OpenTripPlanner

Past, Present and the Future

Hannes Junnila

FOSDEM 2023 — 4 February 2023

About me

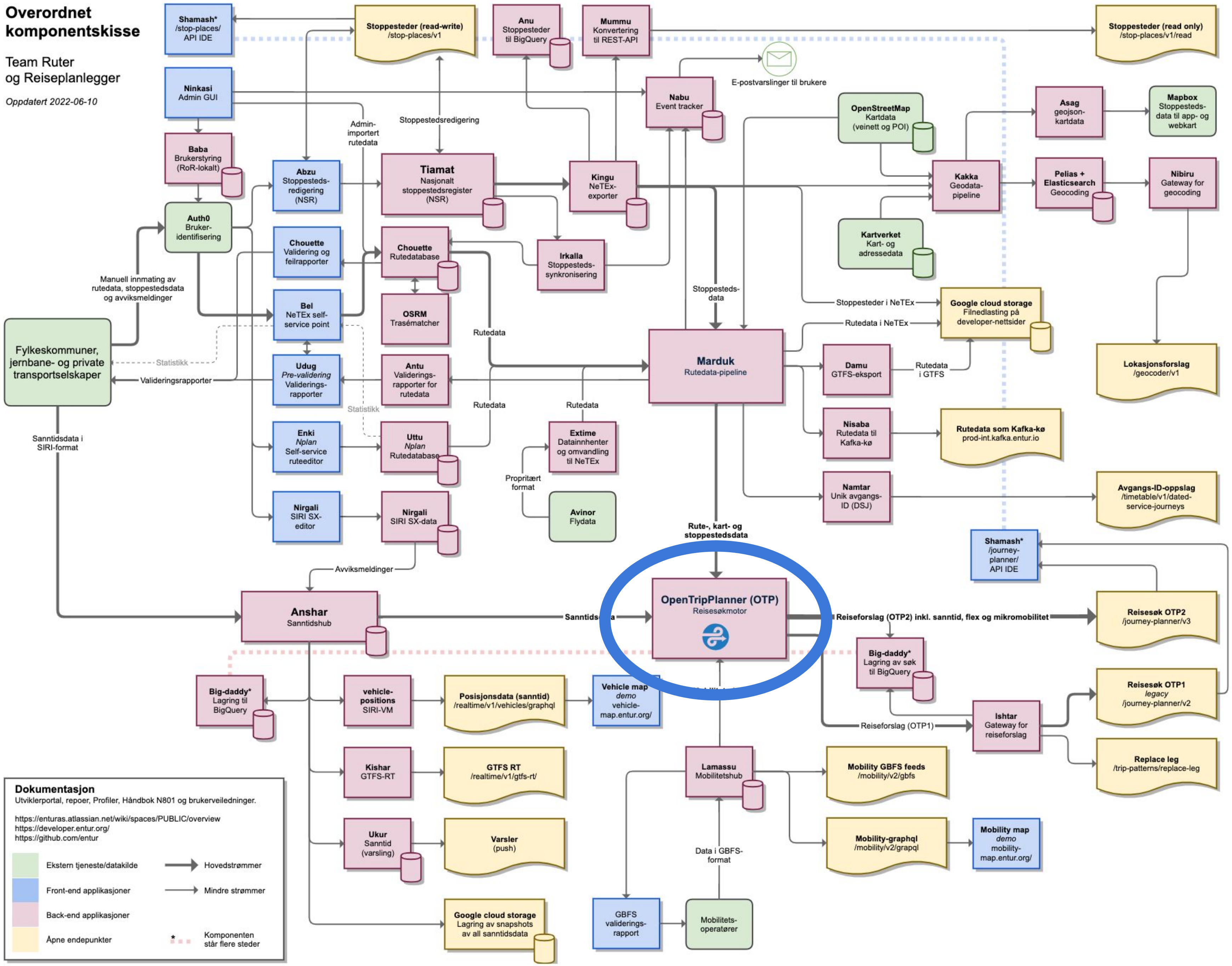
Tinkered with OTP since ~2011

- City of Helsinki 2014
- Helsinki Region Transport & Digitransit 2015–2018
- Kyyti 2020
- Entur 2021–

Overordnet komponentskisse

Team Ruter og Reiseplanlegger

Oppdatert 2022-06-10



Shamash* /stop-places/ API IDE

Stoppesteder (read-write) /stop-places/v1

Anu Stoppesteder til BigQuery

Mummu Konvertering til REST-API

Stoppesteder (read only) /stop-places/v1/read

E-postvarslinger til brukere

Ninkasi Admin GUI

Nabu Event tracker

OpenStreetMap Kartdata (veinett og POI)

Asag geojson-kartdata

Mapbox Stoppesteds-data til app- og webkart

Baba Brukerstyring (RoR-lokalt)

Admin-importert rutedata

Stoppestedsredigering

Abzu Stoppesteds-redigering (NSR)

Tiamat Nasjonalt stoppestedsregister (NSR)

Kingu NeTEx-exporter

Kakka Geodata-pipeline

Pelias + Elasticsearch Geocoding

Nibiru Gateway for geocoding

Auth0 Bruker-identifisering

Chouette Validering og feilrapporter

Chouette Rutedatabase

Irkalla Stoppesteds-synkronisering

Kartverket Kart- og adressedata

Manuell innmating av rutedata, stoppestedsdata og avviksmeldinger

Bel NeTEx self-service point

OSRM Trasématcher

Stoppesteds-data

Stoppesteder i NeTEx

Google cloud storage Filnedlasting på developer-nettsider

Rutedata i NeTEx

Rutedata

Fylkeskommuner, jernbane- og private transportselskaper

Statistikk

Valideringsrapporter

Udug *Pre-validering* Validerings-rapporter

Antu Validerings-rapporter for rutedata

Marduk Rutedata-pipeline

Damu GTFS-eksport

Rutedata i GTFS

Lokasjonsforslag /geocoder/v1

Statistikk

Rutedata

Rutedata

Enki *Nplan* Self-service ruteeditor

Uttu *Nplan* Rutedatabase

Extime Datainnhenter og omvandling til NeTEx

Nisaba Rutedata til Kafka-kø

Rutedata som Kafka-kø prod-int.kafka.entur.io

Sanntidsdata i SIRI-format

Proprietært format

Namtar Unik avgangs-ID (DSJ)

Avgangs-ID-oppslag /timetable/v1/dated-service-journeys

Nirgali SIRI SX-editor

Nirgali SIRI SX-data

Avinor Flydata

Rute-, kart- og stoppestedsdata

Shamash* /journey-planner/ API IDE

Avviksmeldinger

Anshar Sanntidshub

Sanntidsdata

OpenTripPlanner (OTP) Reisesøkmotor

Reiseforslag (OTP2) inkl. sanntid, flex og mikromobilitet

Reisesøk OTP2 /journey-planner/v3

Big-daddy* Lagring av søk til BigQuery

Big-daddy* Lagring til BigQuery

vehicle-positions SIRI-VM

Posisjonsdata (sanntid) /realtime/v1/vehicles/graphql

Vehicle map *demo* vehicle-map.entur.org/

Reisesøk OTP1 *legacy* /journey-planner/v2

Reiseforslag (OTP1)

Ishtar Gateway for reiseforslag

Kishar GTFS-RT

GTFS RT /realtime/v1/gtfs-rt/

Lamassu Mobilitetshub

Mobility GBFS feeds /mobility/v2/gbfs

Replace leg /trip-patterns/replace-leg

Ukur Sanntid (varsling)

Varsler (push)

Mobility-graphql /mobility/v2/grapql

Mobility map *demo* mobility-map.entur.org/

Data i GBFS-format

Google cloud storage Lagring av snapshots av all sanntidsdata

GBFS validerings-rapport

Mobilitets-operatører

Dokumentasjon

Utviklerportal, repoer, Profiler, Håndbok N801 og brukerveiledninger.

https://enturas.atlassian.net/wiki/spaces/PUBLIC/overview
https://developer.entur.org/
https://github.com/entur

🟩	Ekstern tjeneste/datakilde	➔	Hovedstrømmer
🟦	Front-end applikasjoner	→	Mindre strømmer
🟪	Back-end applikasjoner		
🟨	Åpne endepunkter	*	Komponenten står flere steder

