

A large satellite dish antenna is the central focus, mounted on a complex metal lattice structure. It sits on a grassy hill. In the background, there are rolling hills and a body of water under a sky filled with dramatic, grey clouds. A small white control building is visible at the base of the antenna's support structure.

Decoding the air around you

with Java and \$7 hardware

Bert Jan Schrijver

bertjan@jpoint.nl



[@bjschrijver](https://twitter.com/bjschrijver)

Let's meet



Bert Jan Schrijver

jpoint.

.nl.
jug

Outline

- Background
- Software Defined Radio
- ADS-B
- AIS
- Decoding ADS-B & AIS
- Demo
- Q&A

How it all started...

Software Defined Radio

- Radio communication system
- Components implemented in software
- Cheap DVB-T with RTL2832U chip (\$7)
- Hence: RTL-SDR
- Details on www.rtl-sdr.com

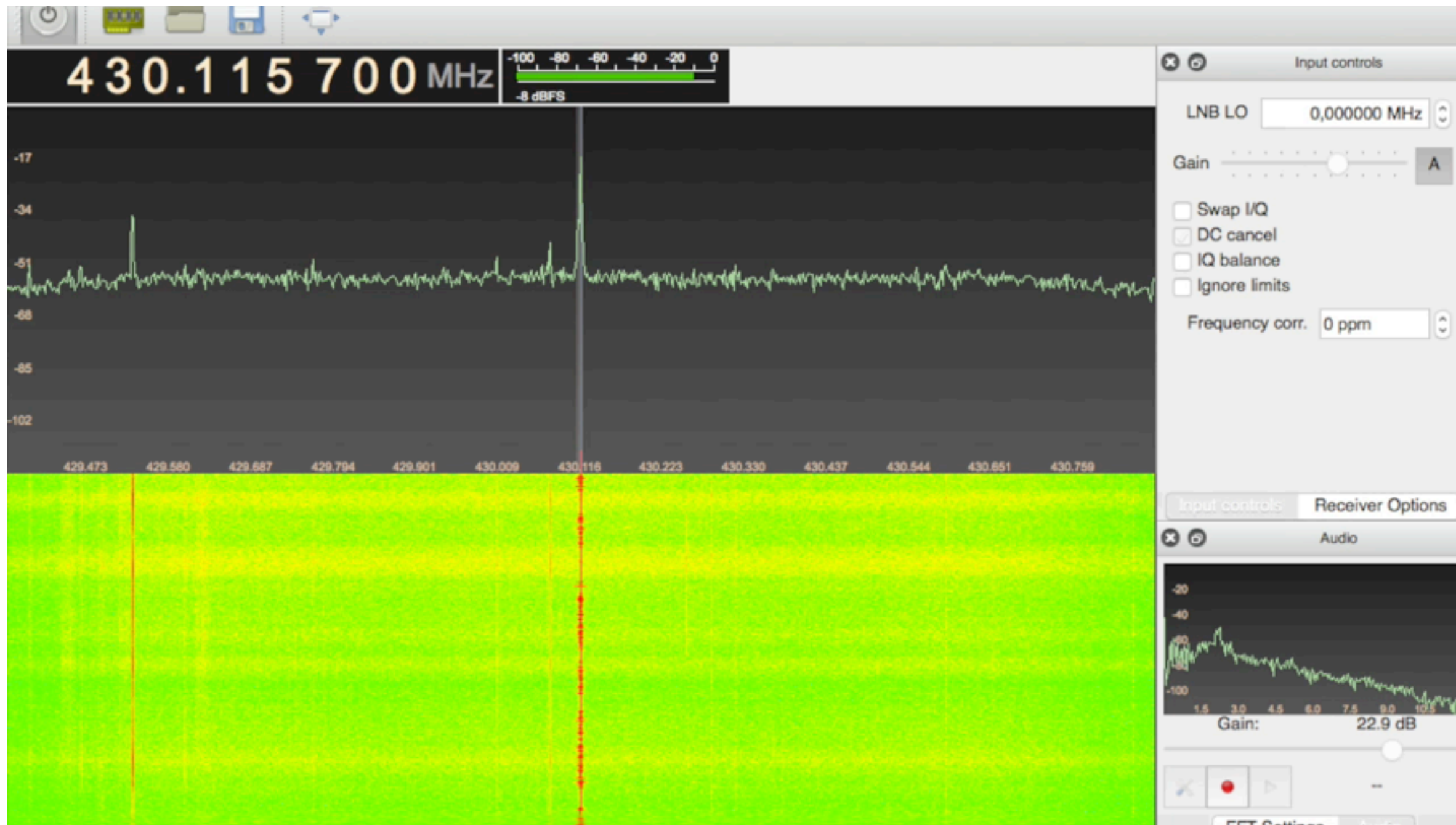
Software Defined Radio



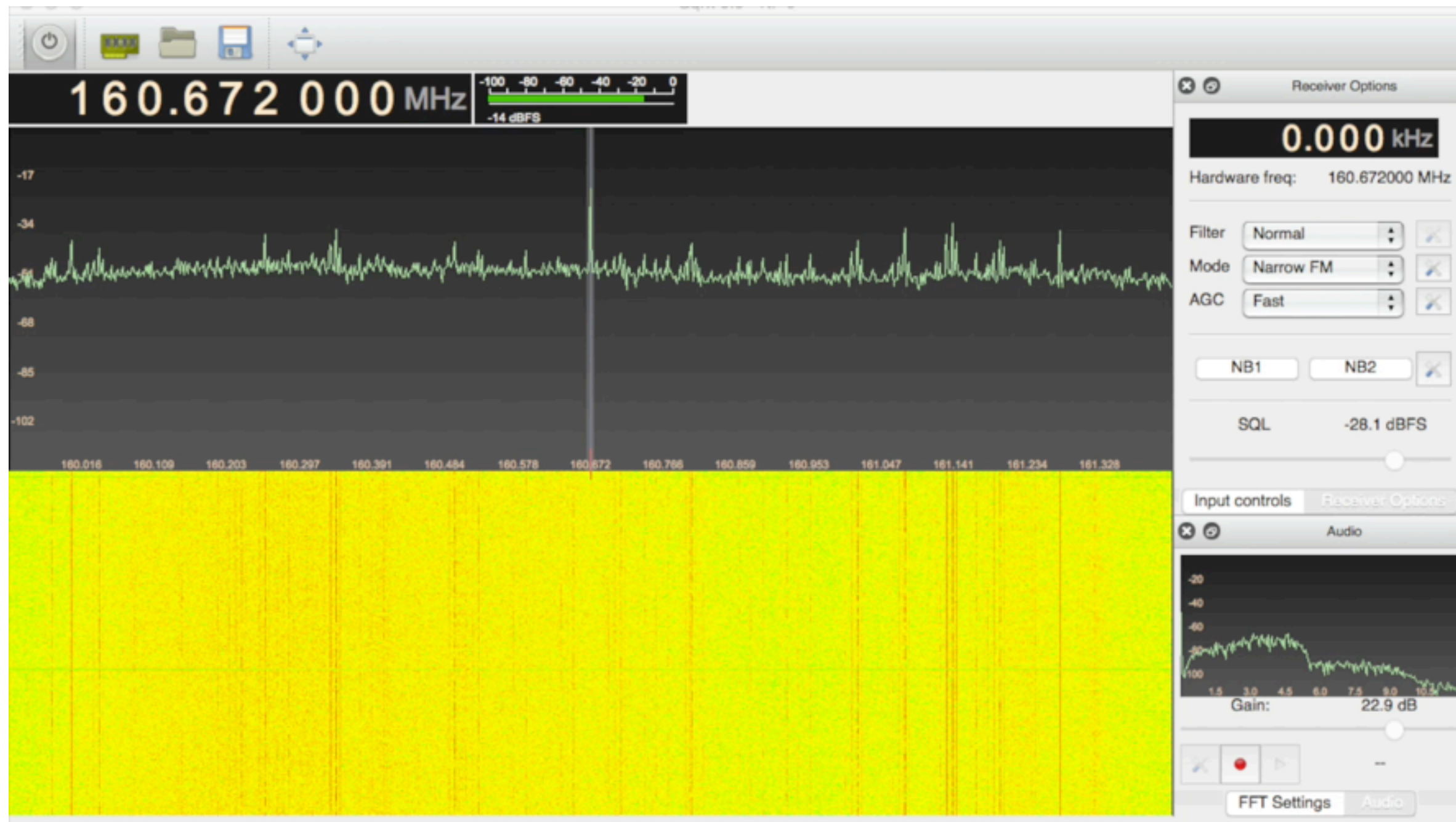
DEMO



Amateur radio



Vessel to shore



POCSAG pager messages

```
bertjan — grep — 118x27
POCSAG1200: Address: 119978 Function: 0 Alpha: a<CAN>
POCSAG1200: Address: 127612 Function: 3 Alpha: 72/1/VP/rg-08/rg-08_P01/ProcedureP01Geen pomp beschikbaar
POCSAG1200: Address: 168881 Function: 3 Alpha: Ketelhuis
POCSAG1200: Address: - Function: - Alpha: K-K<FF>,,<CR>(LLnlLnl<FF>(PUö;<VT>L<FF>,<FF>h<DC4>=<CAN><FS><YÄöää
<FS>=Y<FS>H<FF>(<VT>h<DC4>üÄ<Zä<EM>h<DC4>==Üö<FS>Üöä;<ESC>
POCSAG1200: Address: 175828 Function: 3 Alpha: ÄBEL MBXÜ<EOT><EOT>
POCSAG1200: Address: 175628 Function: 3 Alpha: ÄVan:Ühobbit@ordo.Ä0nderwerp:ÜHobbit Ä870392Ü sapepo:proc
s CRITICAL (RED) ÄInhoud:Üred Mon Jun 15 12:40:27 CES
POCSAG1200: Address: - Function: - Alpha: <EOT><ESC>X5VV<NAK><SYN><ACK><DC4>V
POCSAG1200: Address: - Function: - Alpha: Üß
POCSAG1200: Address: 175828 Function: 3 Alpha: ÄBEL MBXÜ<EOT><EOT>
POCSAG1200: Address: 167455 Function: 3 Alpha: Nagios: PROBLEM W302_eventlog is CRITICAL on enomta1-1.emd.
.net<EOT><EOT>
POCSAG1200: Address: 172992 Function: 3 Alpha: Nagios: ACKNOWLEDGEMENT W302_eventlog is CRITICAL on enomta1-1.emd
.ne
POCSAG1200: Address: 1989211 Function: 1 Alpha: K<DC2>
POCSAG1200: Address: 167455 Function: 3 Alpha: Nagios: ACKNOWLEDGEMENT W302_eventlog is CRITICAL on enomta1-1.emd
.ne
POCSAG1200: Address: 145513 Function: 3 Alpha: SCH_CTF 11:42 2 SUCTION PR. P-1201B L
POCSAG1200: Address: 145513 Function: 3 Alpha: SCH_CTF 11:42 2 SUCTION PRESS P-1201A L
POCSAG1200: Address: 172992 Function: 3 Alpha: Nagios: PROBLEM W302_eventlog is CRITICAL on enomta
POCSAG1200: Address: - Function: - Alpha: +9<ESC>Y+YäÄ=<CAN>=zÄÄK;<EM>
