punctuality plan

Fee Braun
Anna Münster
Markus Arendt
Stefan Keil
Manuel Reich

NCMS
TIGERS
M29
M11
M17

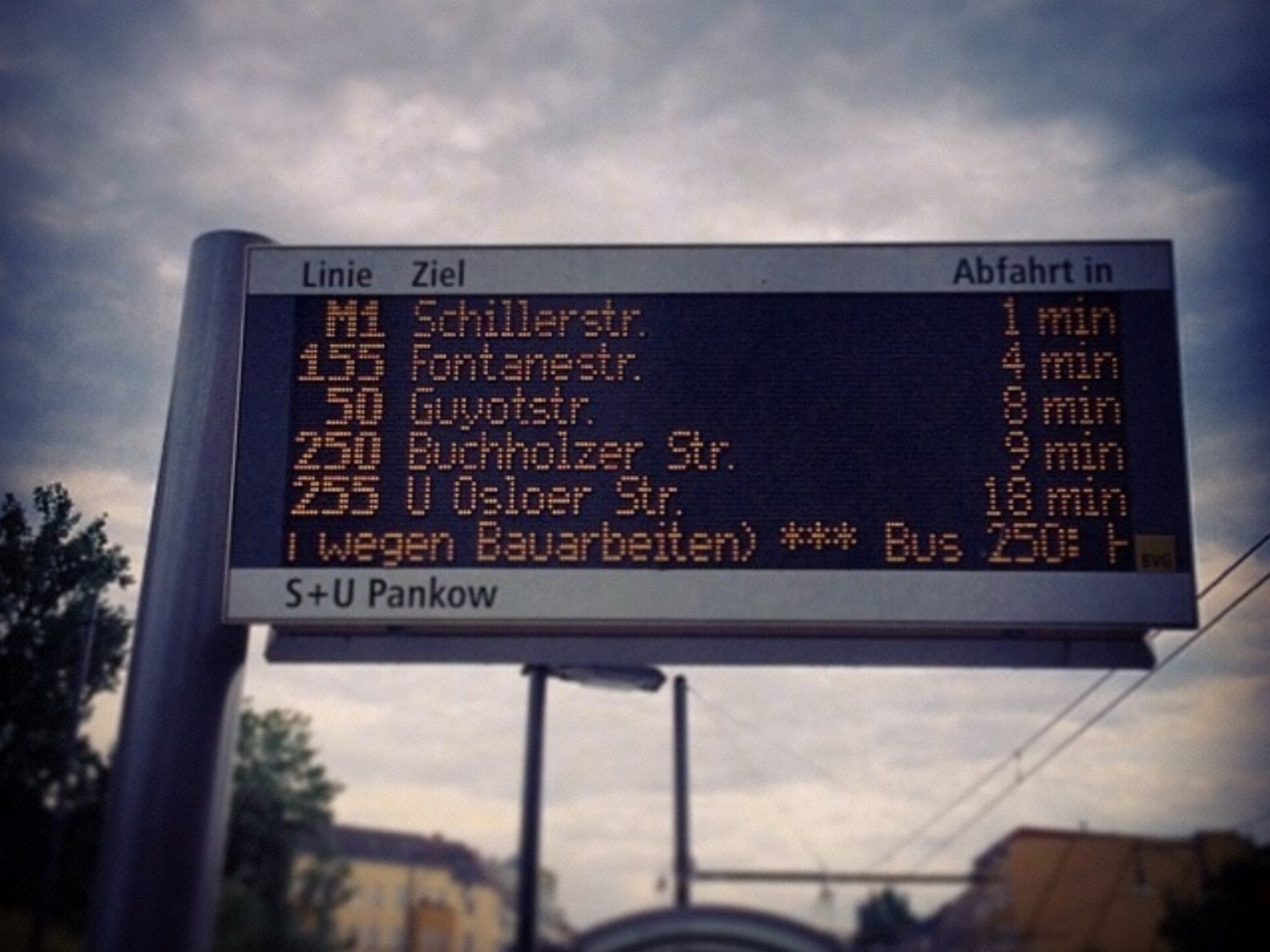
VBB API

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    <xs:element name="ReqC">
        <xs:annotation>
            <xs:documentation>
                The element ReqC is the root element for requests to the HAFAS
                system. It must contain either a location vlidation request, a
                connection request or a connection scroll request. (See the
                corresponding elements for more details).
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice>
                <xs:element ref="ConReq"/>
                <xs:element ref="ConScrReq"/>
            </xs:choice>
            <xs:attributeGroup ref="attlist.ReqC"/>
        </xs:complexType>
    </xs:element>
    <xs:attributeGroup name="attlist.ResC">
        <xs:attribute name="ver" type="xs:string" use="required" fixed="1.1">
            <xs:annotation>
```

Real Time ?

github.com/kr1sp1n/bvg-api

