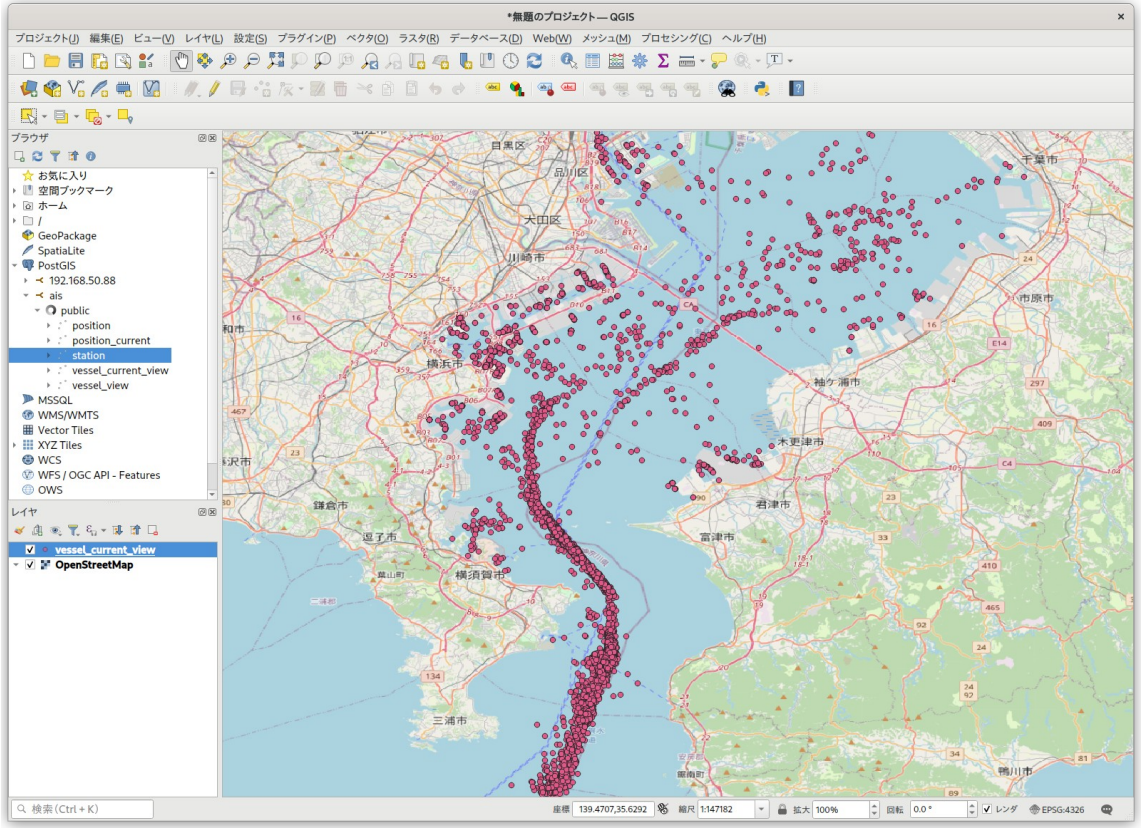
```
vessel_position (
    char(10) mmsi;          # AIS識別子
    numeric lon,lat;        # 緯度経度
    geometry geom;          # 緯度経度を空間座標系にしたもの
    timestamp  createat     # 格納時刻
    text name;              # 船舶名
    int type;               # 貨物船、客船などの種別
    int speed;              # 船舶スピード
    numeric heading;        # 舵の向き
    int status;             # 船の状態
    int to_bow;             # GPS位置から船首までの長さ(m)
    int to_stern;           # GPS位置から船尾までの長さ
    int to_port;            # GPS位置から左舷までの長さ
    int to_starboard;       # GPS位置から右舷までの長さ
);
```

現況データの表示

船舶位置と船舶情報は船舶位置viewで取得可能。
現況は最新の状況のみ必要だが毎回最新を問い合わせるのはDBにとって大きな負担。
最新位置のみを持つテーブルを作成しておけば、そのテーブルを参照するだけで現況は取り出せる。

船舶位置現況テーブルposition_currentを作り、
時系列の蓄積はpositiionにINSERT
現況はposition_currentにMMSIをキーとしUPSERT
(positionと同じスキーマ)
現況view vessel_current_viewも作る

受信したデータをQGIS上へ表示



現況テーブルであっても
一定時間で切り捨てないと
すべての船舶の最後にいた
場所は記録されてる。

AISは最大6分間隔で発信す
ることが決められているの
で、現況ビューの6分以内
を拾えば最新のみが表示で
きる。

Sunbit System

次のステップ

どうせならリアルタイム表示がしたい
(QGIS上でもリロードすれば表示できるけど...)