Agenda

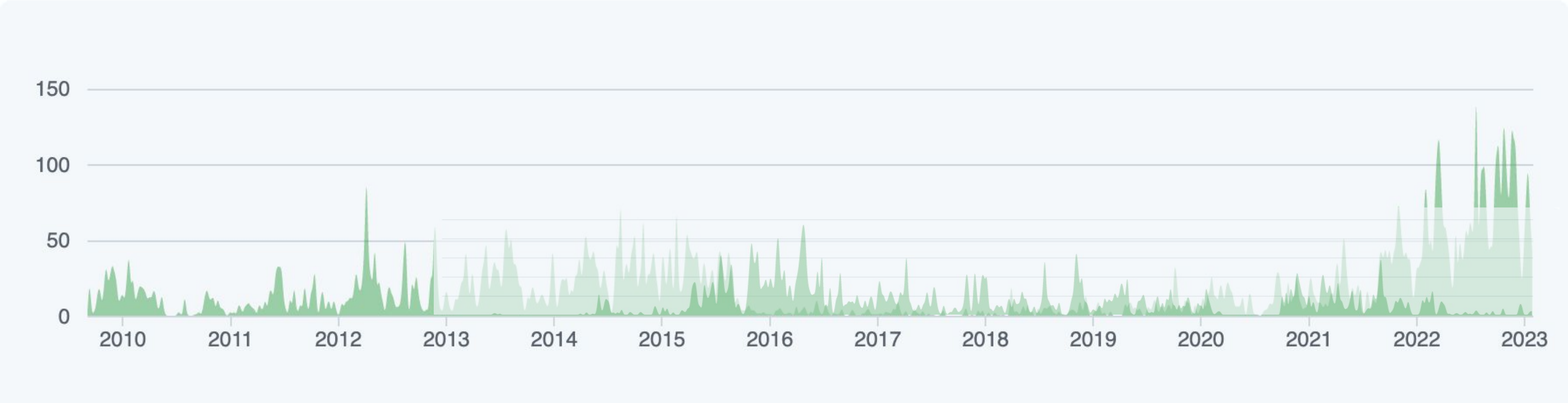
Past — OpenTripPlanner 1

Present — OpenTripPlanner 2

- How it works
- New features
- Sandbox extensions
- Simplified setup

Future — Roadmap

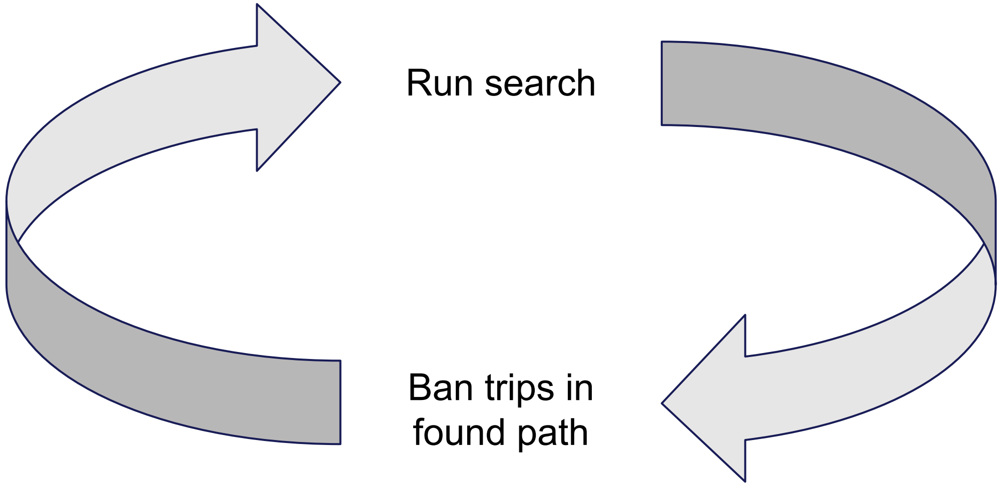
Past


150
100
50
0
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023

Pain points with OTP 1

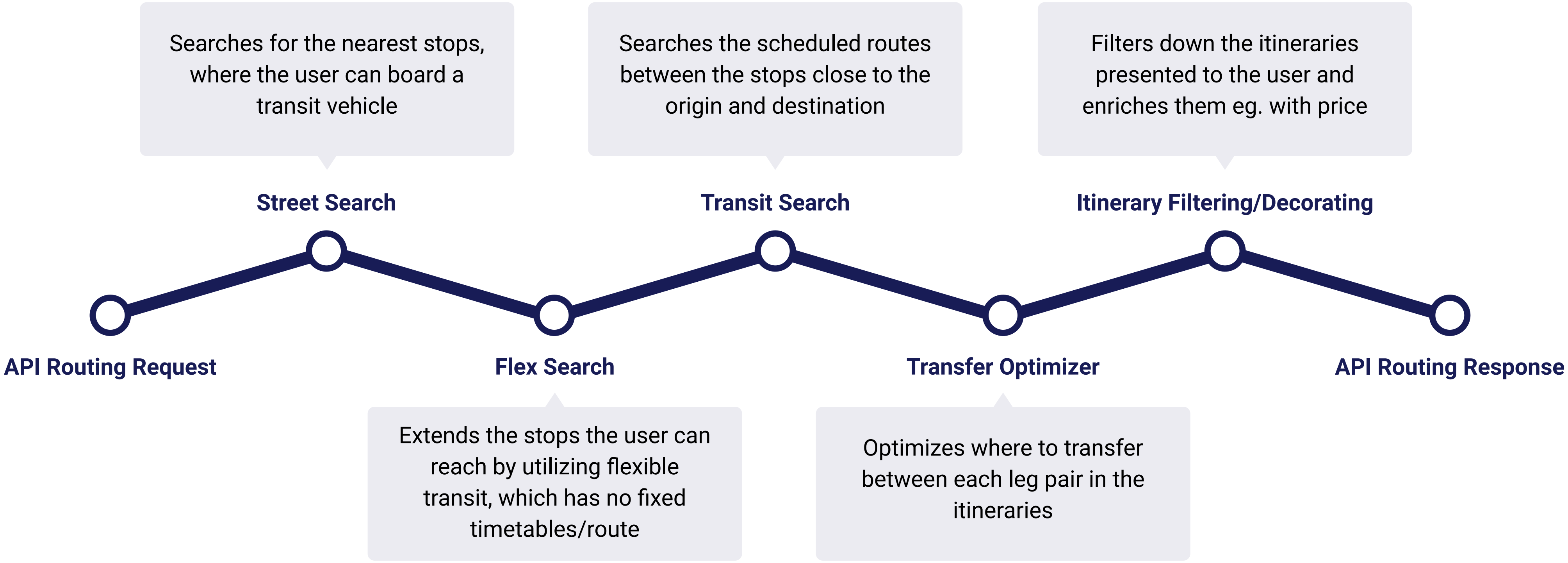
Time-dependent A* search with trip banning

- Insufficient performance for nationwide deployments
- Focus on research capabilities
  - Solved by split into OTP and R5
- Lack of architectural vision and focus
- Fragmented development
  - Each organization had its own fork



Present

OTP Routing Process



API Routing Request

Street Search

Searches for the nearest stops, where the user can board a transit vehicle

Flex Search

Extends the stops the user can reach by utilizing flexible transit, which has no fixed timetables/route

Transit Search

Searches the scheduled routes between the stops close to the origin and destination

Transfer Optimizer

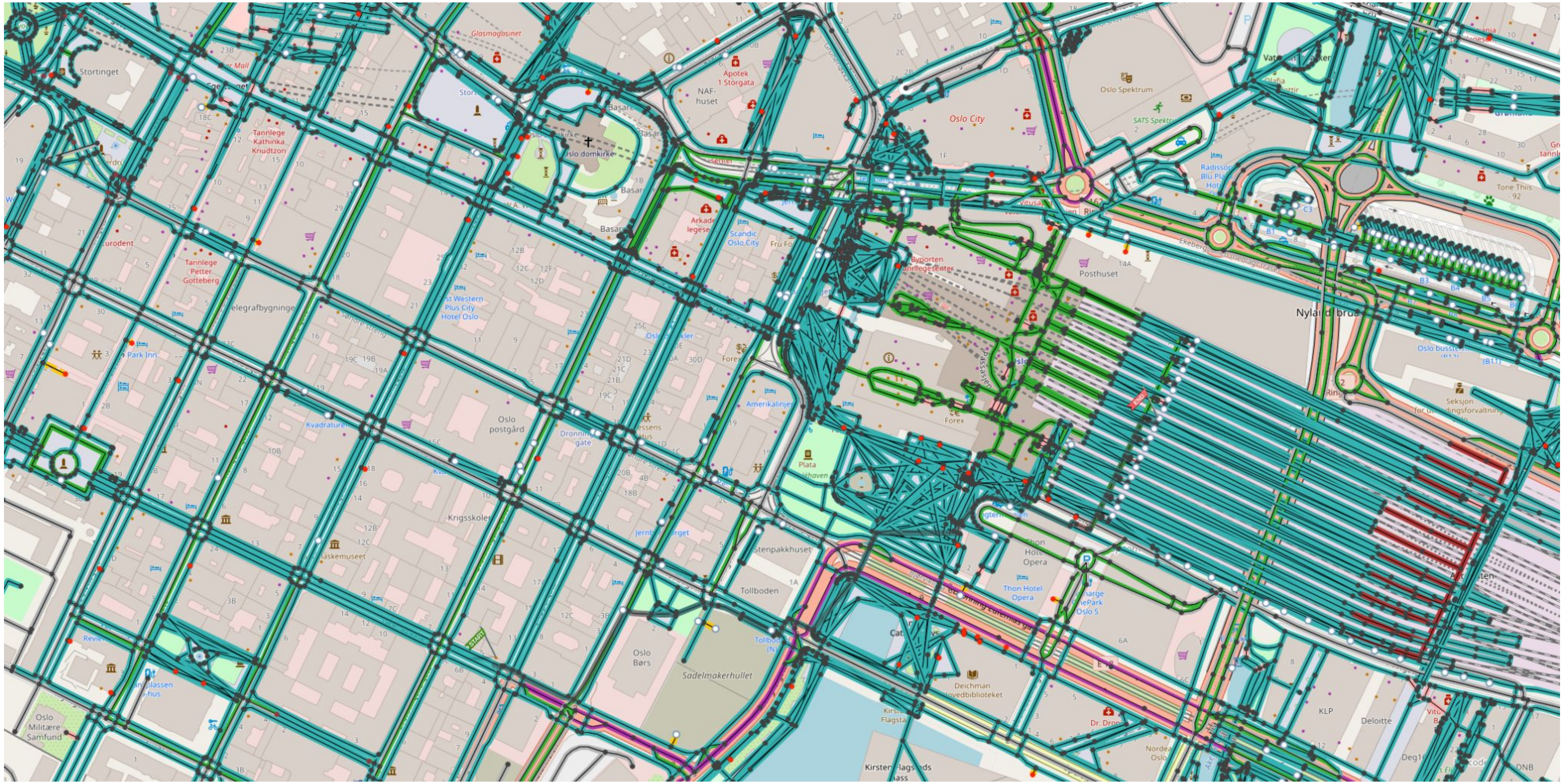
Optimizes where to transfer between each leg pair in the itineraries