POCSAG1200: Address: 167488 Function: 3 Alpha: RG Nieuwe Tonge Alarm prio 3
POCSAG1200: Address: 171714 Function: 3 Alpha: Gebruiker die de Eerste lijns dienst draait heeft zich aan
gemeld.
POCSAG1200: Address: 167488 Function: 3 Alpha: RG Achthuizen Alarm prio 3<EOT><EOT>
```

FLEX pager messages

```
bertjan — root@bj-virtualbox: ~/src/sdr/receivers/flex — ssh — 100x24
root@bj-virtualbox:~/src/sdr/receivers/flex# ./rtl_flex_noX.py -f 169.645M --rx-gain=37.2 --device=rtl=0 2>&1 | grep -v 'IUNKI'
linux; GNU C++ version 4.8.2; Boost_105400; UHD_003.005.005-0-unknown

No database support
gr-osmosdr 0.1.1 (0.1.1) gnuradio 3.7.2.1
built-in source types: file osmosdr fcd rtl rtl_tcp uhd hackrf bladerf rfspace
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Exact sample rate is: 250000.000414 Hz
Using Volk machine: ssse3_64_orc

0 169.645| 920162|ALNIB2 Pet Scan Rijnstate Route 02 Wagnerlaan 55 Arnhem 6815AD 55 42328
0 169.645| 1520021|ALNIA1 van Vredenburgweg/Burgemeester Elsenlaan RSK : 15121 Ritnr: 95377
0 169.645| 1123127|ALNIA1 5708GM 23 : Kloosterstraat Helmond Obj: praktijken van huisa Huisartsenpraktijk Stiphout Rit: 45649
0 169.645|1599375360|ALNI7.YP 2 Dienstver. Instanties (Openen Deur) (+Inc.net: Reg.Inmeld+) , Stoute nburggracht 30 2e Amsterdam [ AL
0 169.645| 5000|ALNI155-C2000 Testpage
0 169.645| 1520020|ALNIB1 Laan van Meerdervoort 597 DHG 2564AA : 15120 Ritnr: 95378
0 169.645| 1180000|ALNITESTOPROEP GMC BN (1)
0 169.645| 726566|ALNIA2 Standplaats Amersfoort Noord Het Hofslot 1 3823VV 1 Amersfoort
0 169.645|1144915968|ALNI/1.?B2 Warm Thuis De Hulst Noordschermerdijk 5 Oterleek - 1842E
```

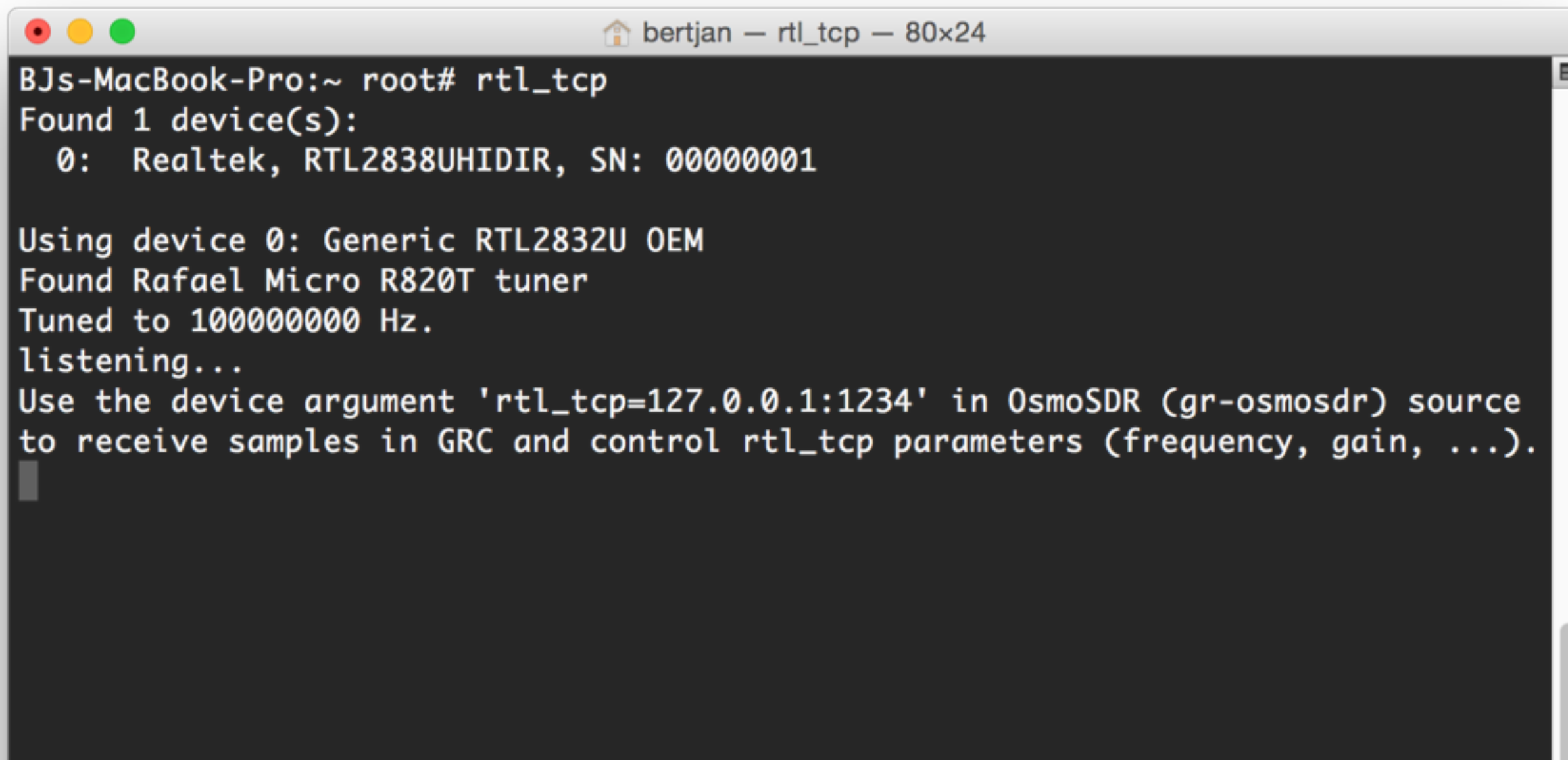

DEMO



So how about **Java**?

Java and RTL-SDR

RTL_TCP: Provides socket for RTL-SDR device

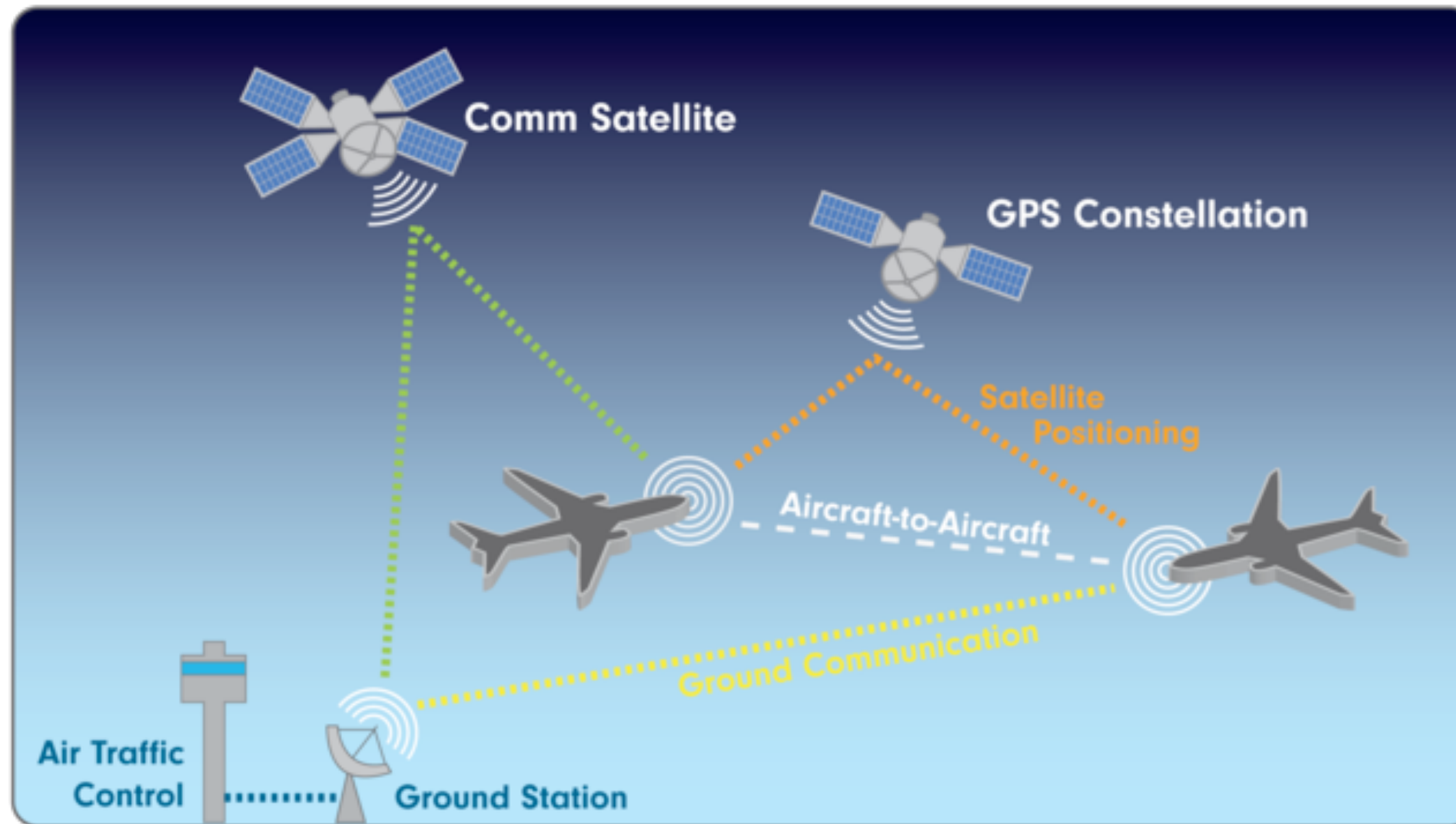
A screenshot of a macOS terminal window titled 'bertjan — rtl_tcp — 80x24'. The terminal shows the execution of the 'rtl_tcp' command. It reports finding one device (Realtek, RTL2838UHIDIR, SN: 00000001) and using it (Generic RTL2832U OEM). It also reports finding a Rafael Micro R820T tuner and tuning to 100000000 Hz. The terminal ends with 'listening...' and a usage instruction: 'Use the device argument 'rtl_tcp=127.0.0.1:1234' in OsmoSDR (gr-osmosdr) source to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).'.

```
BJs-MacBook-Pro:~ root# rtl_tcp
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Tuned to 100000000 Hz.
listening...
Use the device argument 'rtl_tcp=127.0.0.1:1234' in OsmoSDR (gr-osmosdr) source
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).
```