船の大きさがわかるんだから、実寸表示したい

船の航跡表示できないの？＜お客様(他人)ご要望

WEB上でリアルタイム表示を考える。

AIS情報をWFS(地物サーバ)でサービス

AIS情報をWEBブラウザから問い合わせできるようにする

地図サーバの検索機能をうまく使えば現況も航跡も表示できそう

ブラウザ側の表示ライブラリにはOpenLayersを使う
この機能で船舶オブジェクトを作る
　船舶情報から(地図上の)実寸オブジェクトにする

MapServer AIS情報をWFSサーバでサービス

PostGIS(Postgresqlの地図情報拡張)やshp等を地図DBとしてWMS、WFS準拠の問い合わせにより地図を返すCGIプログラム

proj(地図投影ライブラリ）により様々な投影座標系への変換が可能

内部的にはWGS84(EPSG:4326, GPS座標系と同等）でデータを格納、空間参照系はEPSG:3857（Google Mapと同じ)で行うと扱いやすそう。
(EPSG:3857は最初はEPSG:900913と呼ばれました。なぜでしょう？)

AIS情報を地図サーバでサービス

MapServerの定義にWFSとしてAIS位置情報を追加

```
 LAYER
   NAME 'vessel_pos'
   type POINT
   METADATA
     'wfs_featureid'       'vessel_pos'
     'wfs_enable_request'  '*'
     "wfs_getfeature_formatlist" "*,geojson"
     'gml_featureid'       'vessel_pos'
     'gml_include_items'   'all'
     'ows_title'       'vessel_pos'
     'wfs_srs'         'EPSG:4326 EPSG:3857'
   END
   CONNECTION "user=wms host=192.168.1.1 dbname=ais port=5432"
   CONNECTIONtype postgis
   DATA "geom from (select geom,id,to_number(mmsi,'999999999') as
mmsi,status,shipname,shiptype,callsign,lat,lon,(360 - heading) as heading,speed,(360 - course) as
course,turn,to_bow,to_stern,to_port,to_starboard,(to_bow + to_stern) as length,(to_port +
to_starboard) as width,draught,destination,create_at,created from vessel_view) as foo using unique
id using srid = 4326"
```

AIS情報を地図サーバでサービス

MapServerに問い合わせるとgeojsonでデータが返る

GET /cgi-bin/mapserv?
service=WFS&request=GetFeature&version=1.1.0&typename=vessel_pos&srsname=EPSG:4326&OUTPUTFORMAT=geojson&....


```
 {
"type": "FeatureCollection",
"features": [
{ "type": "Feature", "properties": { "id": "10032950", "mmsi": "431003094", "status": "0", "shipname": "AKEBONO MARU", "shiptype": "83", "callsign": "JD3204", "lat": "42.594772", "lon": "141.630180", "heading": "269.0", "speed": "21.5", "course": "286.0", "turn": "127", "to_bow": "75", "to_stern": "24", "to_port": "16", "to_starboard": "1", "length": "99", "width": "17", "draught": "4.6", "destination": "JP HKP OFF", "create_at": "2018-06-06 18:49:27", "created": "1528310967" }, "geometry": { "type": "Point", "coordinates": [ 141.63018, 42.594772 ] } }
 }
```