Itinerary Filtering/Decorating

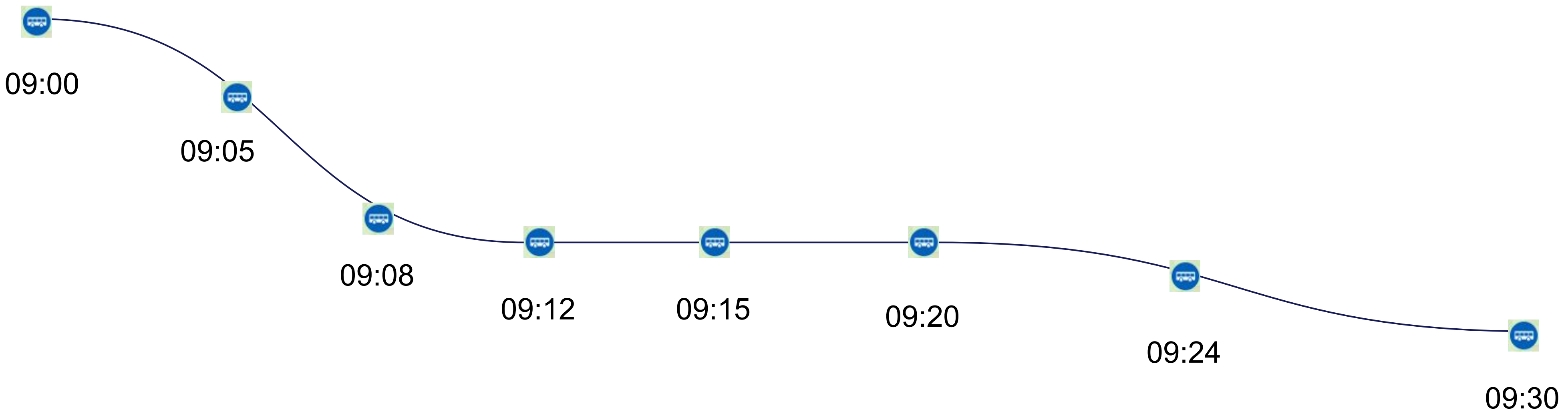
Filters down the itineraries presented to the user and enriches them eg. with price

API Routing Response

Street Search

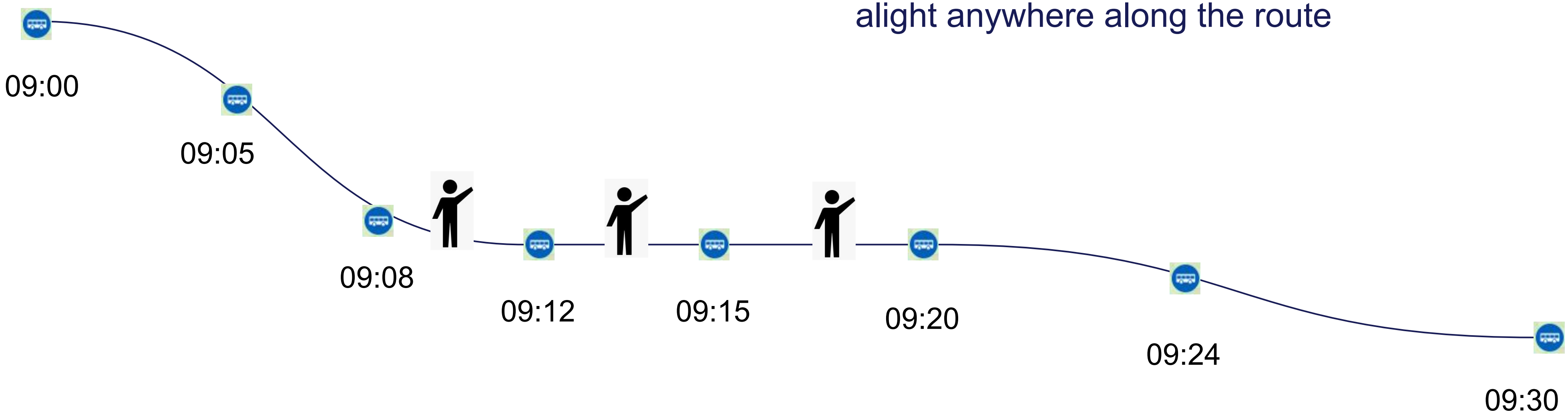


Fixed Route

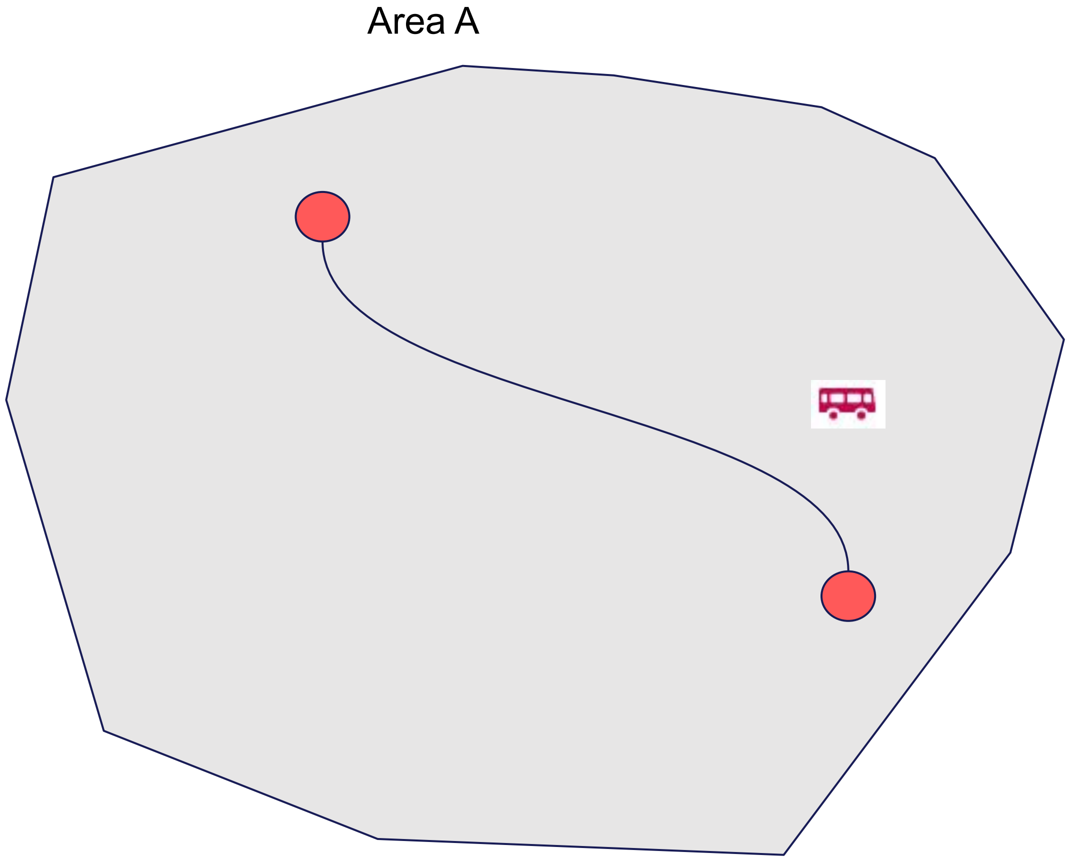


Hail and Ride Sections

- Fixed route and schedule
- Between stops **3** and **6,** you can board or alight anywhere along the route



