Open Data:
Receive it Yourself

0xA, Marenz, Tassilo (dump@dvb.solutions)

Datenspuren 2022, Dresden

Radio & VDV 420

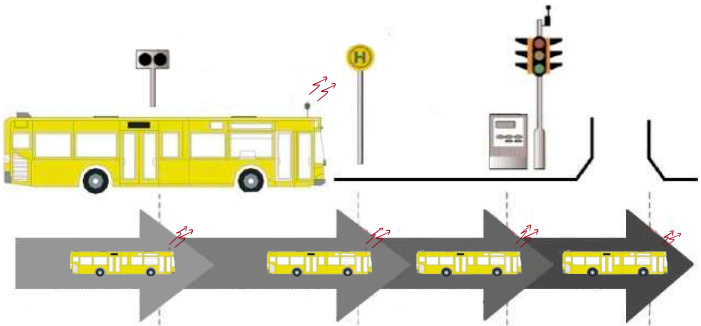


Figure 1: Traffic light controlled by radio link of busses and trams. Modified graphic from urbic

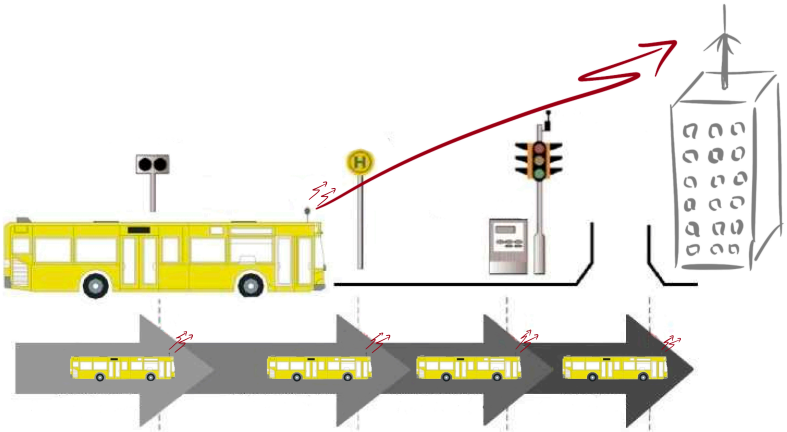


Figure 2: Radio signals can be received by our antennas

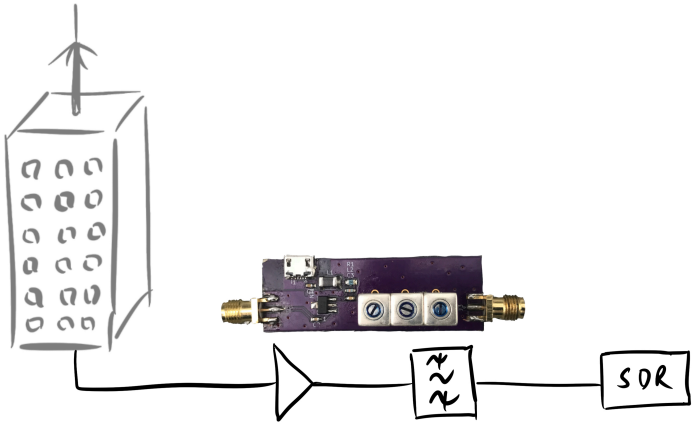


Figure 3: Schematic overview of the receiver hardware

- R09.1x telegrams of trams and busses standardized in VDV 420
- R09.18 is specified not in the standard, but there is a wireshark dissector for it