AIS情報を地図サーバでサービス

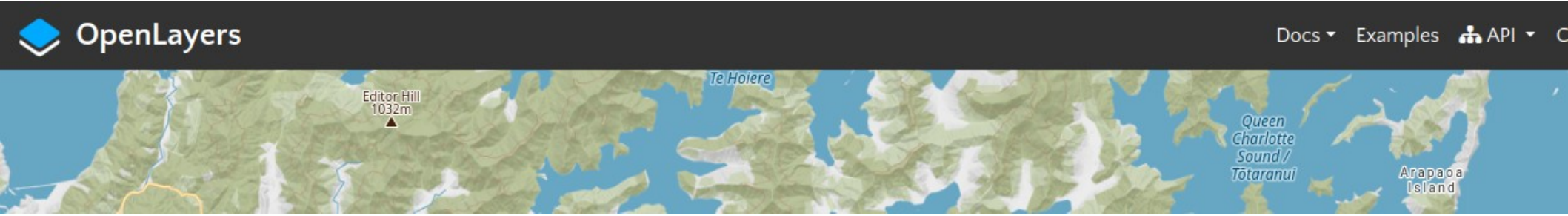
気をつけること

WFSの時刻検索は数値比較でないとうまくいかない
（UNIX Epoc等。 YYYY-MM-DD形式はダメ）

できないと現況のための検索や、一定時間内の航跡表示の検索ができない

>viewにUNIX Epocを追加しておくなどの対応が必要

OpenLayersで船舶を描画する



ブラウザで地図データを表示する、JavaScriptライブラリ。2条項 BSDライセンスで提供。

船舶を地図上の原寸で書きたい。

OpenLayersで地図データを描画する手順

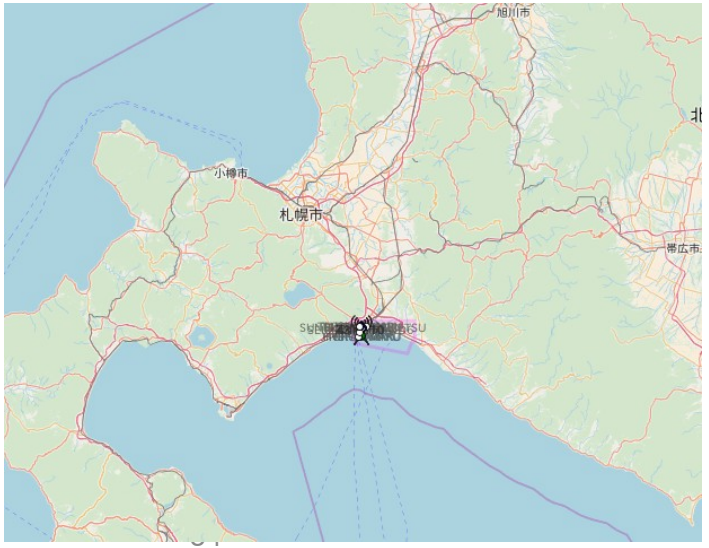
平面直角座標系(投影座標)や初期表示範囲を決める

ベースレイヤを作成する　(OSMでいいよね)

重ねるレイヤを作成する　(船舶情報)

Mapオブジェクトにレイヤを追加する

レンダリング



Sunbit System

空間座標系や初期表示範囲

位置情報はEPSG:4326, その投影座標はEPSG:3857
中心緯度経度は表示したいところの真ん中
初期表示範囲はズーム値で14くらい
という感じでViewを作る。


```
view = new ol.View({
          center: ol.proj.transform(
                [141.6263,42.6337],
                'EPSG:4326','EPSG:3857'),
          zoom: 14
        });
```

ベースレイヤを作成する

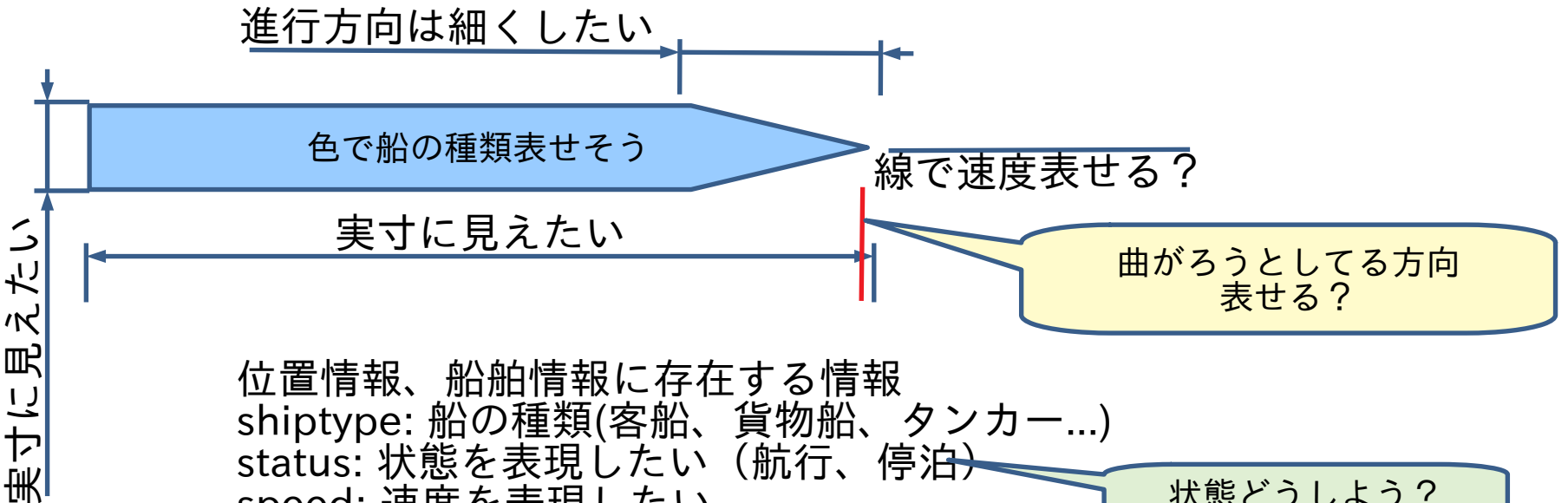
OpenStreetMapを使うなら簡単

```
// ベースレイヤを作成
lbase = new ol.layer.Tile({
            source: new ol.source.OSM()
        });
```