ADS-B

Automatic Dependent Surveillance Broadcast

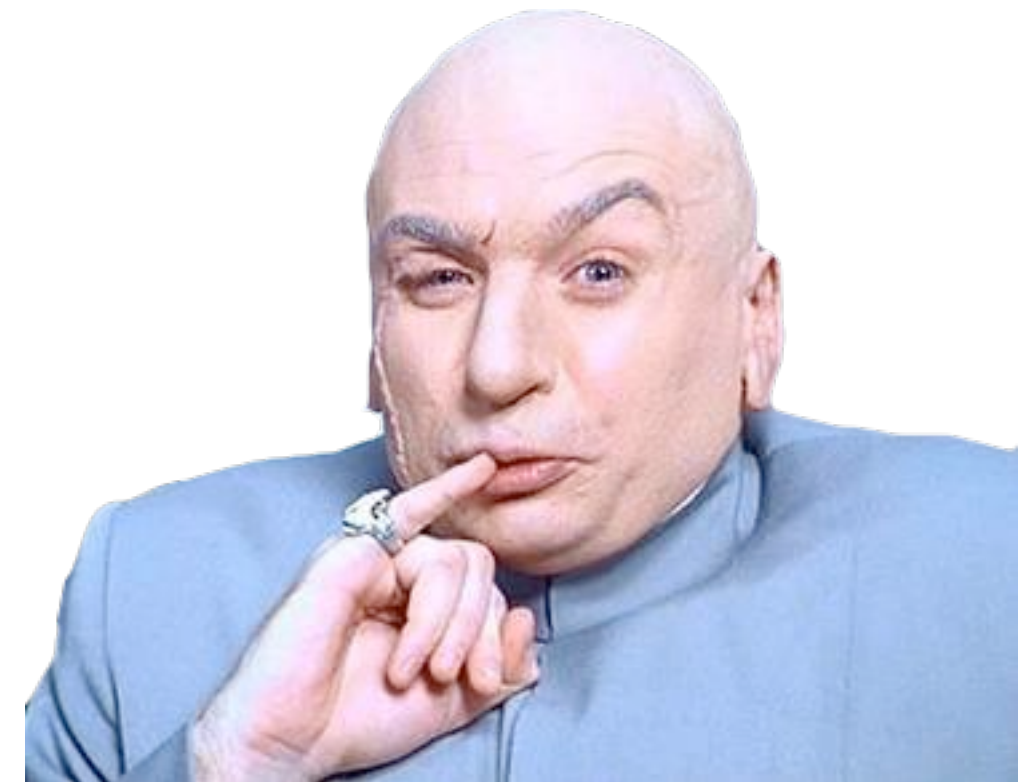


Source: simflight.nl

Java and ADS-B

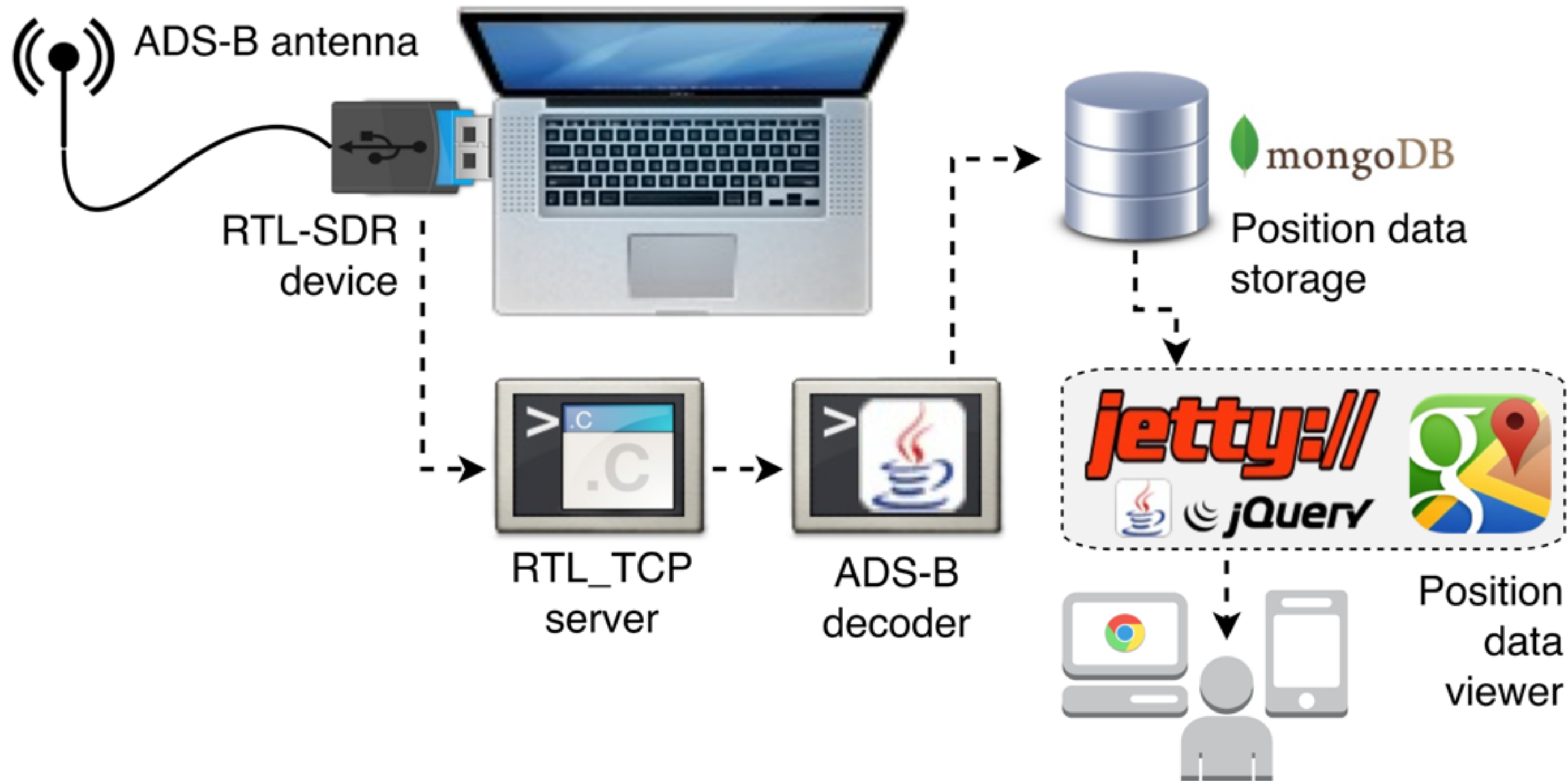
- Java connects to RTL_TCP socket
- But now what?
- No open source Java decoders available
- No other option than to write own decoder...

...or is there?