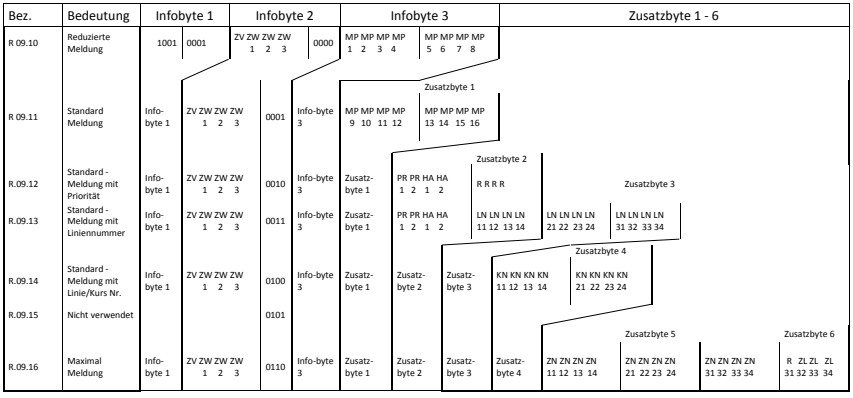


Figure 4: VDV 420 specifies R09.1x types containing different amount of data

What data do we receive?

- Tram identification: line number (3 decimal digits), run number (2 decimal digits), destination number (3 digits)
- Location: Reporting point (16 bit) which might contain 2 bit registration_type (Pre-registration, Registraion, De-registration, Doors closed), traffic light id and direction
- Other data: Delay (± 7), train length (until now always zero) etc.

- Representation of data might be different from the original standard
- Dresden: Reporting Point $= ((< TRAFFIC\ LIGHT\ ID > * 10 + < DIRECTION >) << 2) \mid < REGISTRATION >$
- Tirol: Reporting Point $= ((< DIRECTION > * 1000 + < TRAFFIC\ LIGHT\ ID >) << 2) \mid < REGISTRATION >$

<16>	10010001	ZV	ZW	0110	MP	MA	PR	HA	LN	KI	KO	FW	0	ZL
<16-4>	10010001	ZV	ZW	0110	MP	MA	PR	HA	LN	FW		ZN	0	ZL
<16-6>	10010001	ZV	ZW	0110	MP	MA	PR	HA	LN	KN		ZN	0	ZL
<16-5>	10010001	ZV	ZW	0110	MP		PR	HA	LN	KN		ZN	0	ZL
VDV R09.16	10010001	ZV	ZW	0110	MP		PR	HA	LN	KN		ZN	0	ZL

Figure 5: VDV 426 specifies different formats for the data fields

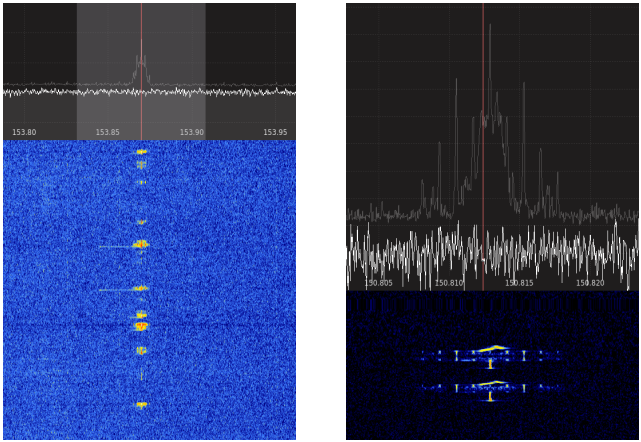


Figure 6: Screenshots of the bursty transmittion pattern

- Bursty transmittion
- Minimum Shift Keying with 2400 Baud

Figure 7: Telegram receiver on the right

What are the frequency ranges to look for the signal?

- Technical documentation of different receivers, i.e. RBL-380 or WZ LSA 2-3/G , provide these details
- Frequency 70cm Band 450 MHz – 470 MHz
- Frequency 2m Band 146 MHz – 174 MHz
- Frequency 4m Band 68 MHz – 87.5 MHz
- We have a table of known frequencies
- OSINT: Search for “R09 frequency <city>” or “Ampelbeeinflussung <city>”