タイル型レイヤに
ソースとしてOSMを指定するだけ

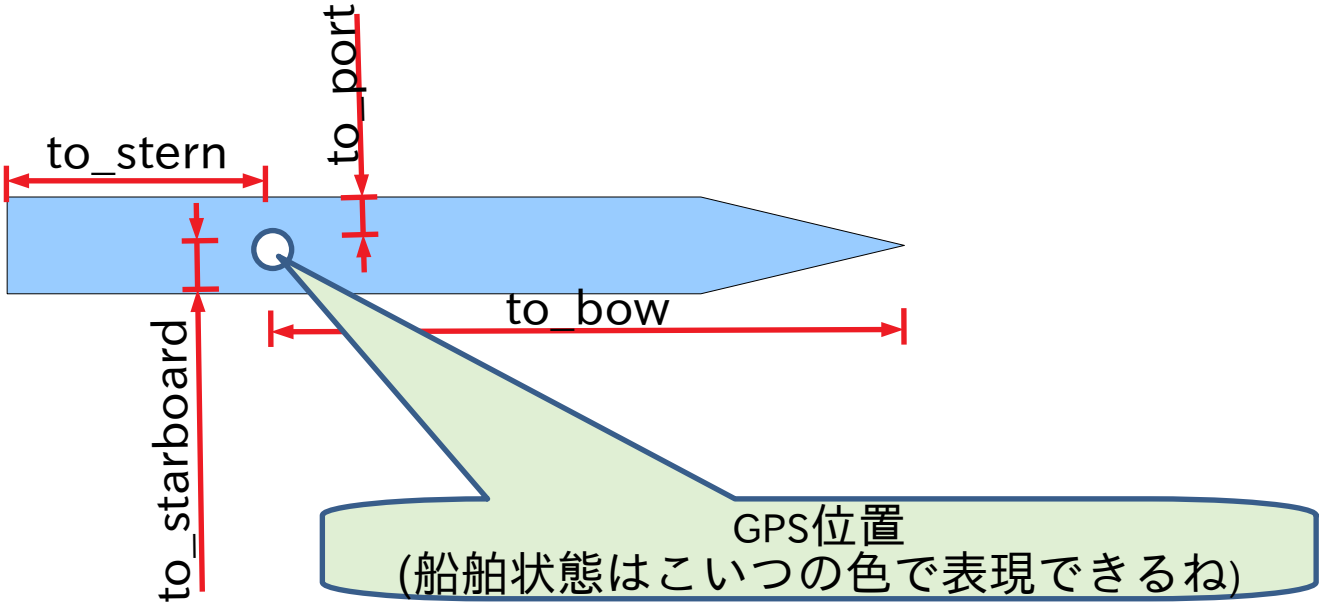
重ねるレイヤを作成する　(船舶情報)

船舶情報をどう表現する？



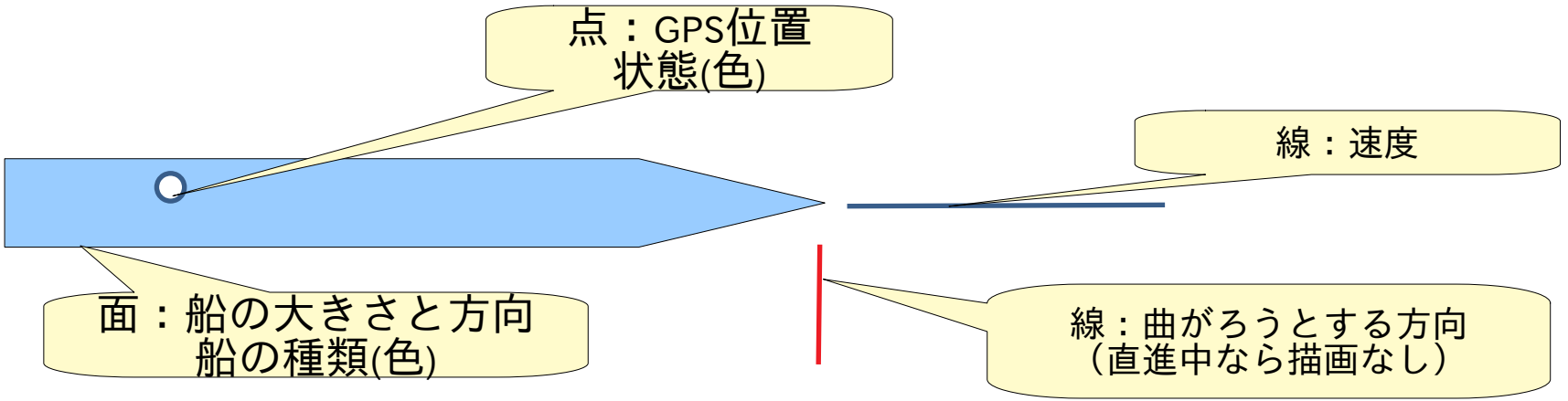
重ねるレイヤを作成する　(船舶情報)