(insert legal notice here)

How stuff works



DEMO



MongoDB data structure

```
{
  "_id" : ObjectId("5583b57bd4c694d7f93fea68"),
  "heading" : 270,
  "latitude" : 51.478271,
  "objectId" : "KLM1001",
  "objectType" : "AIRCRAFT",
  "longitude" : -0.102589,
  "timestamp" : NumberLong("1434695035486"),
  "createdAt" : ISODate("2015-06-19T06:23:55.487Z")
}
```

API data structure

```
{  
  "from":10000,  
  "to":4850,  
  "positions":  
    {  
      "positions":  
        {  
          "heading":"164.0",  
          "objectId":"AFR187F",  
          "objectType":"AIRCRAFT",  
          "timestamp":"Wed Oct 21 12:36:20 PDT 2015"  
        },  
      },  
    },  
}
```


Frontend

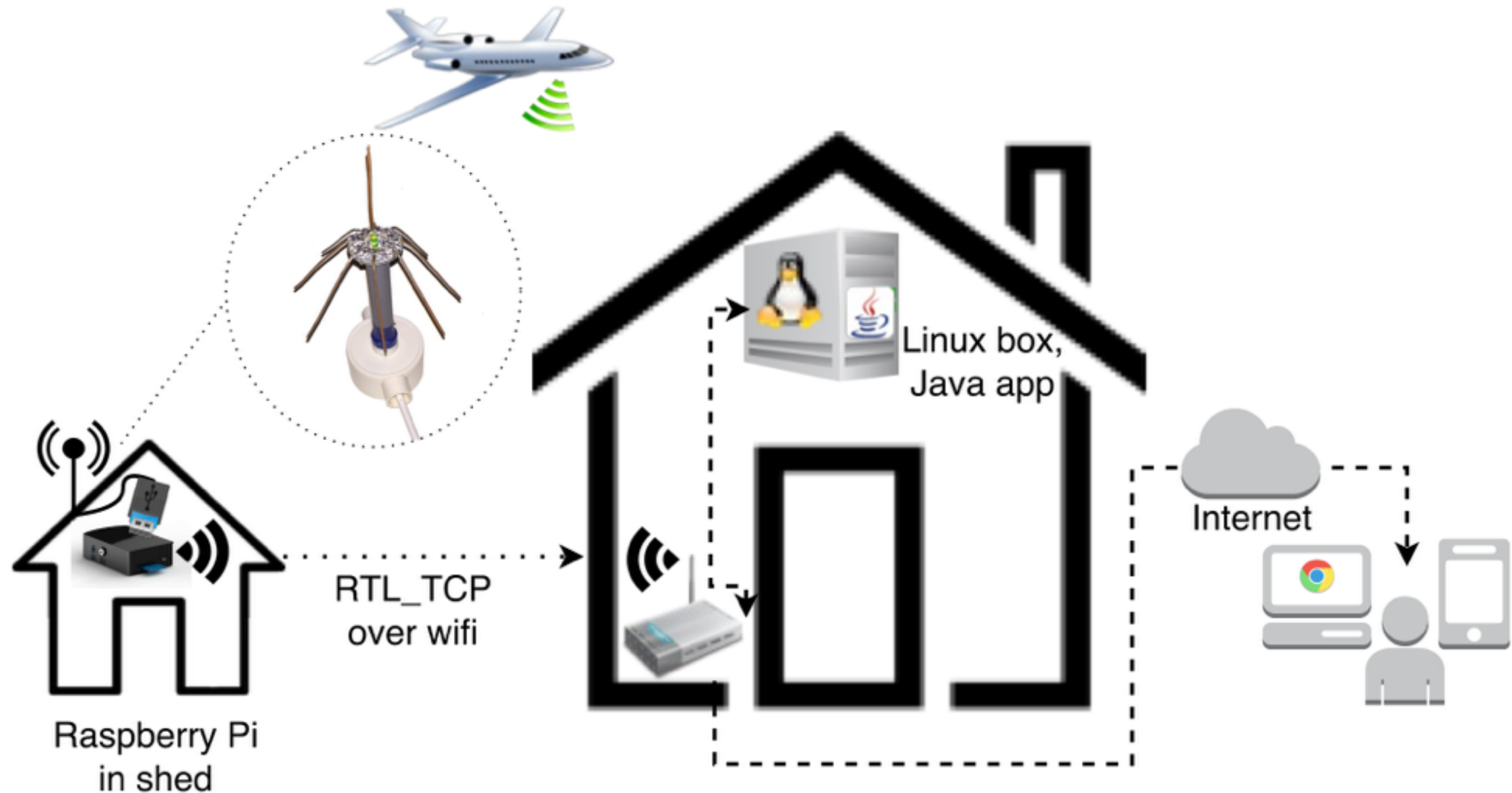
```
marker = new google.maps.Marker({  
    objectId: objectId,  
    position: curPos,  
    title: objectId + ' at ' + timestamp + ' - click to view details',  
    icon: planeIcon,  
    map: map  
});
```

```
// Draw track.  
var track = new google.maps.Polyline({  
    path: trackCoordinates,  
    geodesic: true,  
    strokeColor: pathColor,  
    strokeOpacity: 0.8,  
    strokeWeight: 0.7  
});
```

That's cool, but...

- In-house reception is sub-optimal
- Always-on service would be nice
- Laptop needed for work now and then

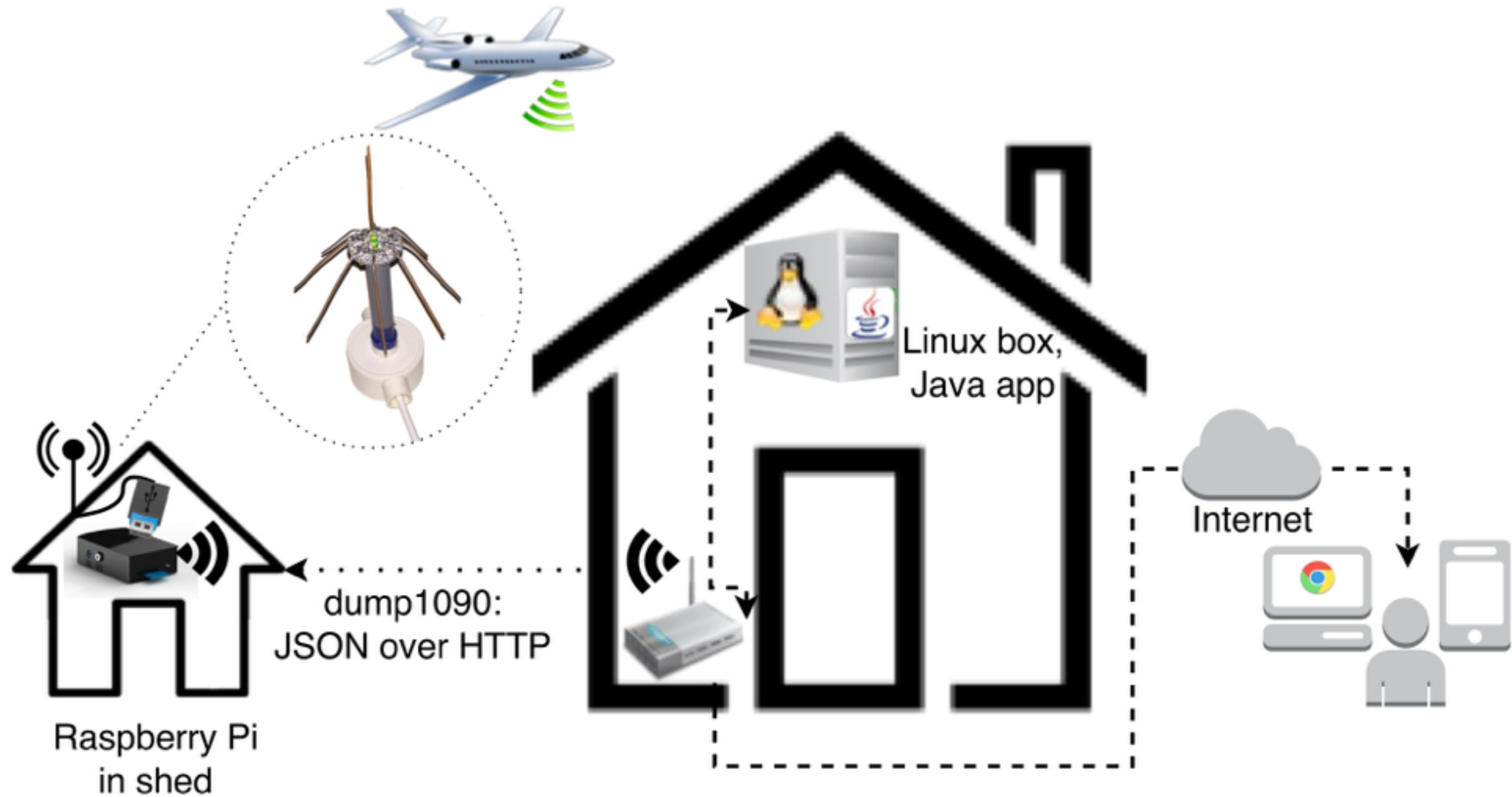
Getting serious



That's cool, but...

- Raw RTL_TCP stream is huge
- Problematic over wifi
- Would like to give AIS another try
- Want to throw out horrible decompiled code

Getting serious - v2.



Getting serious - v2.

bertjan — bertjan@raspberrypi: ~ — ssh — 80x24

Hex	Mode	Sqwk	Flight	Alt	Spd	Hdg	Lat	Long	Sig	Msgs	Ti/
400935	S								6	1	3
484C53	S			1683					6	7	1
406131	S			738	390	160	52.283	4.833	6	19	0
471F5F	S	3443	WZZ4NJ	11880	748	072	52.577	4.197	6	218	1
4B1A37	S	2023	EZS1522	11887	827	159	52.337	4.882	9	1140	0
484C24	S	1000	KLM1359	3769					7	736	3
484CB7	S	1000	KLM1969	3251	550	142	52.141	4.998	23	1334	0
3986E5	S	5605	AFR1764	11270					7	787	1
4492E3	S	7105	BEL1EW	7417	618	019	52.020	4.486	12	766	0
471F57	S	2017	WZZ397	10357	940	262	52.054	4.537	12	3309	0
47853B	S	7567	NAX97B	11270	772	020	52.218	4.242	8	2247	3
424946	S	3521	AFL2584	10357	905	240	52.277	4.118	9	1596	1
471F4F	S	2704	WZZ767	9389	942	277	51.888	3.703	8	3234	0
405A48	S	6620	BAW30CV	11552	900	269	51.762	3.252	6	1589	1