```
{
  "stations" : [{
    "id"          : "309056",
    "departures" : [{
      "time"        : "01:48",
      "line"        : "Tram M10",
      "direction" : "S+U Warschauer Str."
    }]
  }]
}
```



Linie	Ziel	Abfahrt in
M1	Schillerstr.	1 min
155	Fontanestr.	4 min
50	Guyotstr.	8 min
250	Buchholzer Str.	9 min
255	U Osloer Str.	18 min

i wegen Bauarbeiten) *** Bus 250:

S+U Pankow

crawler

```
for each stop do
  json = get ("bvg-api" + stop.name)
  db.insert(json, currentTime)
end
```

crawler

Berlin hat
7.608 Haltestellen:
auf zu einer
Entdeckungstour.

crawler

```
424 Haltestellen

1 Haltestelle   ≈ 4 Datensätze

alle 10 min     = 144 Aufrufe / Tag
144 * 424 * 4   = 244.224 Datensätze / Tag

Nach 3 Monaten = 21.980.160 Datensätze
```

Statistik

21.980.160 Datensätze

Rechenzeit ≈ 1 Sekunde / Datensatz

21.980.160 Sekunden = 366.336 Minuten
366.336 Minuten = 6105,6 Stunden
6105,6 Stunden = 254,4 Tage

fast 1 Jahr

to Thorpes
THAWPIT
Ty-Ph
BISTO
For all MEAT DISHES
LONDON TRANSPORT
64

Nochmal ganz von Anfang

Statistik sofort berechnen

alle 10 Minuten Statistik aktualisieren

für jede Linie 1 Statistik / Tag

≈ 200 Datensätze / Tag

schneller code

```
SELECT
    "stop_times"."id" AS t0_r0,
    "stop_times"."trip_id" AS t0_r1,
    "stop_times"."arrival_time" AS t0_r2,
    "stop_times"."departure_time" AS t0_r3,
    "stop_times"."stop_id" AS t0_r4,
    "stop_times"."stop_sequence" AS t0_r5,
    "stop_times"."pickup_type" AS t0_r6,
    "stop_times"."drop_off_type" AS t0_r7,
    "stop_times"."created_at" AS t0_r8,
    "stop_times"."updated_at" AS t0_r9,
    "stop_times"."stop_headsign" AS t0_r10,
    "stop_times"."shape_dist_traveled" AS t0_r11,
    "stops"."id" AS t1_r0,
    "stops"."stop_id" AS t1_r1,
    "stops"."stop_code" AS t1_r2,
    "stops"."stop_desc" AS t1_r3,
    "stops"."stop_name" AS t1_r4,
    "stops"."stop_lat" AS t1_r5,
    "stops"."stop_lon" AS t1_r6,
    "stops"."zone_id" AS t1_r7,
    "stops"."stop_url" AS t1_r8,
    "stops"."location_type" AS t1_r9,
    "stops"."parent_station" AS t1_r10,
    "stops"."created_at" AS t1_r11,
    "stops"."updated_at" AS t1_r12,
    "trips"."id" AS t2_r0,
    "trips"."route_id" AS t2_r1,
    "trips"."service_id" AS t2_r2,
    "trips"."trip_id" AS t2_r3,
    "trips"."trip_headsign" AS t2_r4,
    "trips"."trip_short_name" AS t2_r5,
    "trips"."direction_id" AS t2_r6,
    "trips"."block_id" AS t2_r7,
    "trips"."shape_id" AS t2_r8,
    "trips"."created_at" AS t2_r9,
    "trips"."updated_at" AS t2_r10,
    "routes"."id" AS t3_r0,
    "routes"."route_id" AS t3_r1,
    "routes"."agency_id" AS t3_r2,
    "routes"."route_short_name" AS t3_r3,
    "routes"."route_long_name" AS t3_r4,
    "routes"."route_desc" AS t3_r5,
    "routes"."route_type" AS t3_r6,
    "routes"."route_url" AS t3_r7,
    "routes"."route_color" AS t3_r8,
    "routes"."route_text_color" AS t3_r9,
    "routes"."created_at" AS t3_r10,
    "routes"."updated_at" AS t3_r11,
    departure_time
FROM
    "stop_times"
INNER JOIN "trips" ON "trips"."trip_id" = "stop_times"."trip_id"
LEFT OUTER JOIN "stops" ON "stops"."stop_id" = "stop_times"."stop_id"
LEFT OUTER JOIN "routes" ON "routes"."route_id" = "trips"."route_id"
WHERE ("trips"."service_id" NOT IN
    (SELECT service_id FROM "calendars"
    WHERE ("calendars"."service_id"
    NOT IN
        (SELECT service_id
        FROM "calendar_dates"
        WHERE "calendar_dates"."date" = '20140717'))))
AND ("stop_times"."arrival_time" BETWEEN '16:00:00' AND '16:10:00')
ORDER BY trips.route_id
```

2,5h

schneller code

nur Metro Busse oder Trams

= 27 Linien

nur jede 10. Haltestelle

≈ 300 API Requests

7 min

frontend

daten kommen per “API”

Diagramme mit Chart.js

Klar gehaltene Visualisierung
für schnelle Erkenntnisse

DEMO

puncplan.
canopus.
uberspace.
de