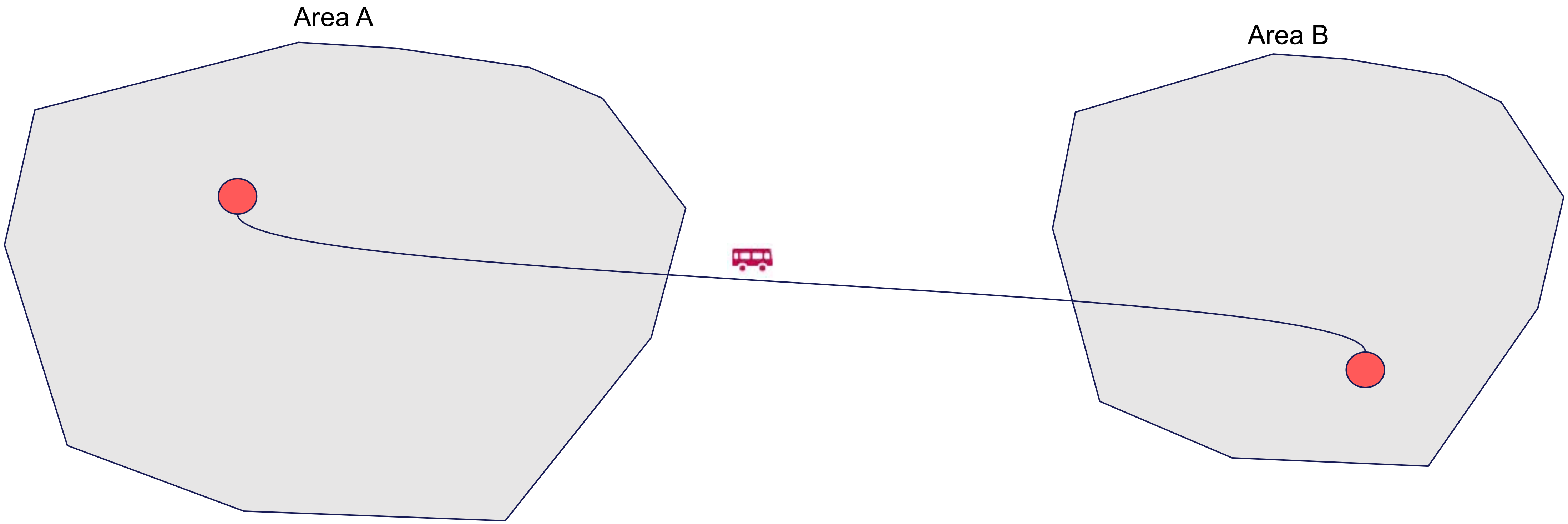
Flexible Areas



- Door to door anywhere within a service area

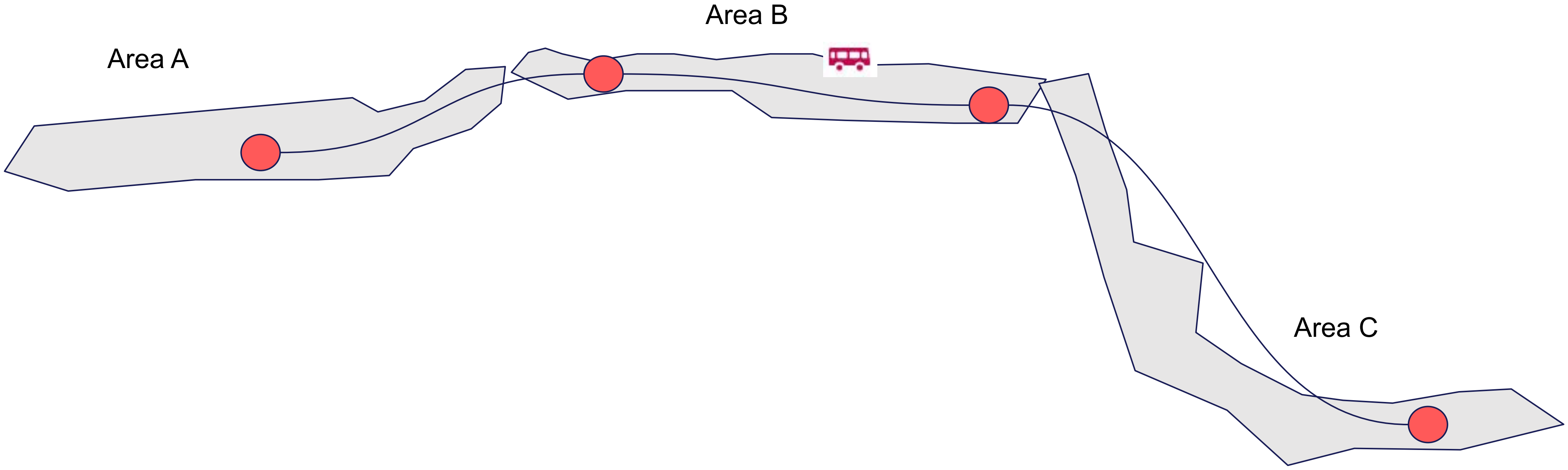
Flexible Areas

- One area for boarding and another for alighting

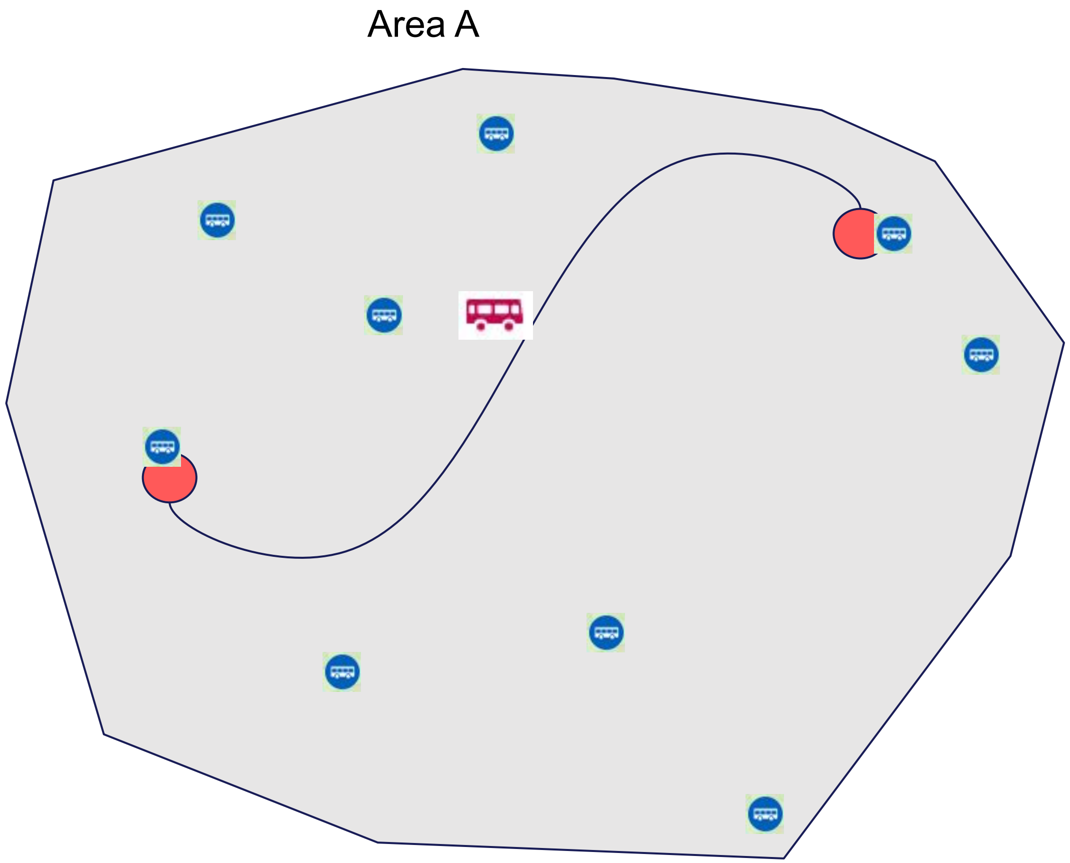


Flexible Areas

- Any number of areas, some with only boarding some with only alighting



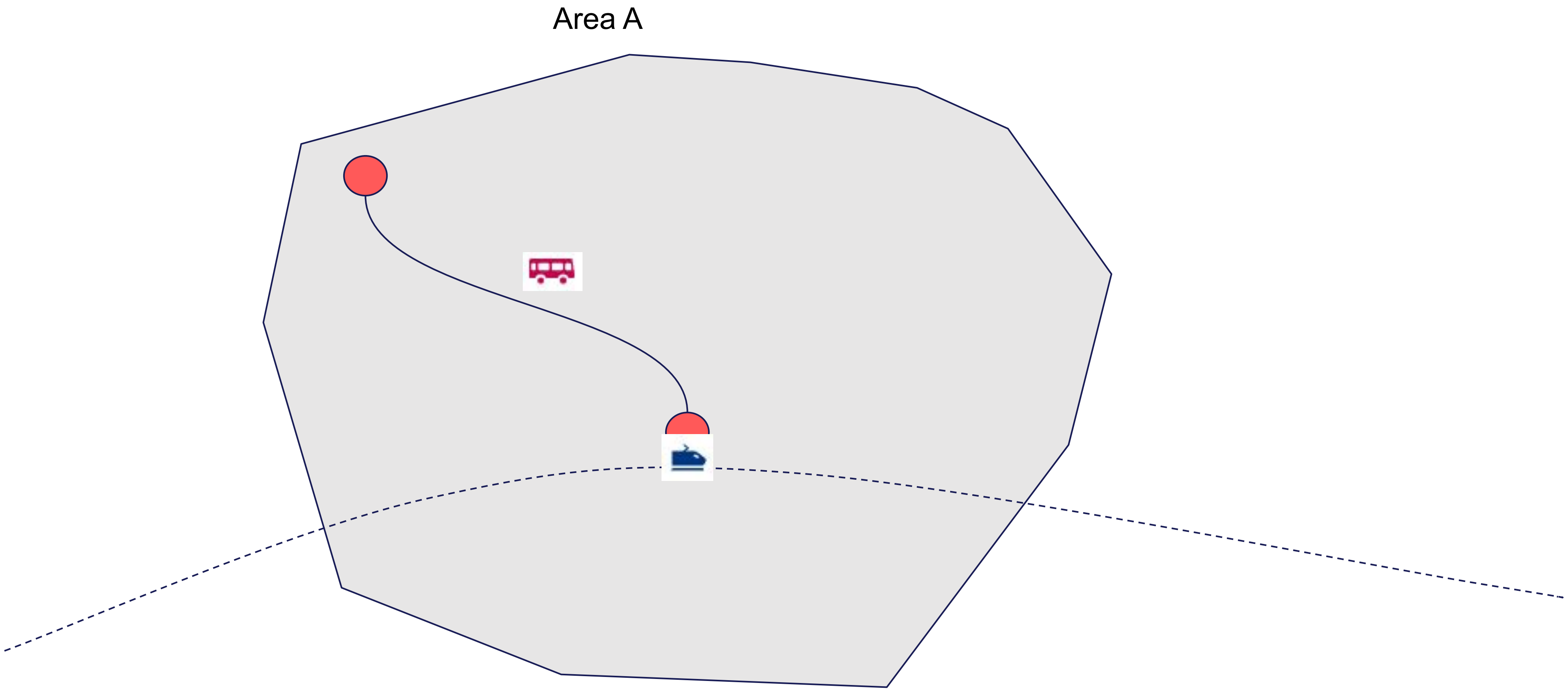
Fixed Stops in an Area



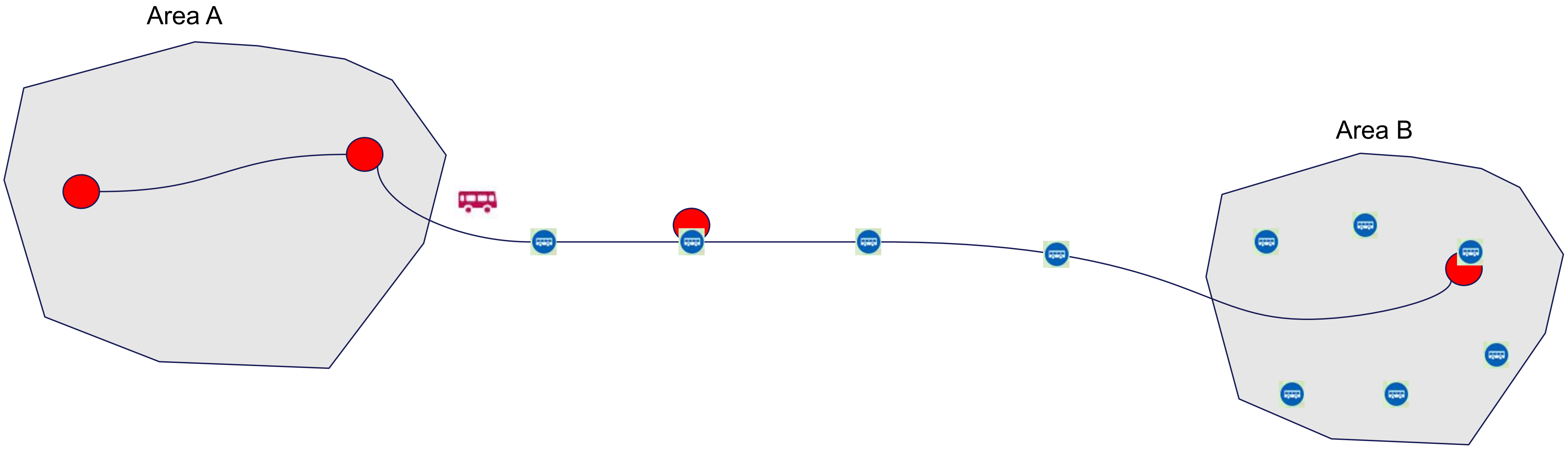
- Stop to stop within an area

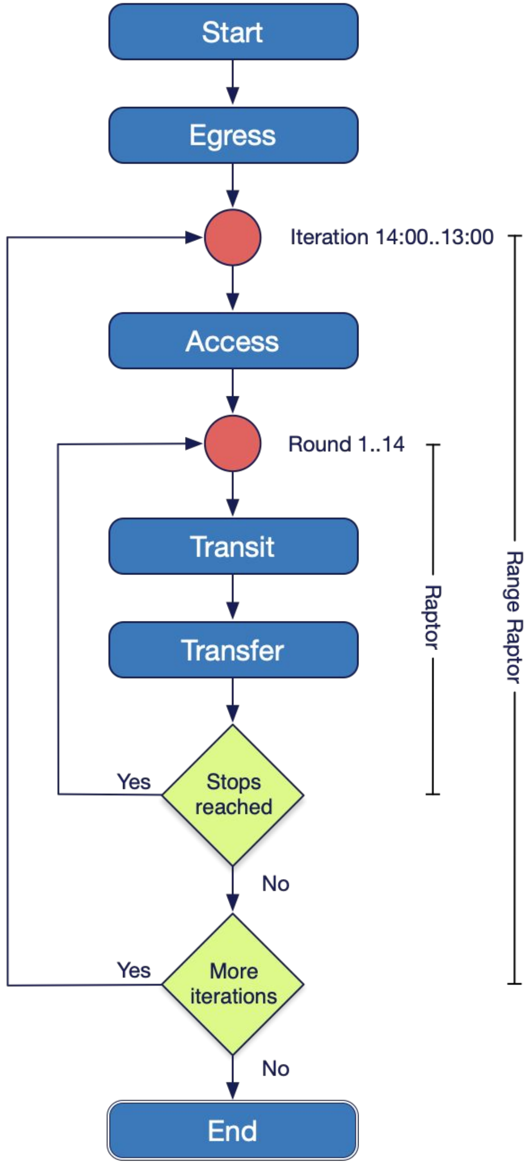