dump1090:
JSON over HTTP

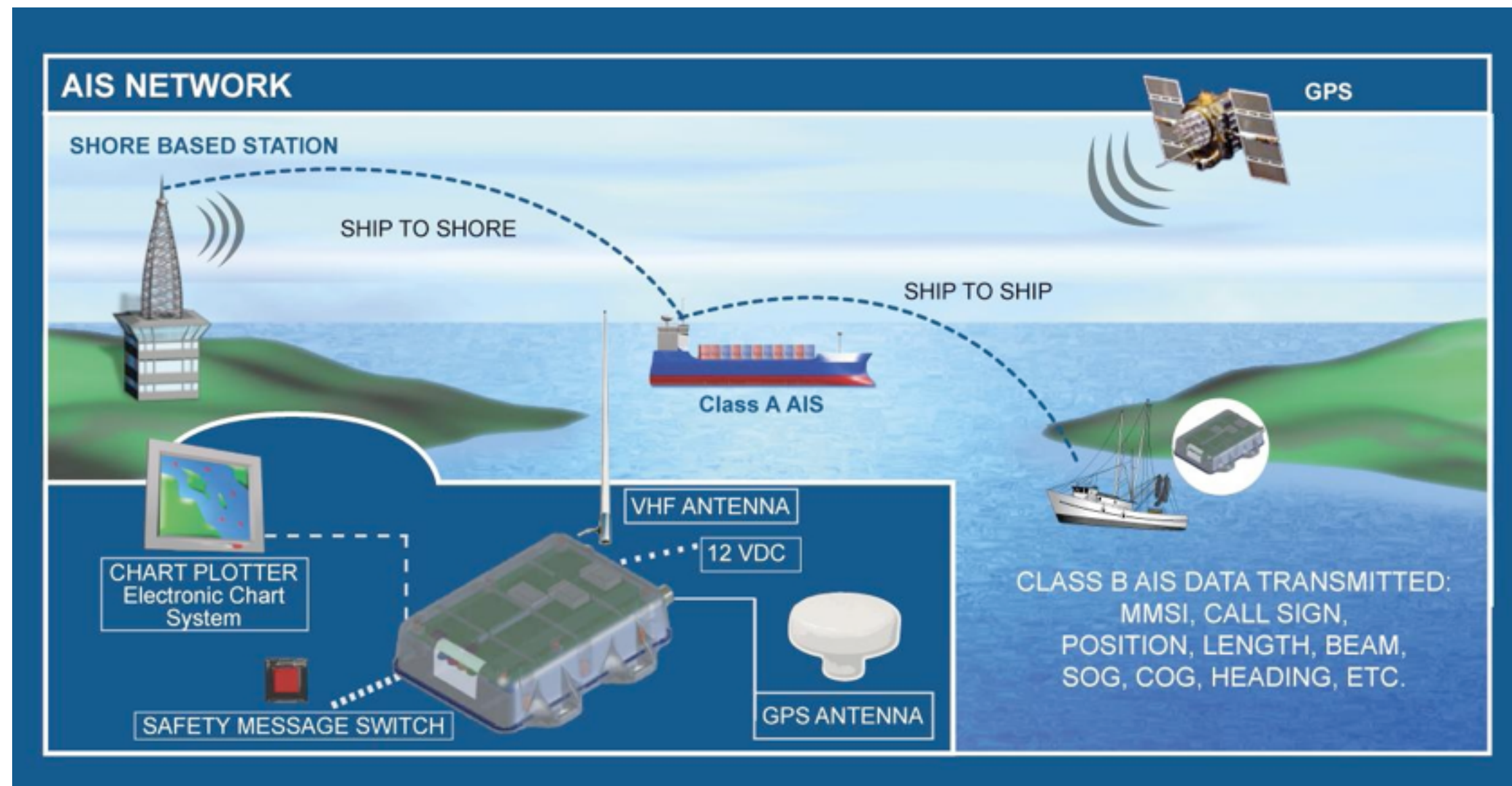
Raspberry Pi
in shed

Linux box
Java app

Internet

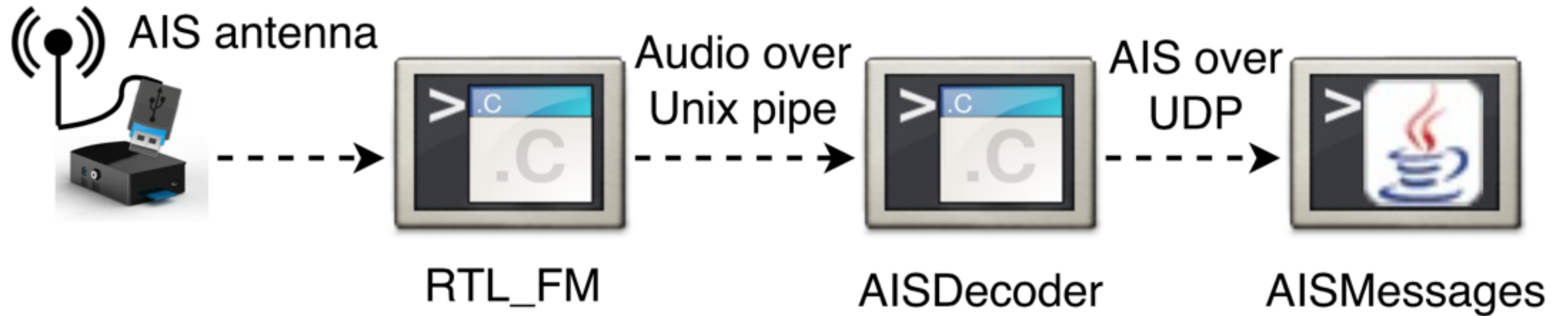
AIS

Automatic Identification System



Source: hollandmarinehardware.nl

Decoding AIS



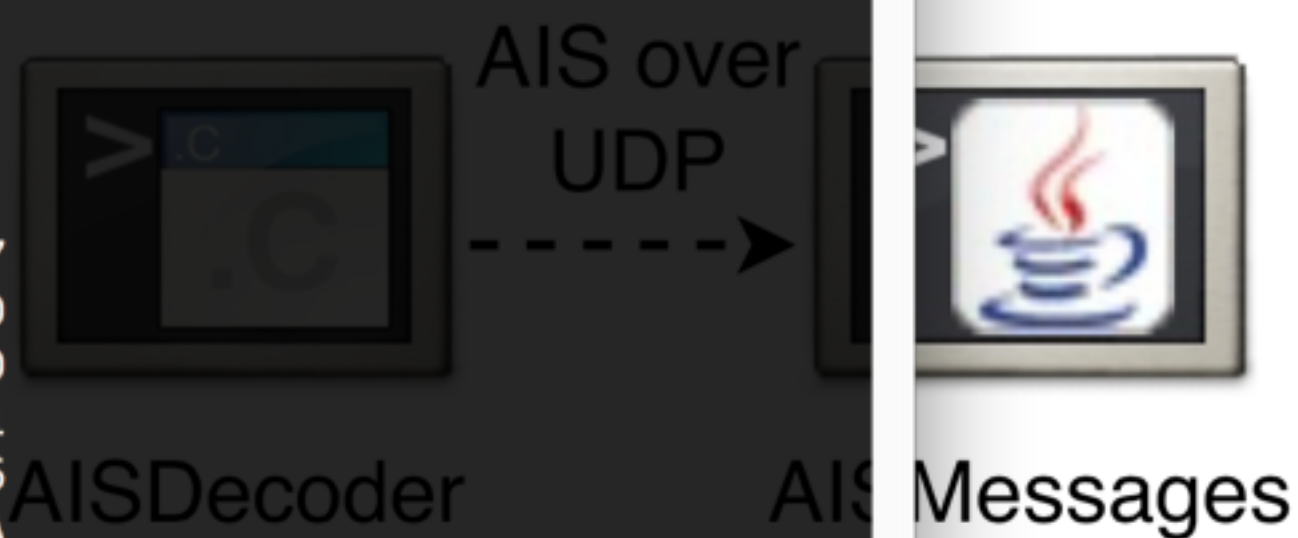
```
rtl_fm -g 9999 -f 161975000 -p 10 -s 48k -r 48k | aisdecoder -h 192.168.2.2 -p 1234 -a file -c mono -d -f /dev/stdin
```


Decoding AIS

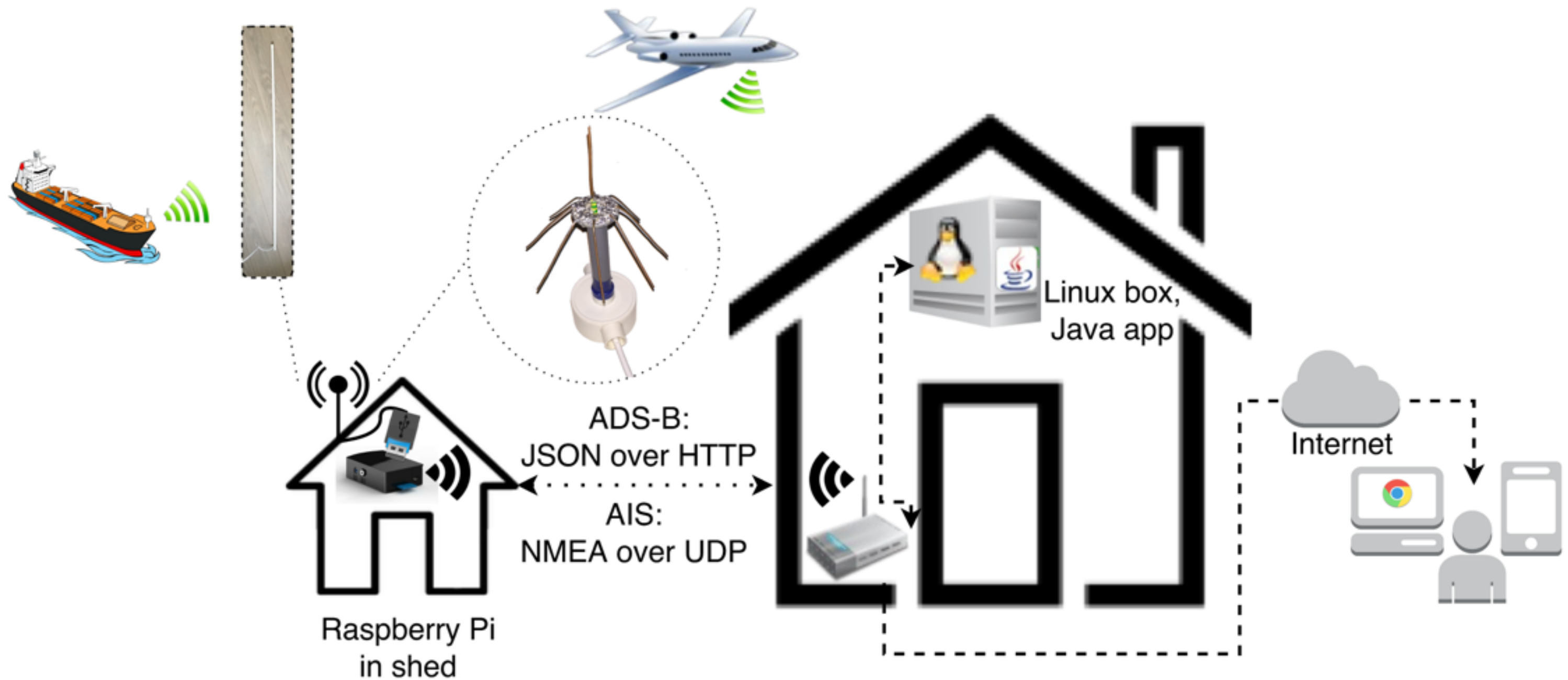


```
bertjan — bertjan@raspberrypi: ~ — ssh — 80x24
Found Rafael Micro R820T tuner
Tuner gain set to 49.60 dB.
Tuner error set to 10 ppm.
Tuned to 162227000 Hz.
Oversampling input by: 21x.
Oversampling output by: 1x.
Buffer size: 8.13ms
Exact sample rate is: 1008000.009613 Hz
Sampling at 1008000 S/s.
Output at 48000 Hz.
!AIVDM,1,1,,A,139`g0PP0s0Fwh@MmQmL>?vP`8:R,0*07
!AIVDM,1,1,,A,13aEPdPP00PG;>HML0l0h0?vb26J<,0*6D
!AIVDM,1,1,,A,33`e=v@P@pPG>sfMlMSu5:<P0000,0*5D
!AIVDM,1,1,,A,33`e=v@P@pPG>ptMlN0M3:8h00nA,0*01
!AIVDM,1,1,,A,13aDCK?P1CPG;SbMlf9E>0w22@Ca,0*56
!AIVDM,1,1,,A,33`e=v@P@qPG>o6MlNAdv:6r010@,0*0A
!AIVDM,1,1,,A,13aEPdPP00PG;>LMl0kh0?w@2<19,0*39
!AIVDM,1,1,,A,13aDrc@P1@PG8uTmlrJU>gwB26J<,0*2D
!AIVDM,1,1,,A,33`e=v@P@oPG>f6MlOG<`b1T00ri,0*47
!AIVDM,1,1,,A,33`e=v@00qPG>cDml0bLcqwh00s0,0*59
!AIVDM,1,1,,A,139`g0PP0q0FwGJMmSQ<<gwh`6J<,0*1D
!AIVDM,1,1,,A,13aGE0P12PG?IrMlF5=wgWl2HR0,0*50
!AIVDM,1,1,,A,13aEPdPP00PG;>HML0l00?v220Re,0*22
```

```
rtl_fm -g 9999 -f 162227000 -s 1008000 -A 10 -x 1 -y 1 -z 1 -c 192.168.2.2 -p 1234 -a file -c mon -d -f /dev/stdin
```