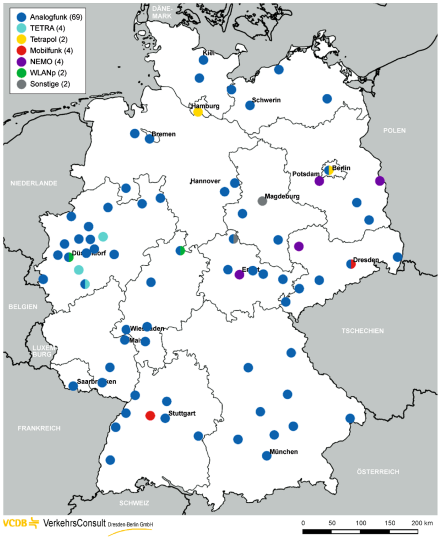


Figure 8: Map from bast displaying selected cities with traffic lights controlled by public transport. We implemented the standard blue points use.

- Different physical layer implementations exist too, i.e. Berlin with Tetrapol
- VDV 426 has more information on this topic
- Encoded data doesn't seem to be different

Mapping

The Problem

Where the Foxtrot Telegram Was Transmitted

- R09Telegram doesn't provide map position data
- location is identified by arbitrary integer reporting_point

```
{
  ...
  "reporting_point":8366,
  "junction":209,
  "direction":1,
  "request_status":2,
  ...
}
```

The Problem

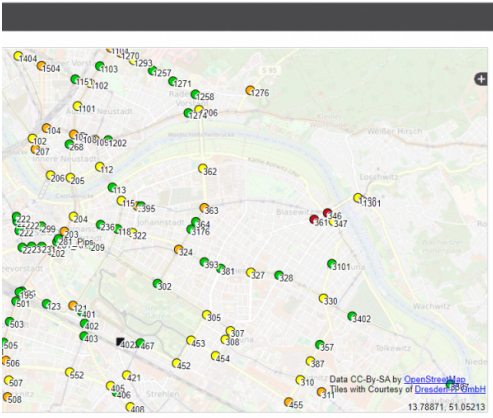
Where the Foxtrot Telegram Was Transmitted

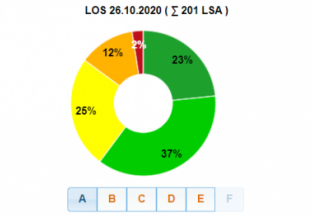
- R09Telegram doesn't provide map position data
- location is identified by arbitrary integer reporting_point
- To the Google we go!

```
{
  ...
  "reporting_point":8366,
  "junction":209,
  "direction":1,
  "request_status":2,
  ...
}
```

