Building an API for real time transit data

Dennis Wilhelm
 Origins

- Stadtwerke Münster GmbH is using *IVU System* to manage their bus network internally
  - Management Tool to plan routes, timetables, etc.
- However:
  - Complex hierarchical XML format for specific software components
  - Not scalable for many users
  - Only available through a VPN tunnel
API

• REST endpoint to query specific information
  > Bus stops, routes, vehicles, etc.
  > Simple (Geo-)JSON format

• Websocket endpoint
  > Realtime push notifications to update vehicle positions

• Publicly available:
  > http://api.busradar.conterra.de/
Building an easy to use and scalable API

- Using the Open API Specification ➔ API complies to existing standards
- Separation of data processing and data delivery
- Periodic data processing using FME Server in the background
  - Access to the original IVU data
  - Format conversion and reformatting
- Data delivery using scalable managed AWS services

More information about FME: [https://conterra.de/fme](https://conterra.de/fme)
Architecture

Realtime Interface

IVU System
Stadtwerke MS

VDV 453, 454
EPON CSV
Bus stops
Bus routes

FME Cloud

Engine Level
FME Engine 1
FME Engine 2
FME Engine ...

FME Cloud

Amazon Services

Geo DB

DynamoDB

Rest Service
Web Socket

Live Public Transport Service

GTFS Real Time
Real Time Feed

Amazon Web Services

Rest Service
Web Socket

External
Internal

Live Public Transport Service

VDV 453, 454
Bus stops
Bus routes

EPON CSV

© con terra GmbH
Netzplan Münster

Google Maps
Final remarks

- The API is ready to use:
  > [http://api.busradar.conterra.de](http://api.busradar.conterra.de)

- Feedback:
  > d.wilhelm@conterra.de