Getting serious - v3.



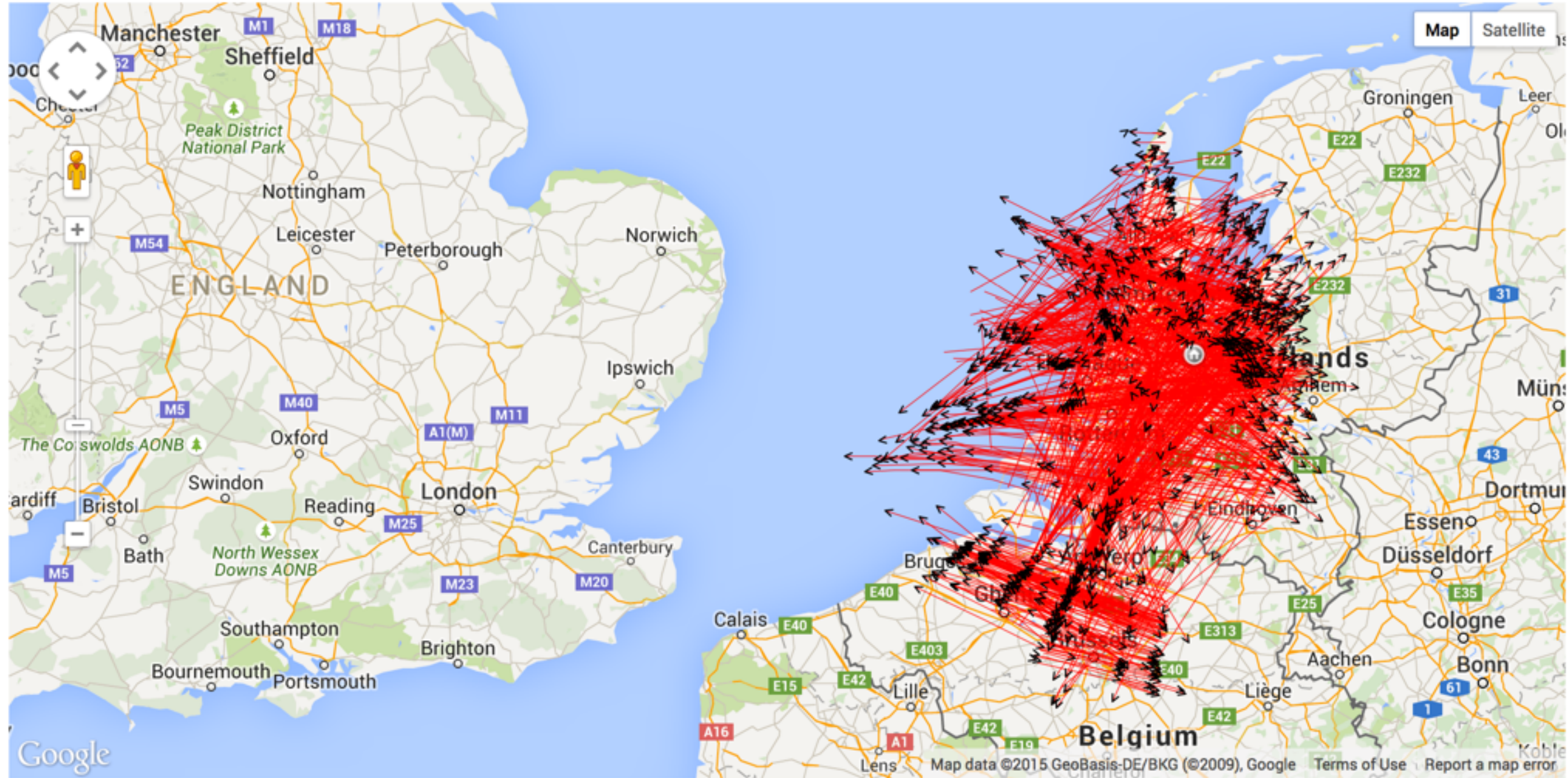
DEMO



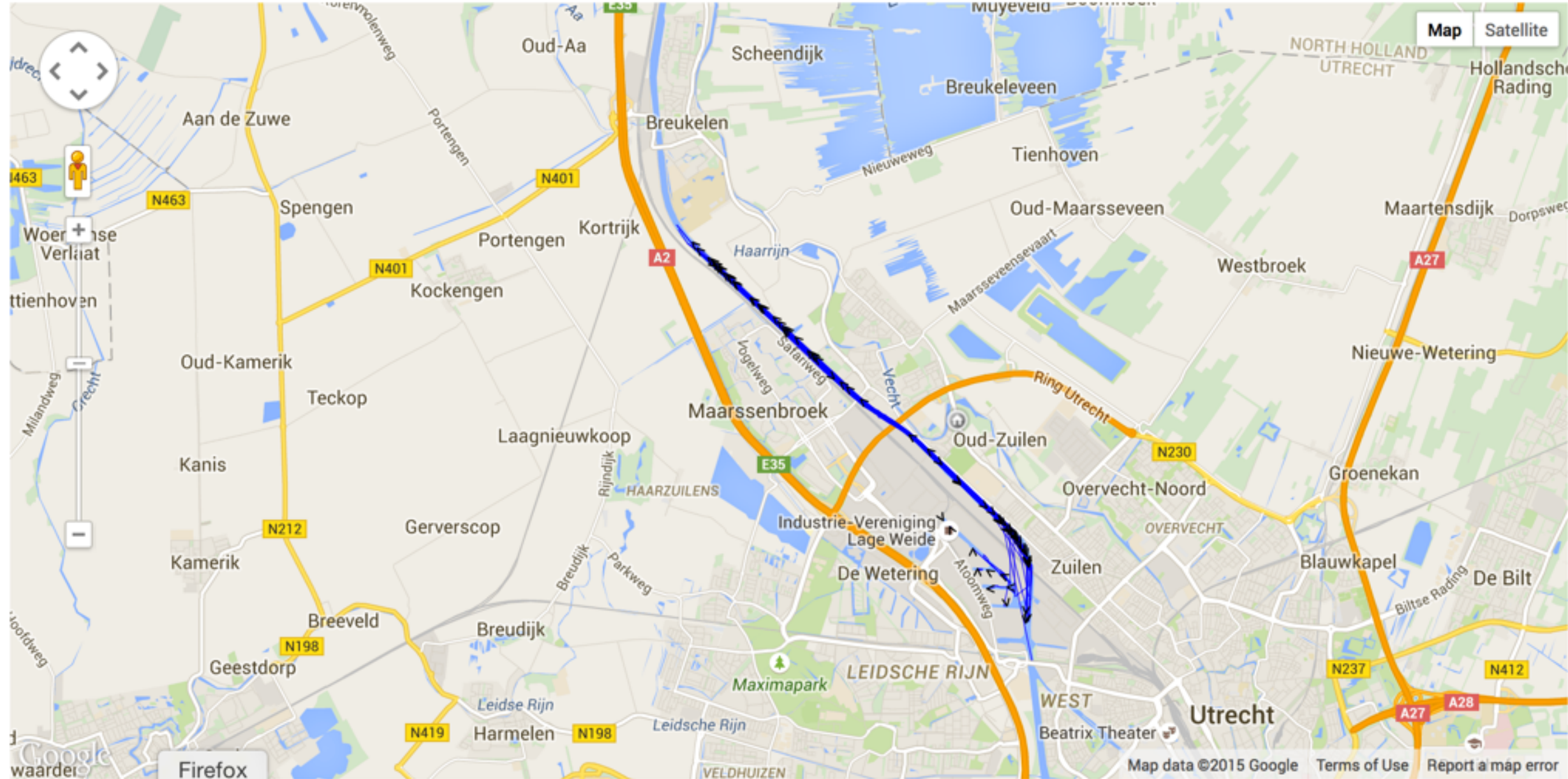
We love data

- Keep system running for a while
- Nicely stocked MongoDB
- Now what?
- Visualise it!