GPS位置情報は点情報
　to_bow, to_starboard てそういうことか！



重ねるレイヤを作成する　(船舶情報)

必要なレイヤ



位置情報、船舶情報に存在する情報
shiptype: 船の種類(客船、貨物船、タンカー...)
status: 状態を表現したい（航行、停泊）
speed: 速度を表現したい
turn:　曲がろうとしてるかを知りたい

重ねるレイヤを作成する　(実際の長さ比率)

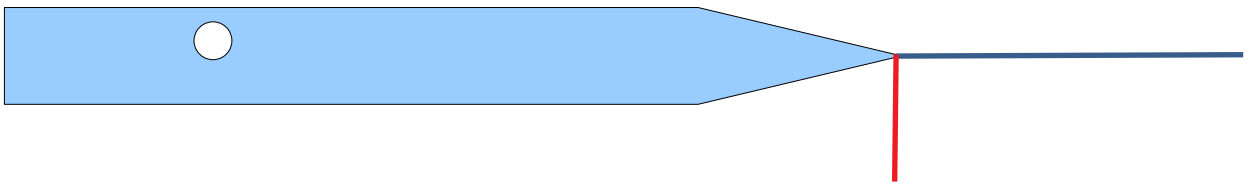
縮尺係数

平面直角座標系は原点位置からの距離を表わしている
　http://www.gsi.go.jp/sokuchikijun/datum-main.html
原点から遠い程、同じ距離は地図上で長く投影される
＝同じ船を書いても原点からの距離で書くべき長さが変わる

Openlayersには縮尺係数を得る関数がある
```
e = getExpansion([緯度, 経度]);　//位置で係数を得る
bow = to_bow * e;　　// その位置での描画長さを得る
```

重ねるレイヤを作成する　(船舶情報)

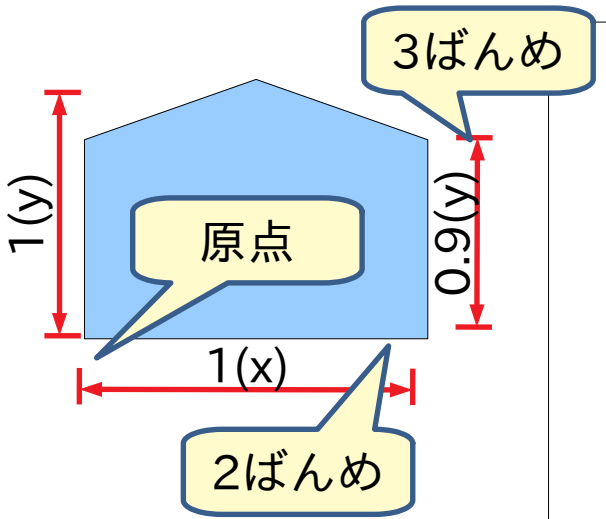
1) GPS位置に点を描き、状態色で塗る
2) 船舶の基準形に全長全幅を乗じて形を作り
　GPS位置からto_port, to_stern分ずらして配置し、船舶の方向に傾けて描く。船舶種別色で塗る
3) 船舶の先端位置から速度分の線を進行方向に描く
4) 旋回中であれば速度線と同じように線を描き、
　旋回方向に回転する



レイヤに描く情報
shiptype: 船の種類で色を決める
status: 船の状態で色を決める
speed: 速度と長さを比例関係で長さを決める
turn:　長さは一定で旋回方向に回転する