OSINT Data

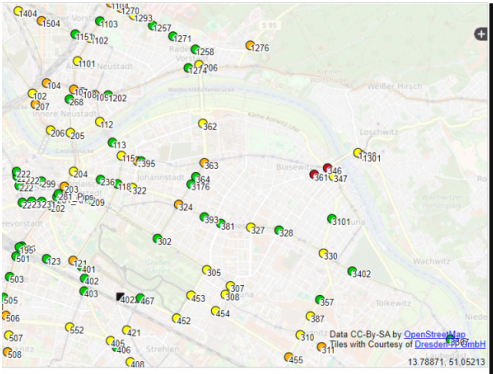
Or Google Your Own LSA ID's





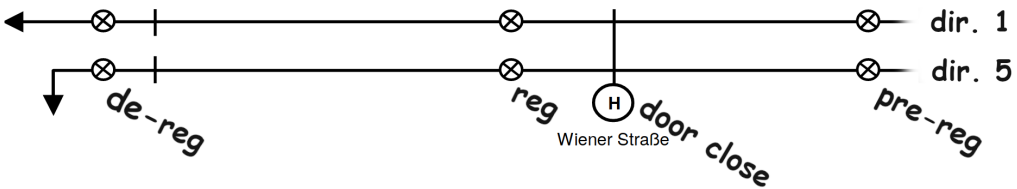
Mapping

JSON Sweatshop, aka Force Your Daughter to Work Day




```
"102": {
    "name": "LSA Anton-/Leipziger Straße",
    "lat": 51.063033,
    "lon": 13.736958
},
"103": {
    "name": "Anton-/Zur Eisenbahnstraße",
    "lat": 51.064067,
    "lon": 13.736935
},
"104": {
    "name": "Schlesischer Platz",
    "lat": 51.064783,
    "lon": 13.740362
},
"107": {
    "name": "Albertplatz",
    "lat": 51.063252,
    "lon": 13.746367
},
"108": {
    "name": "Albertplatz West",
    "lat": 51.064241,
    "lon": 13.744951
},
"109": {
    "name": "Bautzner/Rothenburger",
    "lat": 51.062925,
    "lon": 13.751871
},
"119": {
```

- “by hand” doesn’t scale too well
- osint is unreliable source of information
- junction number corresponds to several reporting points
- need a way to correlate telegrams to map location!



Mapping

Identifying the Vehicle

- Vehicle ID: Line Number & Run Number
- Dresden Trams: Run number present almost always
- Dresden Busses: Hit-or-miss, can be seen on the driver monitor
- Chemnitz: REEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

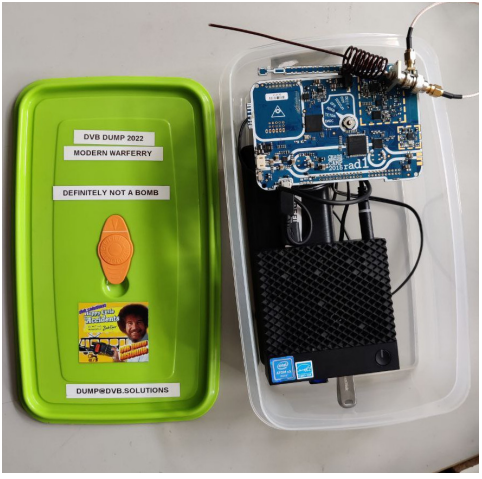
Dump DVB: Modern WarFerry

GPS, Puns and Radio

Telegram Recording:

- SDR with a computer in a tupperware
- Running off a powerbank
- Wartrammer-40k: filters telegrams

Location Recording: Phone
Correlation: lofi

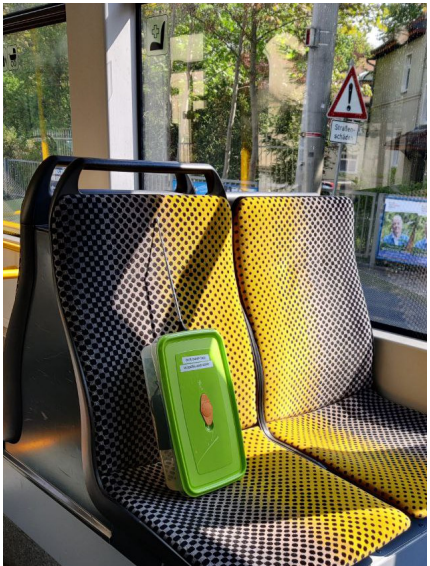


Mapping the City

How It's Done Now

Bootstraping:

- Go around the city with a warferry
- Track your position and line/run number
- Correlate the data



Mapping the City

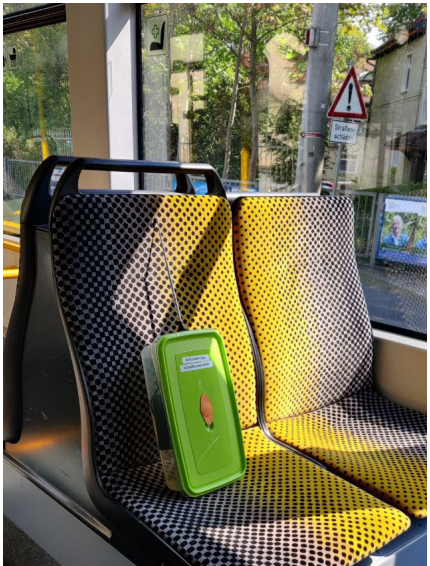
How It's Done Now

Bootstraping:

- Go around the city with a warferry
- Track your position and line/run number
- Correlate the data

Decent city coverage by radio stations:

- Go around the city
- Track your position and line/run number
- Correlate the positions to telegrams from the station



- In some cities there is no easy way to get run numbers $\Rightarrow$ inference from reporting point
- "Go around the city" part sucks
- Nice gps track submission web thing (together with line number tracking)

Architecture & Infra

Receivers in Operation

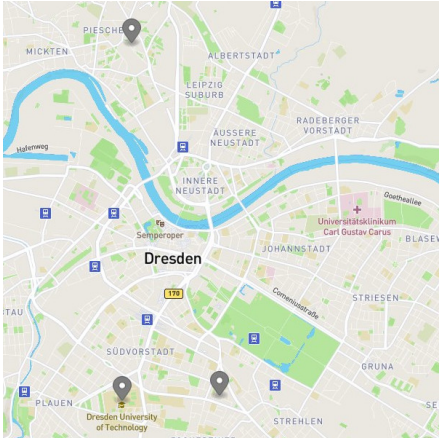


Figure 9: Receivers in Dresden

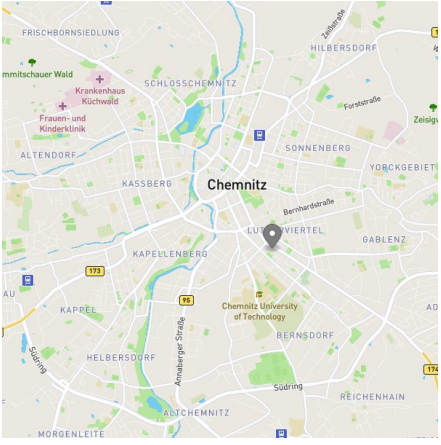
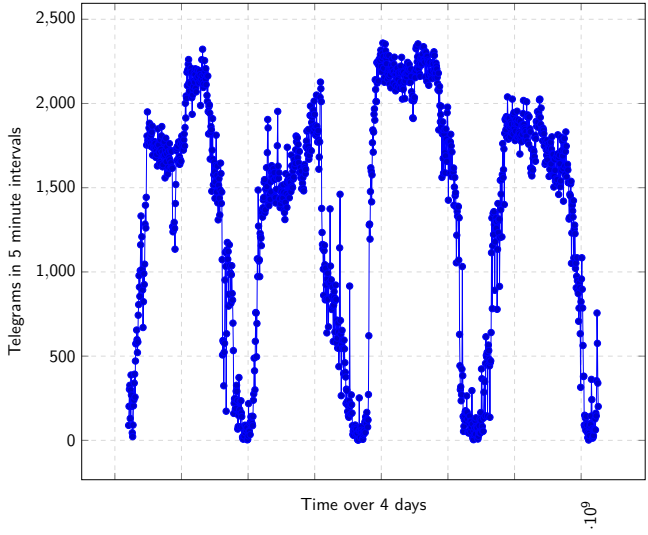


Figure 10: Receivers in Chemnitz

Received Data



What can we see

Figure 11: visible reporting points in the city

LSA-Statistik					
LSA-Typen	Anzahl	mit Infrarot-Datenfunk (IDF)		Verlustzeit	
		absolut	relativ	in s	QSV
Knoten-LSA	243	242	100%	20	C
Fußgänger-LSA	151	145	97%	5	A
Bahnübergänge (BOStrab)	33	33	100%	0	A
Bahnübergänge (SOStrab)	9	9	100%	0	A
Haltlicht-LSA	14	14	100%	0	A
Summe	**450**	**443**	**Ø 98%**	**Ø 13**	**Ø B**

Figure 12: Statistic of traffic lights in dresden from urbic 01/2014

- Unique junction id with more than 300 received telegram is 325
- 325 junctions received in a period of 3 months.
- We can see 70–80% of all reporting points of the city.