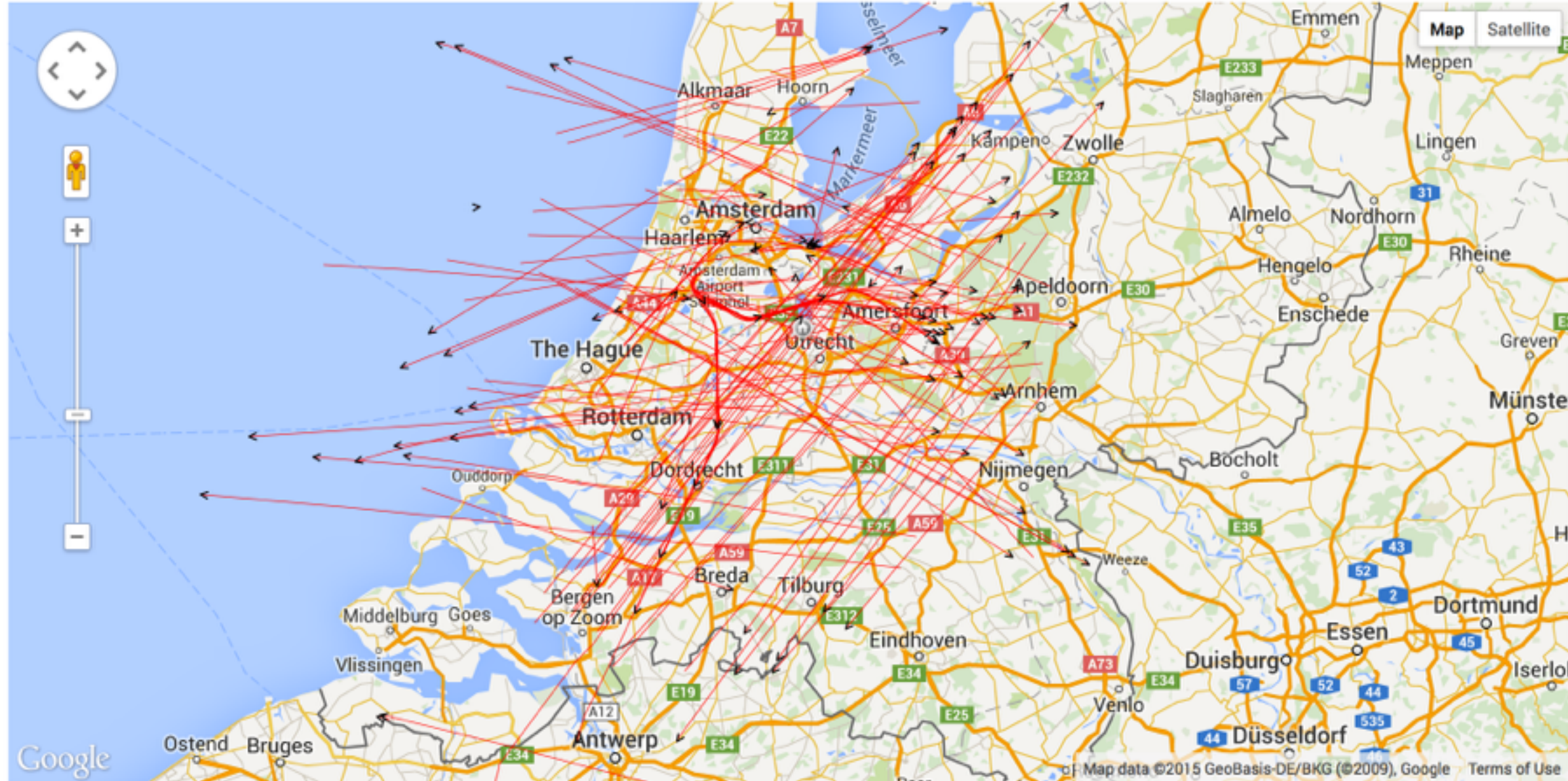
Antenna range: planes



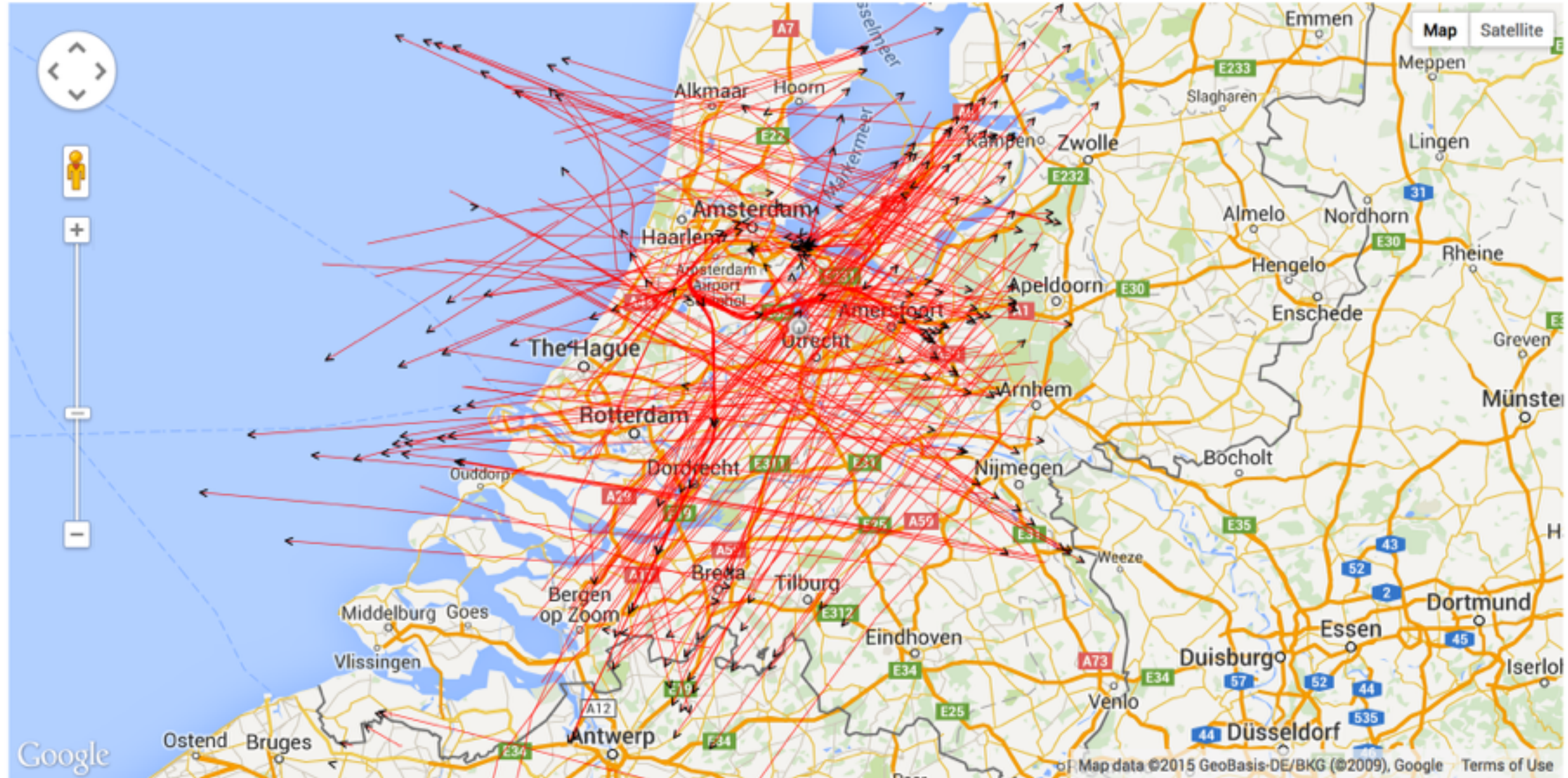
Antenna range: ships



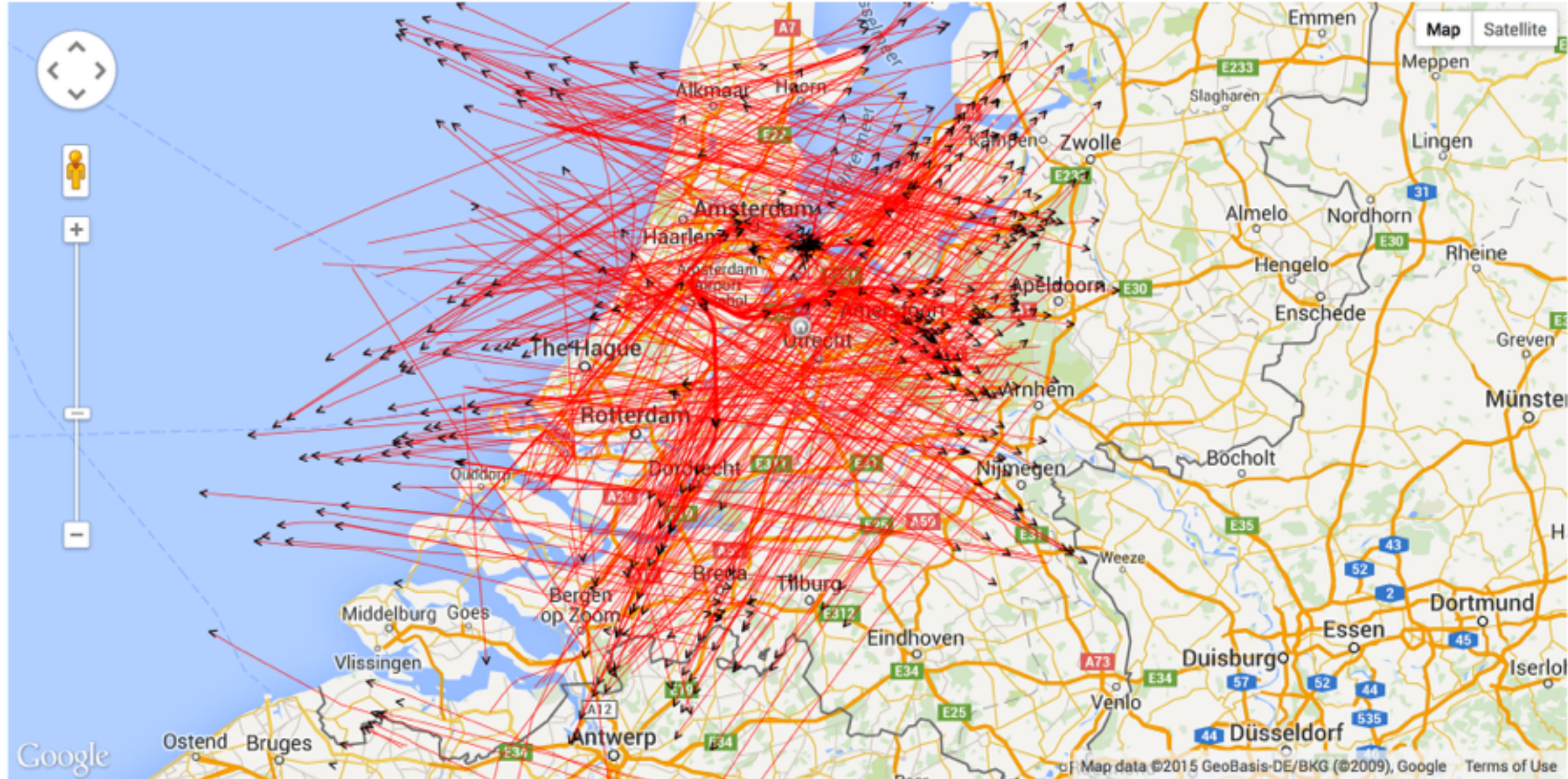
1 hour of data



2 hours



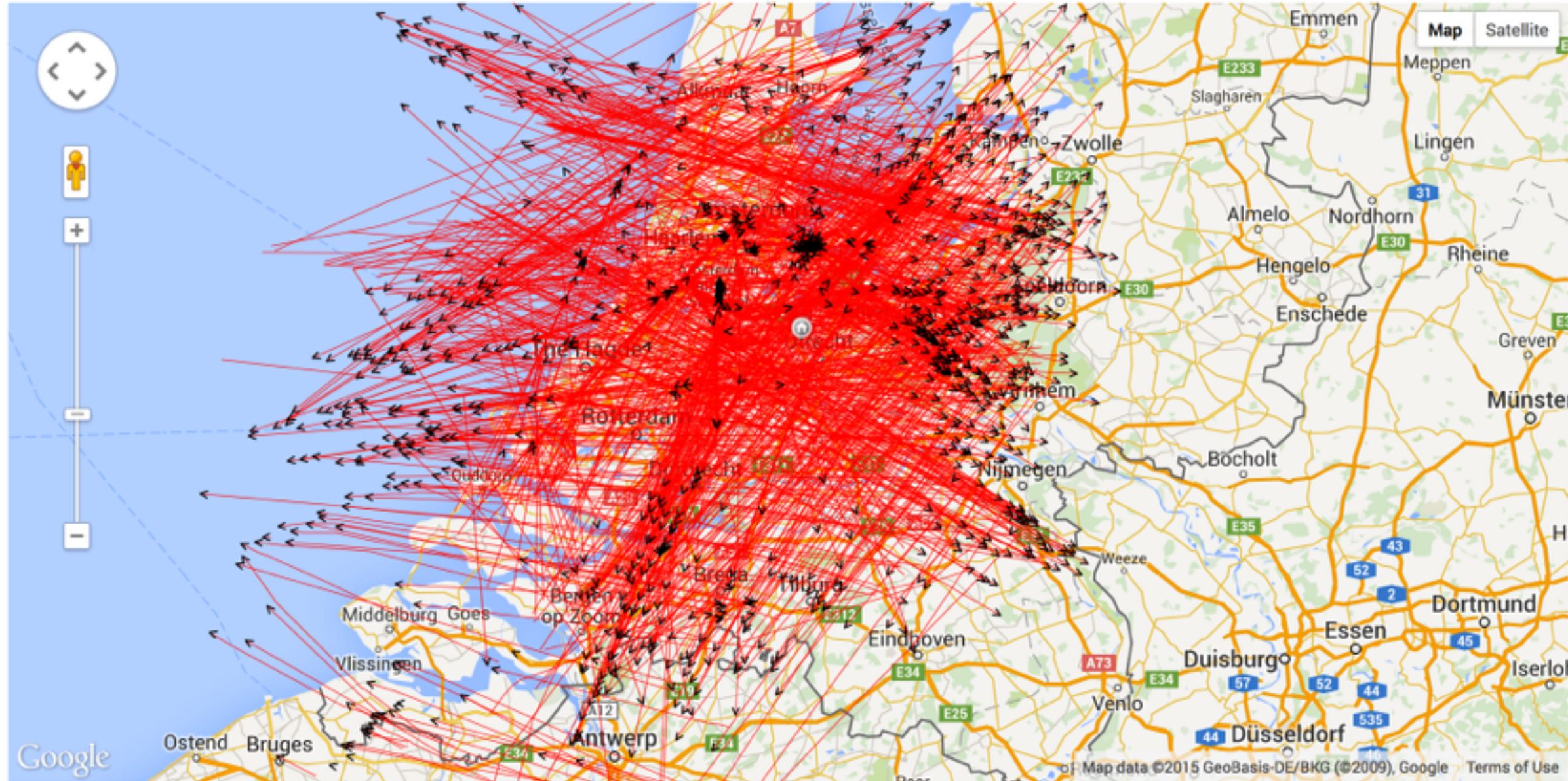
4 hours



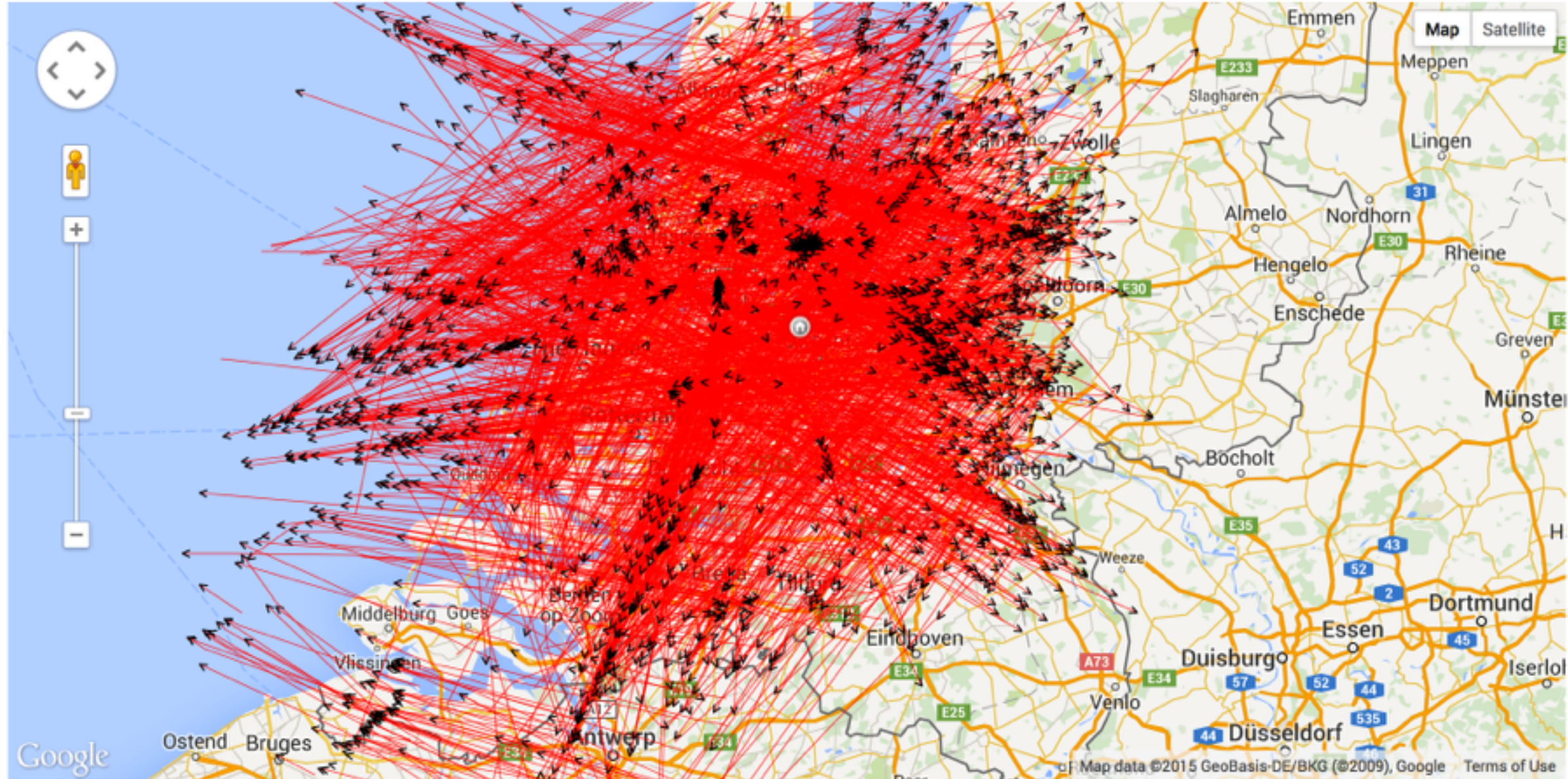
8 hours



16 hours



24 hours



SUMMARY



Summary

- Playing with Software Defined Radio can be loads of fun!
- You can get started for just a few bucks
- Want details?



<https://github.com/bertjan/sdr-java>

- Live demo on <http://sdr.revolution.nl>
- More SDR info on <http://www.rtl-sdr.com>



Q&A



Thanks for your time!

Liked it? Tweet it!

Please care about
conference quality.