船舶のかきかた

船舶の元画像

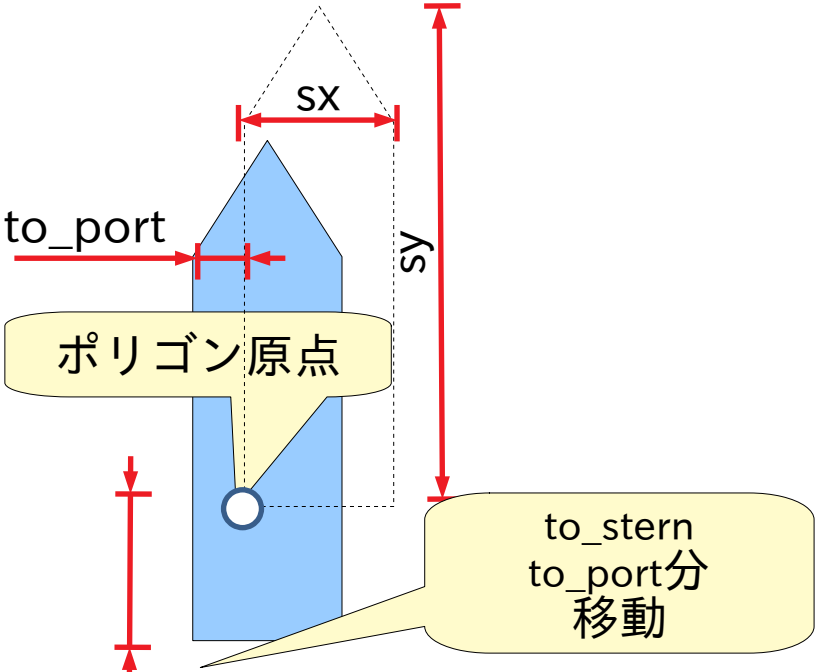


描画上の実寸値でpolygon作成

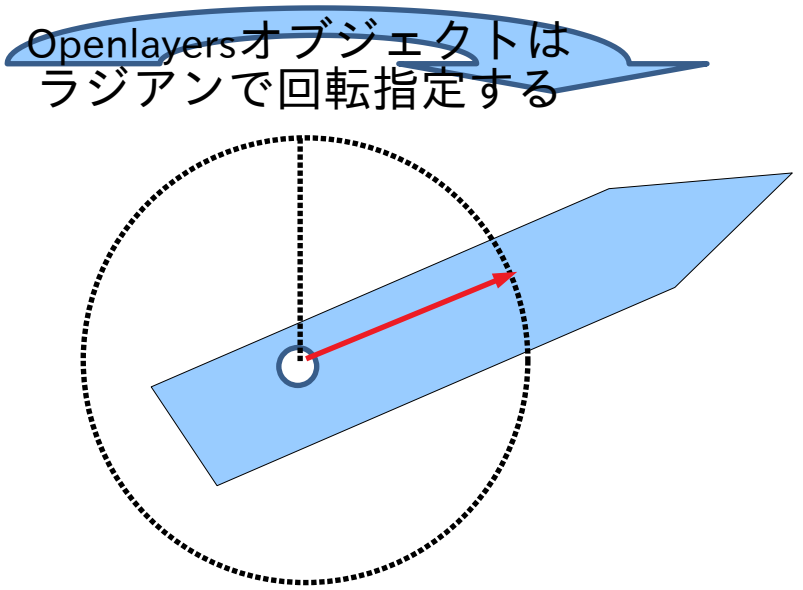
```
// 縮尺係数
e = getExpansion(g.getCoordinates());

// 全長(x), 全幅(y)
sx = (to_port+to_starboard) * e;
sy = (to_bow+to_stern) * e;

 // 船舶の形を作成
 poly = new ol.geom.Polygon([[
        [sx*0, sy*0],              // 原点
        [sx*1, sy*0],              // 2ばんめ
        [sx*1, sy*0.90],           // 3ばんめ
        [sx*0.5, sy*1],
        [sx*0, sy*0.90],
    ]]);
```

Sunbit System

船舶のかきかた




```
// cx,cyはGPS位置

poly.translate(cx-(to_port*e),
                    cy-(to_stern*e));
```




```
// AISは度で来るのでラジアンに
angle = p.heading * (Math.PI / 180);

// GPS位置を中心に回転
poly.rotate(angle,[cx,cy]);
```