Feeder services

- Area to stop and vice versa



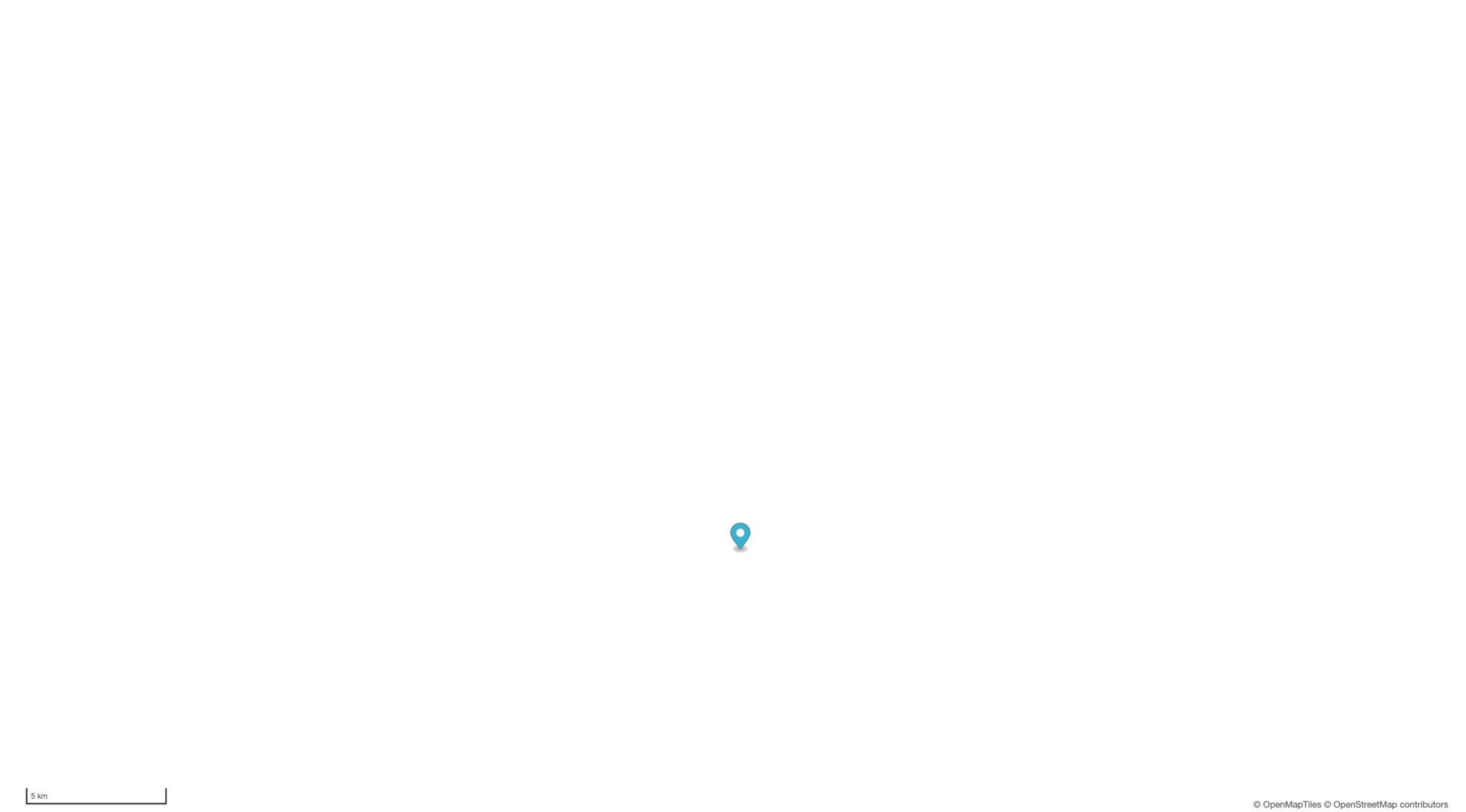
Complex services



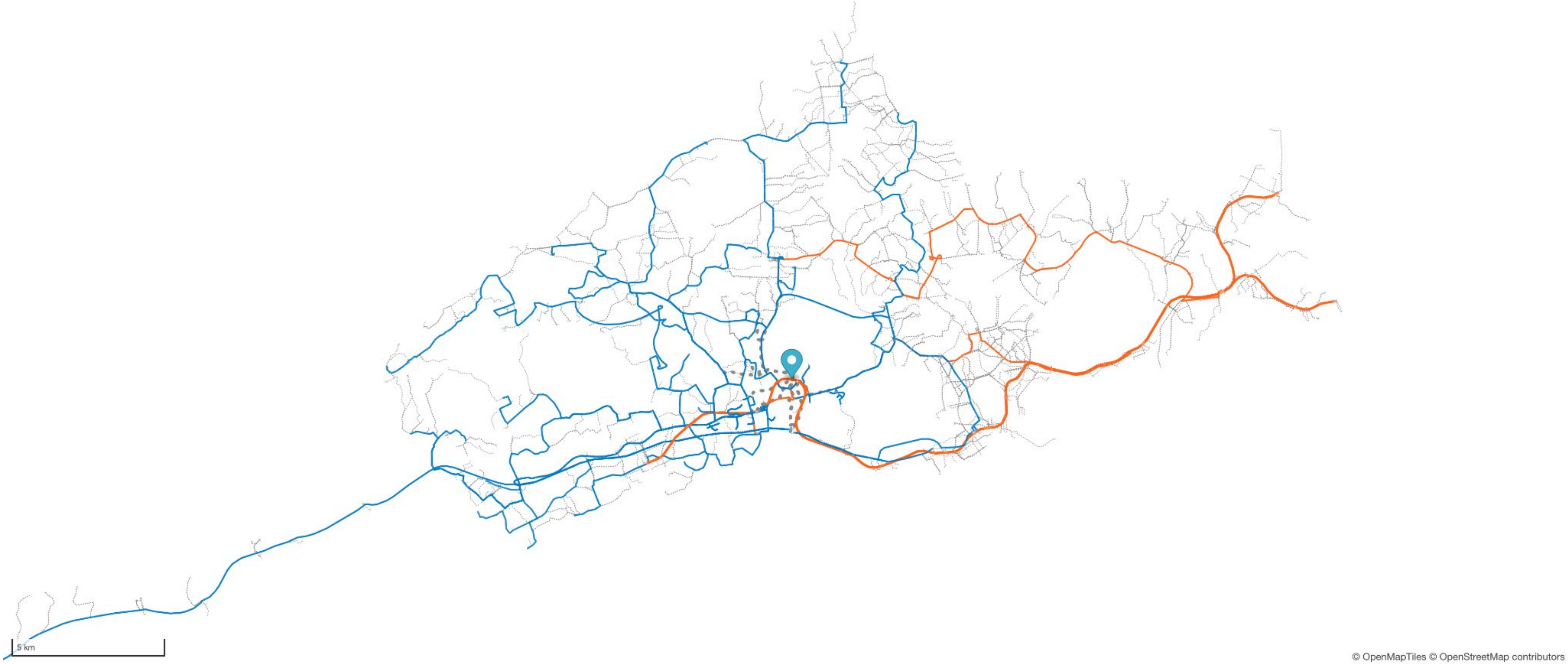


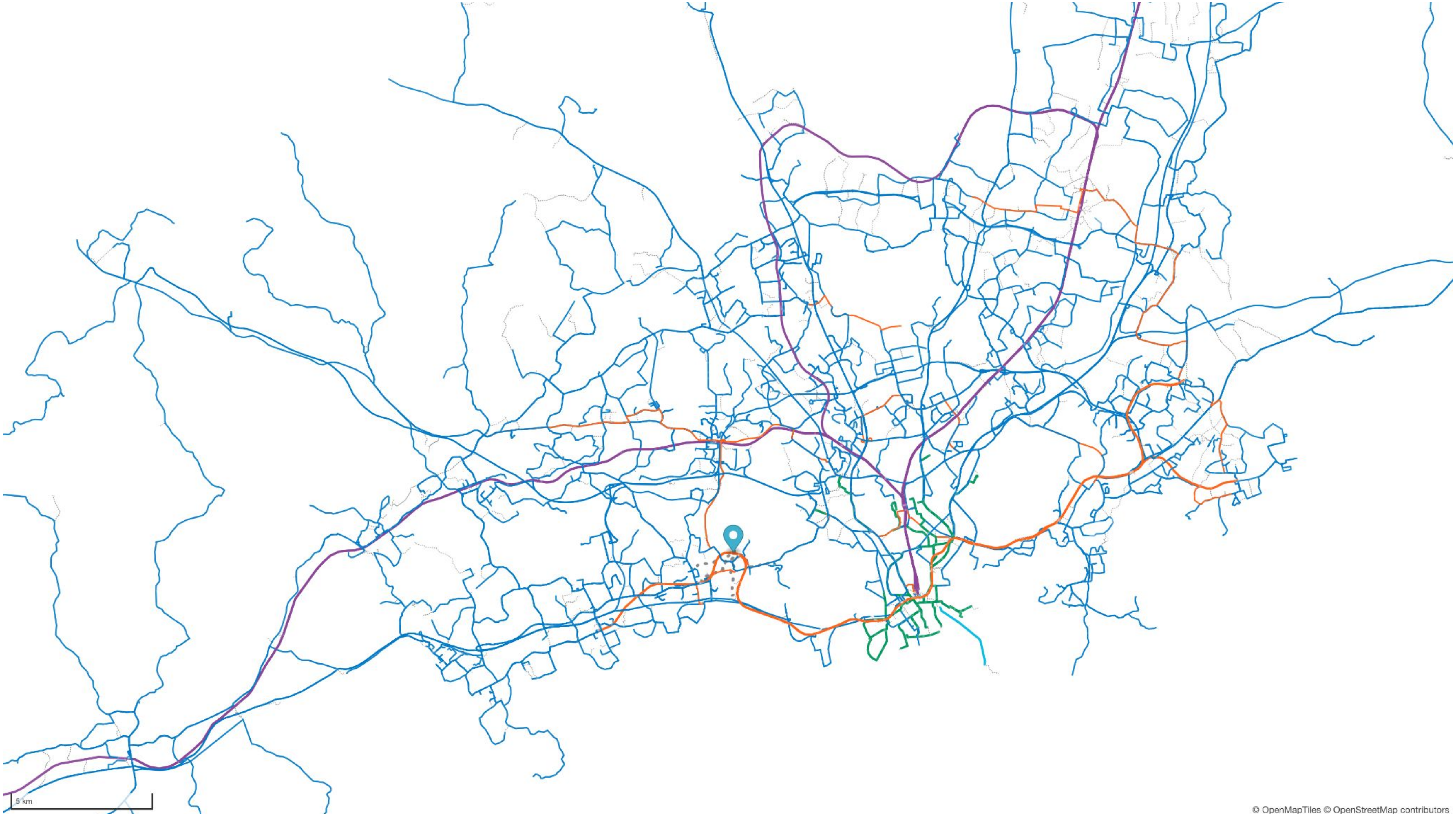
Raptor

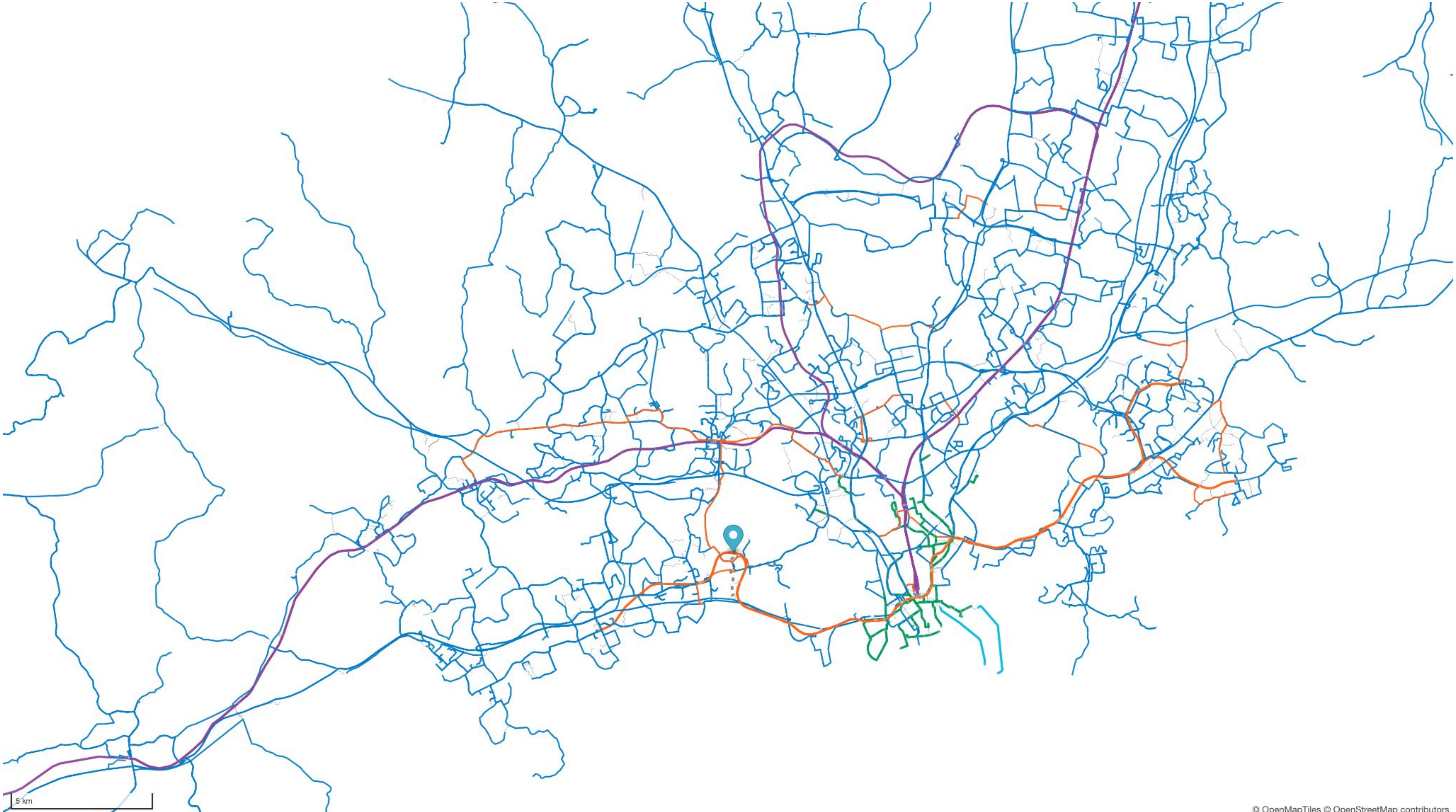
- **Raptor** works in rounds
  - Implicit graph model using memory layout
  - One round for each transit trip & transfer
  - Exploring the transit network following transit routes.
  - Find all pareto optimal paths by
    - [*Arrival time* | *Number of transfers*] – Given *departure time*