Assembeling a Receiver

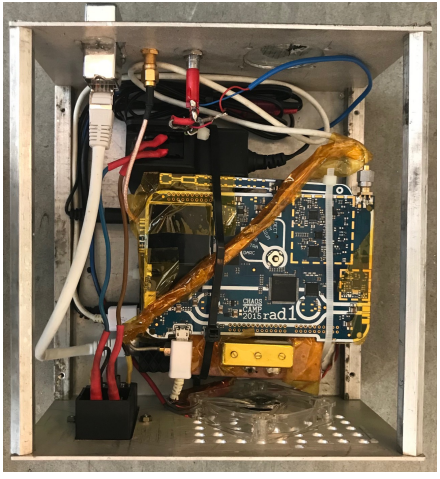
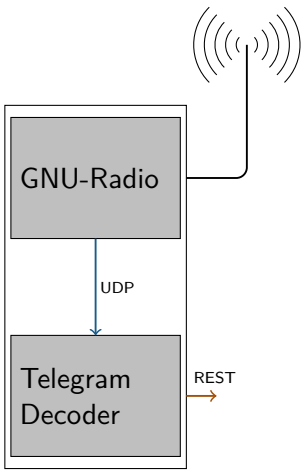


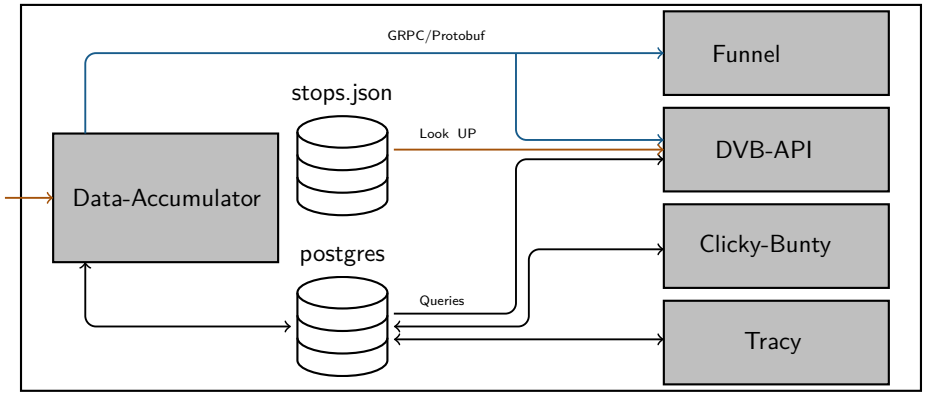
Figure 13: Station Barkhausenbau

- GDR powersuply casing (10€)
- Dell Wyse 3040 (70€)
- Rad1o Badge (e.g RTL SDR 30€)
- Hardware Filter (20€)
- Antenna (15€)
- Miscellaneous items (15€)
- Healthy amounts of kapton

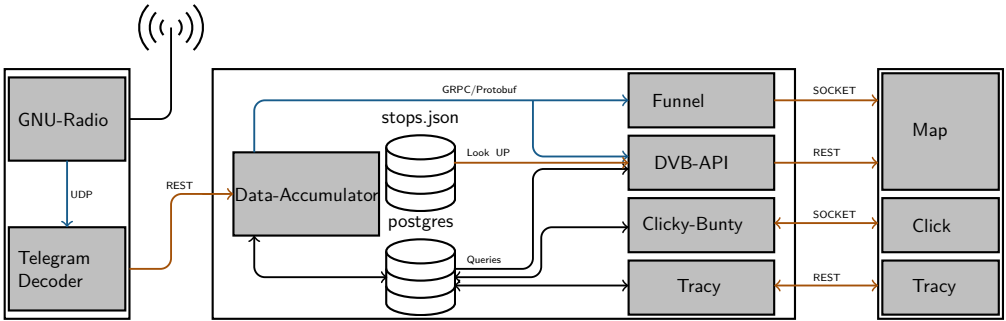
$\Rightarrow$ 160€ per Station



- Region specific encoding depending on quirks from the city
- Currently only parses R09.16 and everything else is recorded as raw-telegram
- Authenticates with station UUID and token