Sunbit System

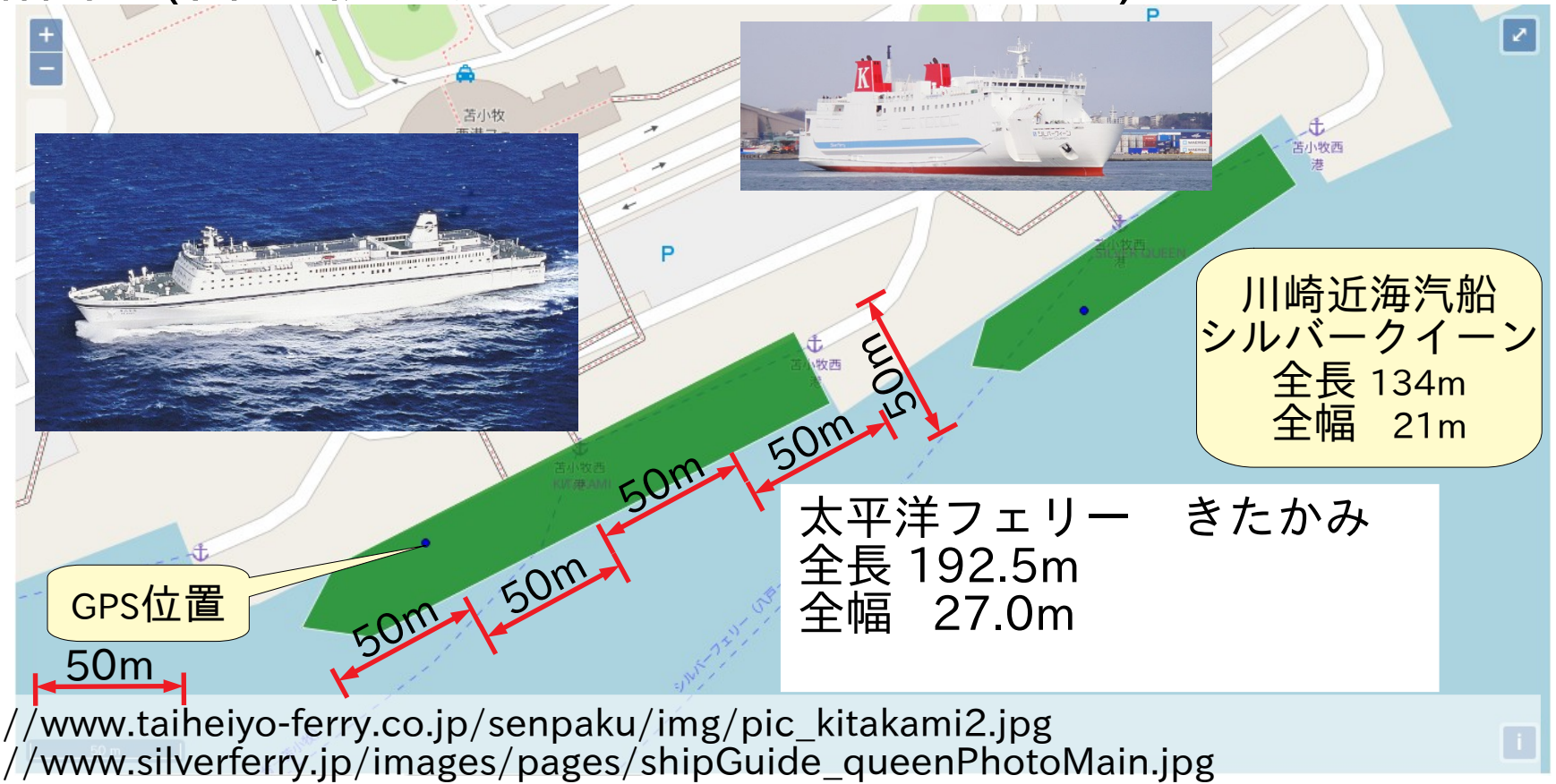
ここまでの考え方で船舶をレイヤに置く

```
// ベクタレイヤのソースはMapServerへの問い合わせ
svessel = new ol.source.Vector({format: new ol.format.GeoJSON(),
        url: function(extent){ return 'http://wms.3bit.co.jp/cgi-bin/mapserv?'...
function createStatus( );    // GPS位置と船舶状態のポリゴン作成
function createCourse( );  // 速度線を作成
function createTurn( );     // 転回方向線を作成
function createVessel( );   // 船舶のポリゴンを作成

// 同じデータソースを元にそれぞれの船舶情報のレイヤを作る
lstatus = new ol.layer.Vector({source: svessel, style: createStatus});

lcourse = new ol.layer.Vector({source: svessel,opacity: 0.7,
                               style: createCourse});

lturn = new ol.layer.Vector({source: svessel, opacity: 0.7,
                               style: createTurn });

lvessel = new ol.layer.Vector({source: svessel,opacity: 0.7,
                               style: createVessel});
```