- **Range Raptor**
  - Iterates backwards over **departure time** within a **search window**
  - Only explores new trips not reached by previous rounds
  - Pareto optimal by
    - [*Departure time* | *Arrival time* | *Number of transfers*]
- **Multi-criteria Range Raptor**
  - One or more additional criteria – with **performance penalty**
  - Pareto optimal by
    - [*Departure time* | *Arrival time* | *Number of transfers* | *Generalized cost*]



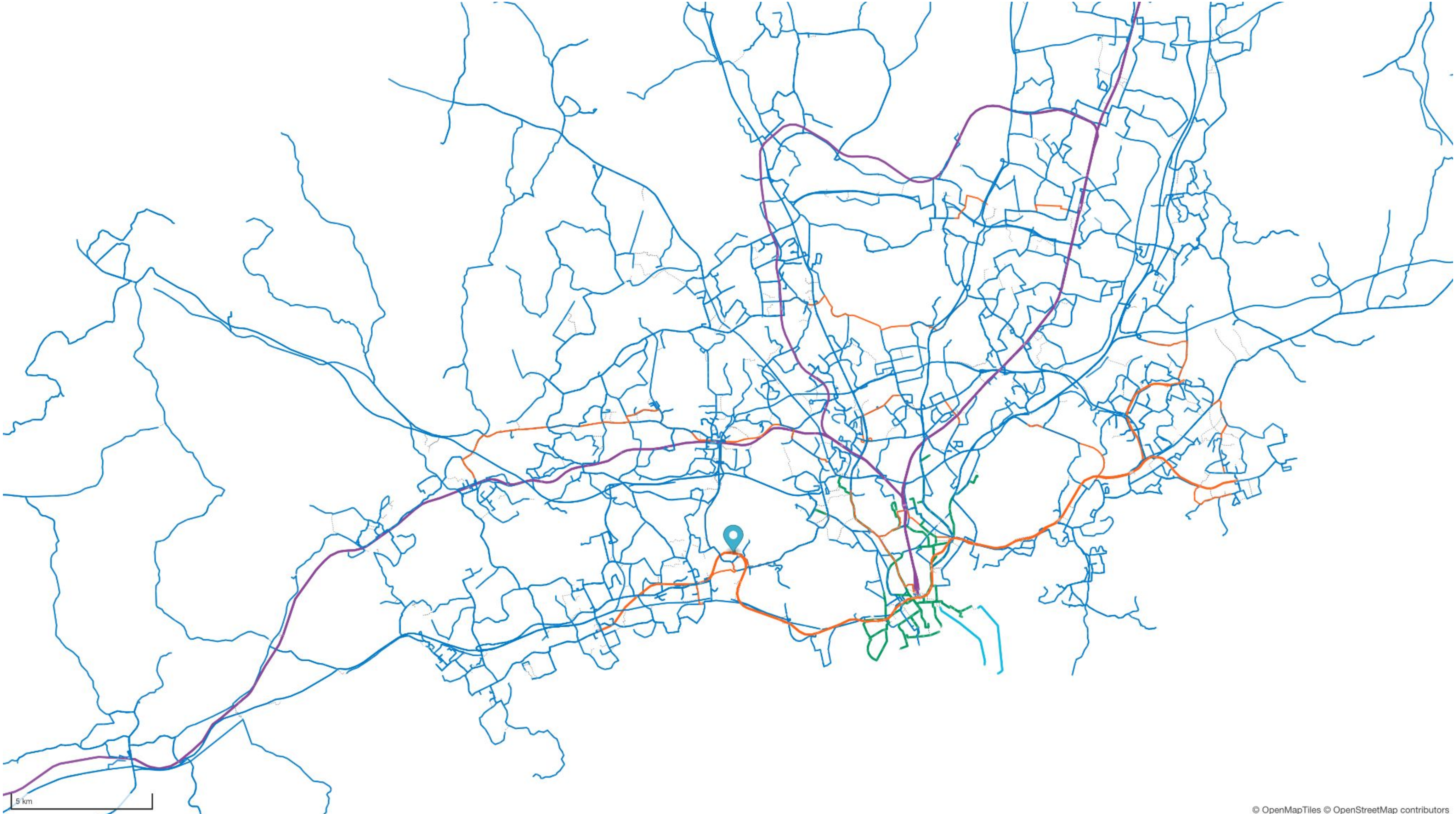
5 km
© OpenMapTiles © OpenStreetMap contributors