GRPC/Protobuf
Funnel
stops.json
Look UP
DVB-API
Data-Accumulator
Clicky-Bunty
postgres
Queries
Tracy

Architecture



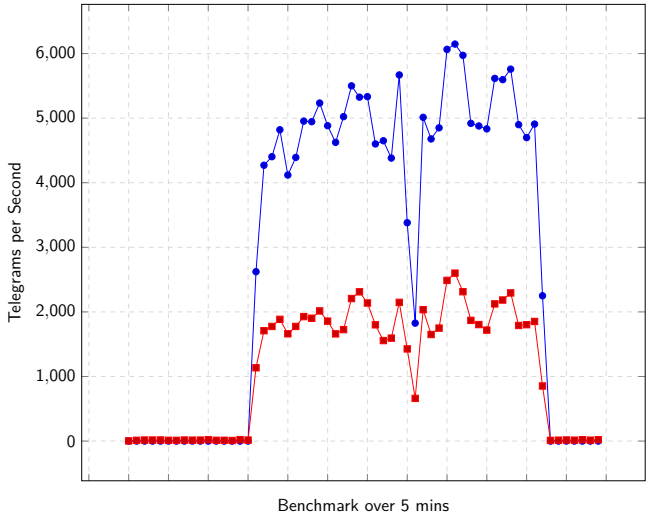
```
dump-dvb.gnuradio = {
  enable = true;
  frequency = 170790000;
  offset = 20000;
  device = "";
  RF = 14;
  IF = 32;
  BB = 42;
};

dump-dvb.telegramDecoder = {
  enable = true;
  server = [
    "https://dump.dvb.solutions"
    "http://dump.staging.dvb.solutions"
  ];
};
```

Figure 14: users receiver configuration

- Building complete Images for x86 and aarch64 on hydra
- custom nixos options for all the services
- wireguard for the receivers that we maintain for punching NAT

Throughput Benchmark



- Influx
  - Database dumps costs 14GB of RAM
  - Server died hourly
- Single point of truth
  - Implement Protocols in Libraries which is then used by both parties.
- Hardware homogenity

Data we Currently Provide

Funnel: Websocket

- 7.7 million telegrams are already recorded
- hourly and daily dumps can be fetched from
  `https://files.dvb.solutions`

Data we Currently Provide

Funnel: Websocket


```
{
  "time":1662932144,
  "station":"97d028ec-43e2-4473-...",
  "region":0,
  "telegram_type":16,
  "reporting_point":8366,
  "junction":209,
  "direction":1,
  "request_status":2,
  "delay":0,
  "priority":0,
  "direction_request":0,
  "line":4,
  "run_number":9,
  "destination_number":31,
  "train_length":0
}
```


- `https://socket.dvb.solutions`
- Configurable Filters (region, line, junction)
- Deduplicated
- Very raw

Data we Currently Provide

stop.json & graph.json

- `https://map.dvb.solutions/stop/<regionid>.json`
- `https://map.dvb.solutions/stop/all.json`
- `https://map.dvb.solutions/graph/<regionid>.json`
- `https://map.dvb.solutions/graph/all.json`

Data we Currently Provide

REST DVB-API

- `https://api.dvb.solutions`
- **GET /vehicles/0/all**
- **POST /vehicles/0/query**
- **POST /network/0/estimated_travel_time**
- **POST /static/0/coordinates**

Call for Action

The Project Is Huge. You Can Help!

- Frontend
  - **Tofu**: landing page (technology open)
  - **Tracy**: Mapping Assistant (technology open)
  - **Click**: User + Station Management (elm-lang)
  - **Windshield** Live Map
- setup stations
- map your city
- find frequencies
- test images on new hardware
- give us roof access

Questions