レイヤ合体

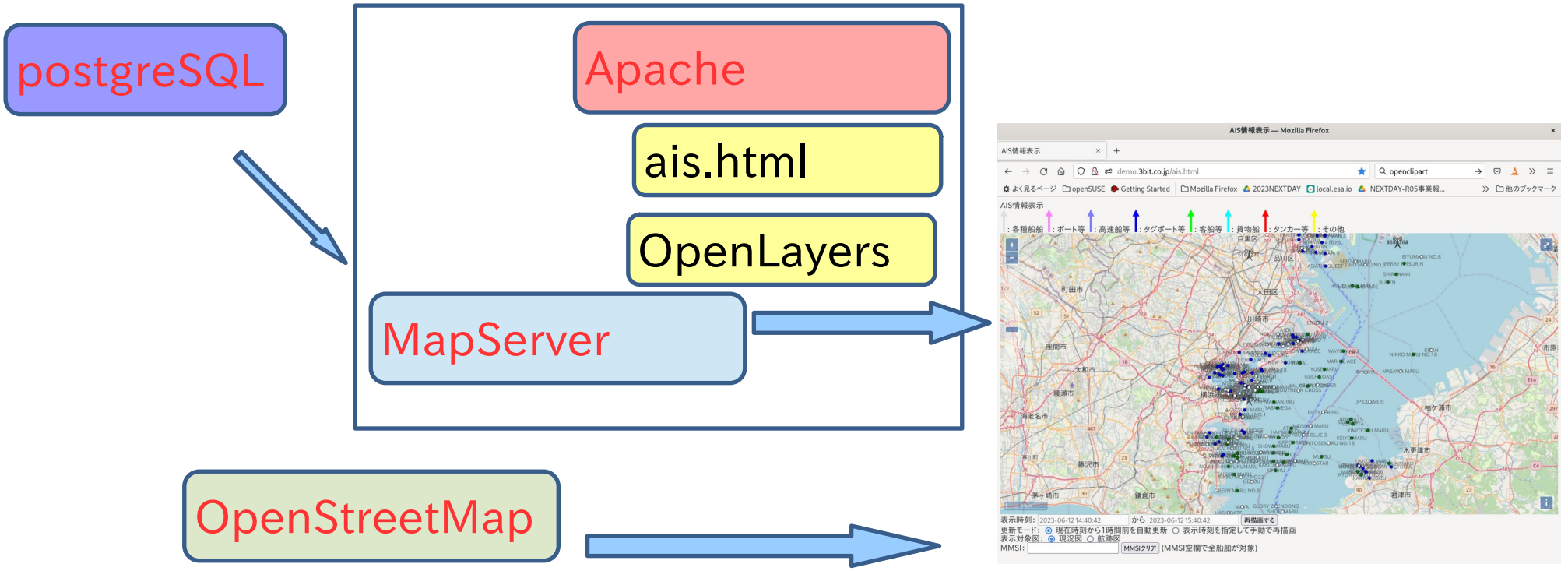
```
// スクロールバー、拡縮などの操作部品
controls = new ol.control.defaults().extend([
                new ol.control.MousePosition({projection: 'EPSG:4326',
                        coordinateFormat: ol.coordinate.createStringXY(4)}),
                new ol.control.ScaleLine(),
                new ol.control.ZoomSlider(),
                new ol.control.FullScreen()]);

// マップ生成
map = new ol.Map(
            {target: 'map',                                          //DOMのmapに描画
             layers: [lbase,lvessel,lcourse,lturn,lstatus],  // レイヤ描画は左が下
             view: view,                                             // 先に作った表示位置
             controls: controls                                      // ブラウザの操作部品
            });
```

表示結果（苫小牧西港フェリーターミナル）



http://www.taiheiyo-ferry.co.jp/senpaku/img/pic_kitakami2.jpg
http://www.silverferry.jp/images/pages/shipGuide_queenPhotoMain.jpg

DBからWEB表示まで



おさらい

東京湾の現況

http://demo.3bit.co.jp/ais.html

gpsdais2pgsql.plはMapServerの設定ファイルや船舶描画のJavascriptと共に公開しています。
http://ossdesk.3bit.co.jp/?action=cabinet_action_main_download&block_id=104&room_id=1&cabinet_id=6&file_id=28&upload_id=60

自治体向け施設利用申請管理システム

あまつみ

http://amatsumi.net/

Sunbit System

Sunbit System　有限会社サンビットシステム

- 1998年8月25日　札幌市東区に登記
- 2009年より札幌市豊平区平岸
- やってること
  - ITインフラ構築、ITシステム設計支援
  - ソフトウエア受託開発,パッケージ開発
  - IT教育など
-