5 km
© OpenMapTiles © OpenStreetMap contributors


5 km
© OpenMapTiles © OpenStreetMap contributors



© OpenMapTiles © OpenStreetMap contributors



© OpenMapTiles © OpenStreetMap contributors

State	Transit Strategy	Search direction	Optimization	Result	Response time
Standard	Standard	Forward	-	Paths [time, transfers]	66 ms
Standard	Standard	Reverse	-	Paths [time, transfers]	68 ms
BestTime	Standard	Forward	-	Best time & hops, No paths	63 ms
BestTime	Standard	Reverse	-	Best time & hops, No paths	60 ms
Standard	NoWait	Forward	1 iteration	Paths [time, transfers]	49 ms
Standard	NoWait	Reverse	1 iteration	Paths [time, transfers]	48 ms
BestTime	NoWait	Forward	1 iteration	Best time & hops, No paths	41 ms
BestTime	NoWait	Reverse	1 iteration	Best time & hops, No paths	37 ms
MC	MC	Forward	-	Paths [time, transfers, cost]	508 ms
MC	MC	Forward	Heuristic Destination Check	Paths [time, transfers, cost]	320 ms

28
Samples

Search
Window
2 - 20 h

Dataset
Norway

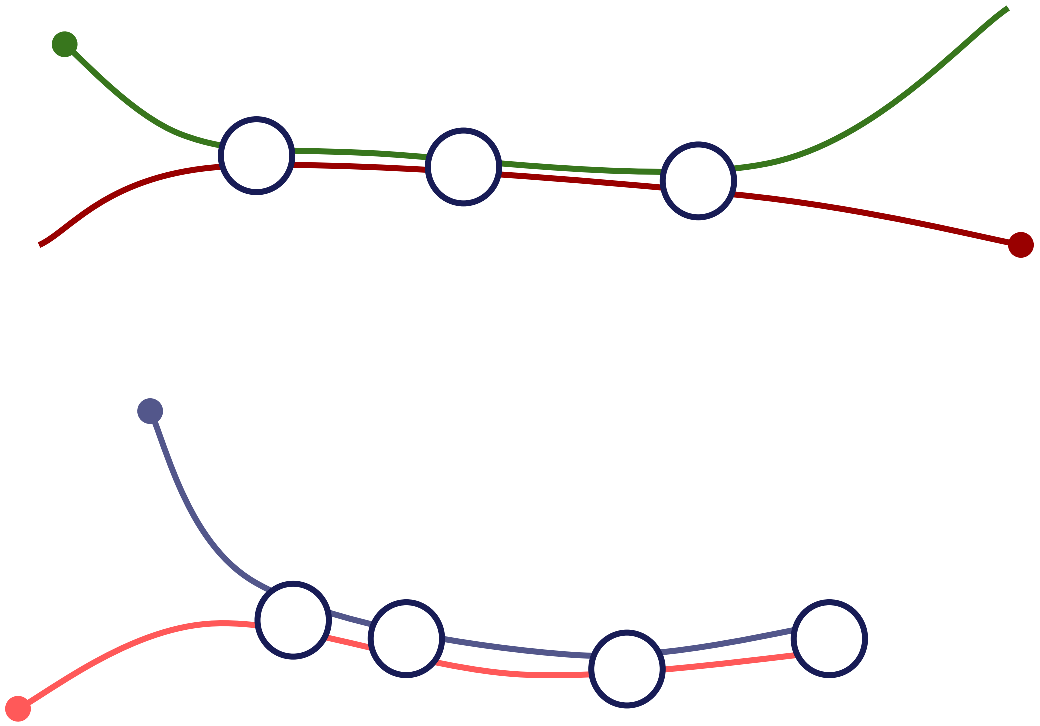
ENTUR

Transfer optimization

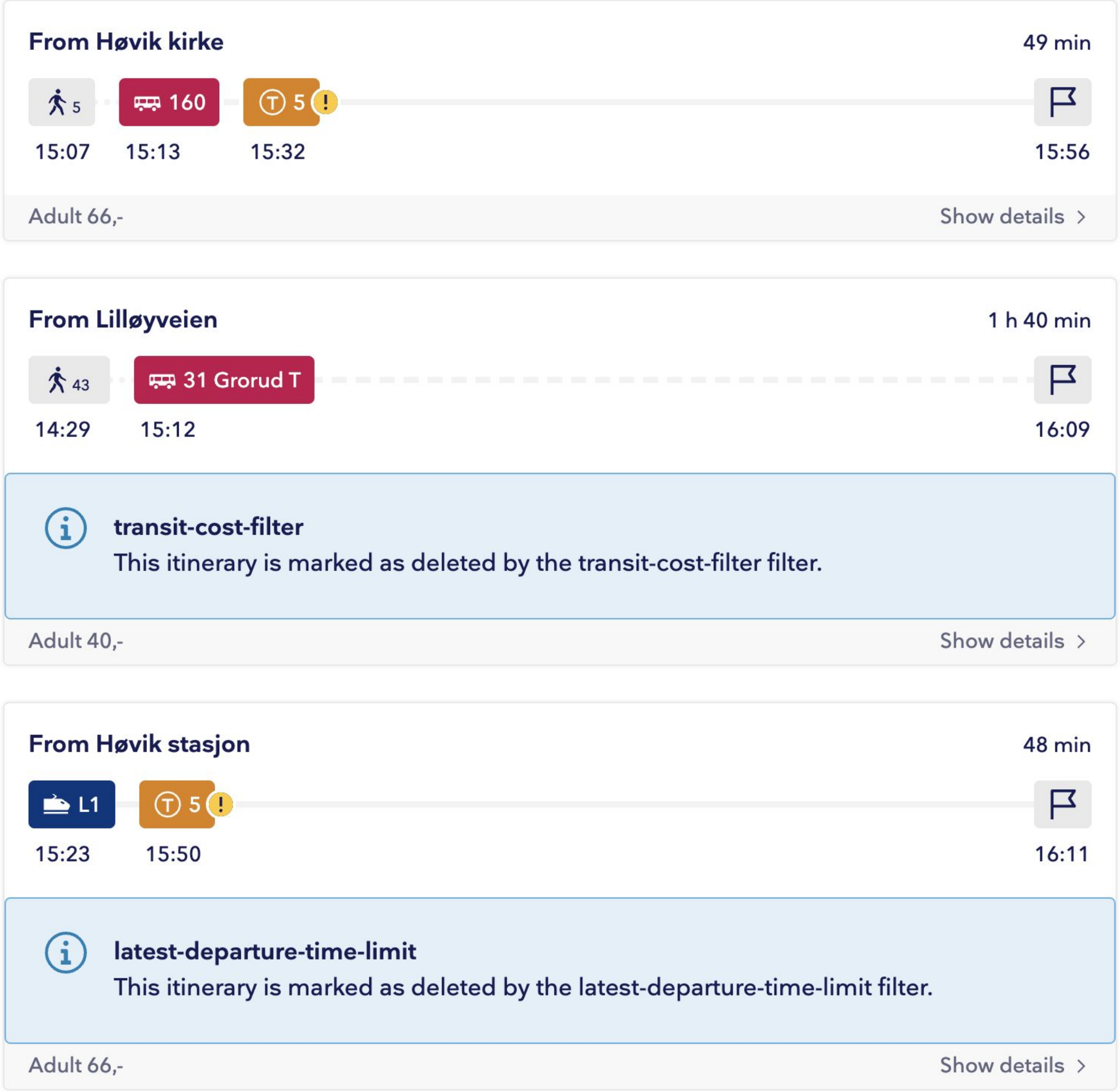
Where to transfer between a pair of trips?

- Transfer priority cost
  - Station transfer priority*
  - Guaranteed transfers
  - In-seat transfers

- Optimal wait time
  - Avoid very short transfer times*
  - Avoid back-travel*



* Not in raptor



From Høvik kirke 49 min

5 · 160 · 5

15:07 15:13 15:32 15:56

Adult 66,- Show details ›

From Lilløyveien 1 h 40 min

43 · 31 Grorud T

14:29 15:12 16:09

transit-cost-filter
This itinerary is marked as deleted by the transit-cost-filter filter.

Adult 40,- Show details ›

From Høvik stasjon 48 min

L1 · 5

15:23 15:50 16:11

latest-departure-time-limit
This itinerary is marked as deleted by the latest-departure-time-limit filter.

Adult 66,- Show details ›

Itinerary Filtering & Decorating

- Limit the number of results
  - Worse but optimal results
  - Grouping too similar results
  - Park & ride, where the car is parked almost immediately

- Decorate results
  - Real-time alerts
  - Price calculation

- Sorting of results

NeTEx–GTFS

- New internal data model independent of the import format
  - OTP 1 used GTFS POJOs internally
- New entities from NeTEX not existing in GTFS

Different formats have different benefits

- GTFS
  - Easy to produce and consume
  - YAGNI — Requires producer and consumer before appending spec
- Netex
  - Much more complex and nuanced
  - Caters for almost all use cases

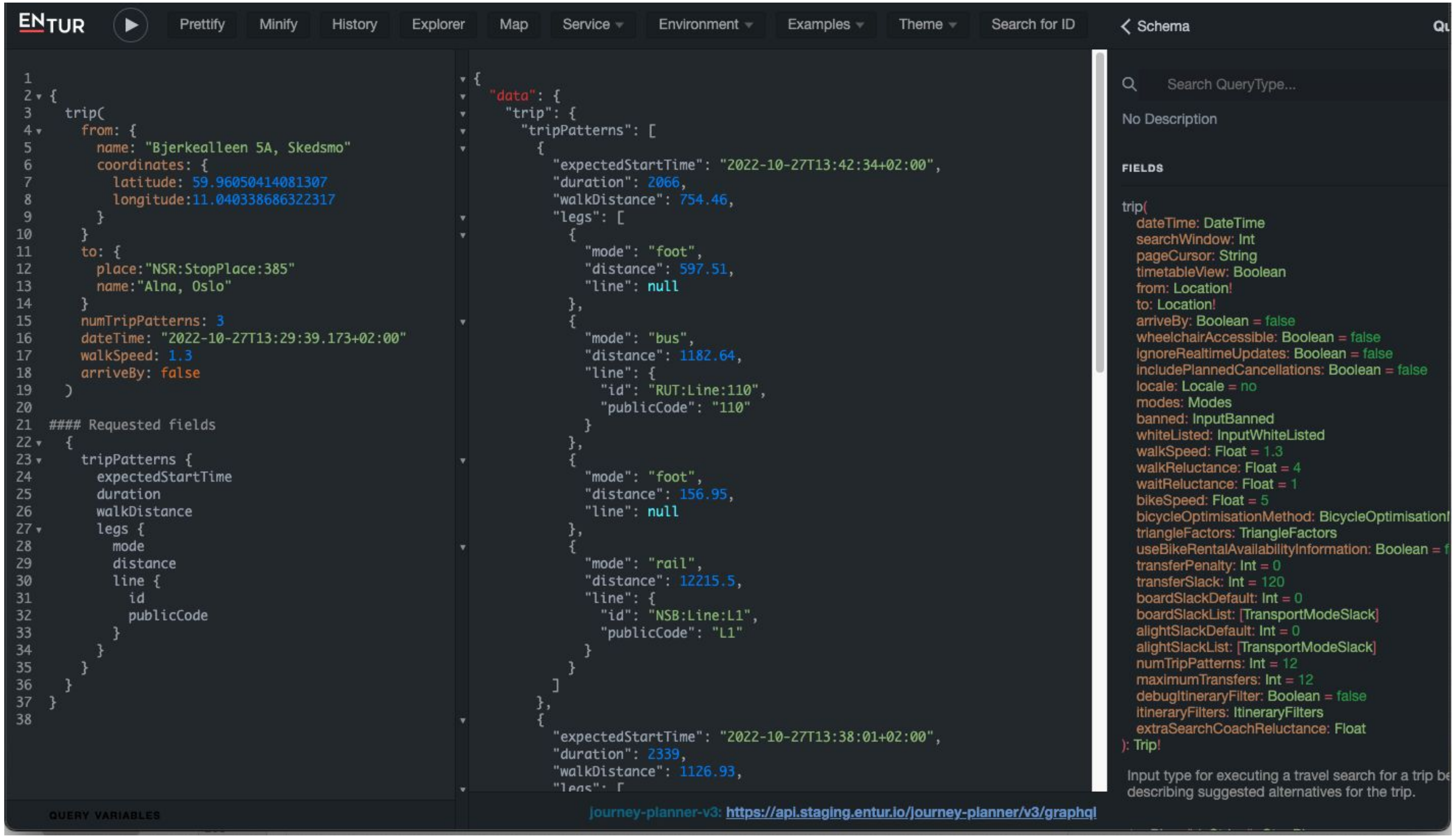
Sandbox extensions

- New feature in OTP 2 for code not suited for core
- Extremely successful, currently 22 extensions
  - New APIs
    - GraphQL
    - Travel time
    - Vector tiles
  - New data formats
    - Data overlay
    - SIRI
  - New functionality under development
    - GTFS-Fares v2
    - GTFS-Flex v2
  - Deployment-specific code
    - Non-GBFS vehicle rental updaters
    - Cloud integrations

GraphQL APIs

Two APIs with different vocabularies

- GTFS
- Transmodel



```
{
  trip(
    from: {
      name: "Bjerkealleen 5A, Skedsmo"
      coordinates: {
        latitude: 59.96050414081307
        longitude:11.040338686322317
      }
    }
    to: {
      place:"NSR:StopPlace:385"
      name:"Alna, Oslo"
    }
    numTripPatterns: 3
    dateTime: "2022-10-27T13:29:39.173+02:00"
    walkSpeed: 1.3
    arriveBy: false
  )

#### Requested fields
  {
    tripPatterns {
      expectedStartTime
      duration
      walkDistance
      legs {
        mode
        distance
        line {
          id
          publicCode
        }
      }
    }
  }
}
```

```
{
  "data": {
    "trip": {
      "tripPatterns": [
        {
          "expectedStartTime": "2022-10-27T13:42:34+02:00",
          "duration": 2066,
          "walkDistance": 754.46,
          "legs": [
            {
              "mode": "foot",
              "distance": 597.51,
              "line": null
            },
            {
              "mode": "bus",
              "distance": 1182.64,
              "line": {
                "id": "RUT:Line:110",
                "publicCode": "110"
              }
            },
            {
              "mode": "foot",
              "distance": 156.95,
              "line": null
            },
            {
              "mode": "rail",
              "distance": 12215.5,
              "line": {
                "id": "NSB:Line:L1",
                "publicCode": "L1"
              }
            }
          ]
        },
        {
          "expectedStartTime": "2022-10-27T13:38:01+02:00",
          "duration": 2339,
          "walkDistance": 1126.93,
          "legs": [
```

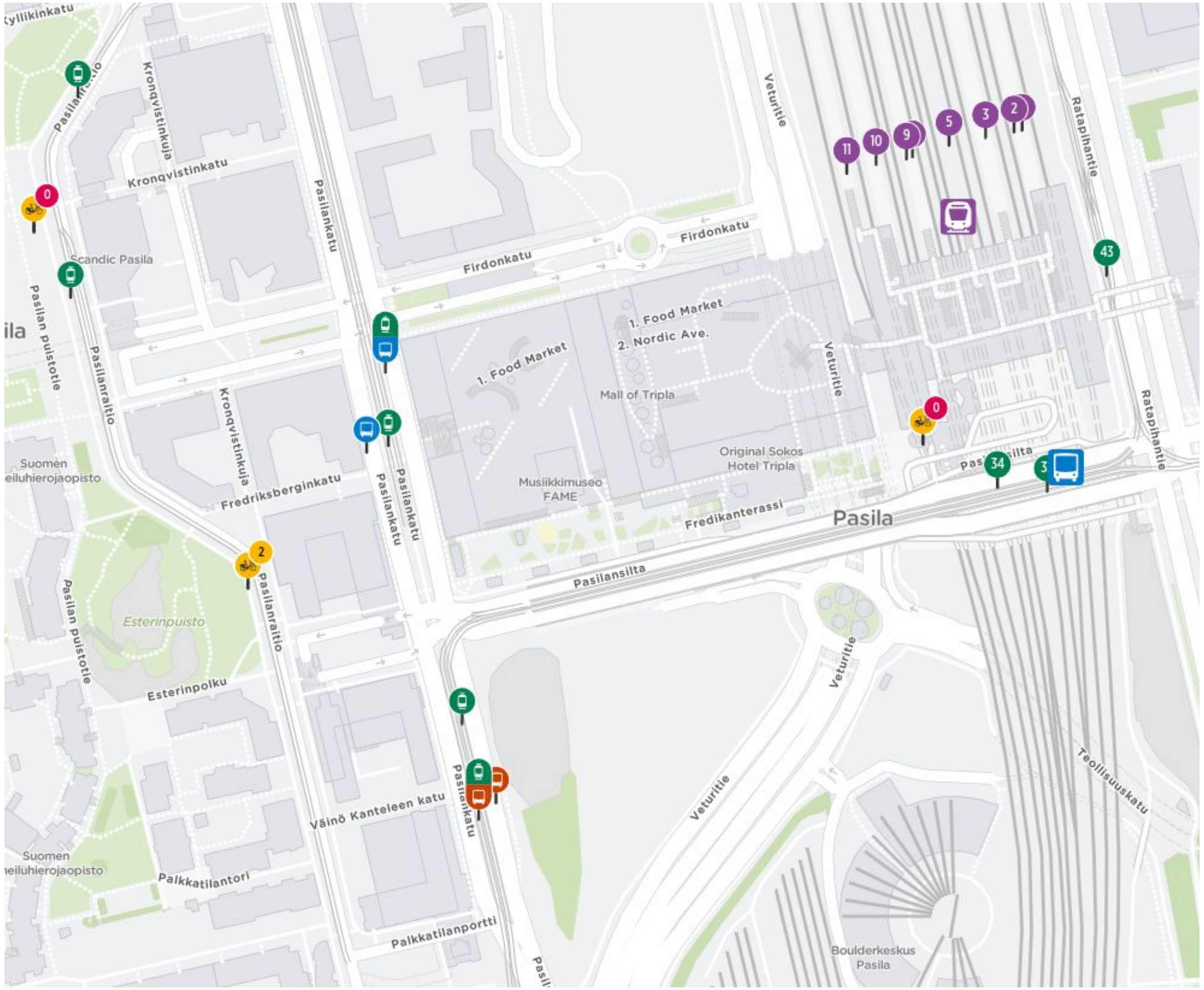
journey-planner-v3: https://api.staging.entur.io/journey-planner/v3/graphql

No Description

FIELDS

```
trip(
  dateTime: DateTime
  searchWindow: Int
  pageCursor: String
  timetableView: Boolean
  from: Location!
  to: Location!
  arriveBy: Boolean = false
  wheelchairAccessible: Boolean = false
  ignoreRealtimeUpdates: Boolean = false
  includePlannedCancellations: Boolean = false
  locale: Locale = no
  modes: Modes
  banned: InputBanned
  whiteListed: InputWhiteListed
  walkSpeed: Float = 1.3
  walkReluctance: Float = 4
  waitReluctance: Float = 1
  bikeSpeed: Float = 5
  bicycleOptimisationMethod: BicycleOptimisation
  triangleFactors: TriangleFactors
  useBikeRentalAvailabilityInformation: Boolean = f
  transferPenalty: Int = 0
  transferSlack: Int = 120
  boardSlackDefault: Int = 0
  boardSlackList: [TransportModeSlack]
  alightSlackDefault: Int = 0
  alightSlackList: [TransportModeSlack]
  numTripPatterns: Int = 12
  maximumTransfers: Int = 12
  debugItineraryFilter: Boolean = false
  itineraryFilters: ItineraryFilters
  extraSearchCoachReluctance: Float
): Trip!
```

Input type for executing a travel search for a trip be
describing suggested alternatives for the trip.

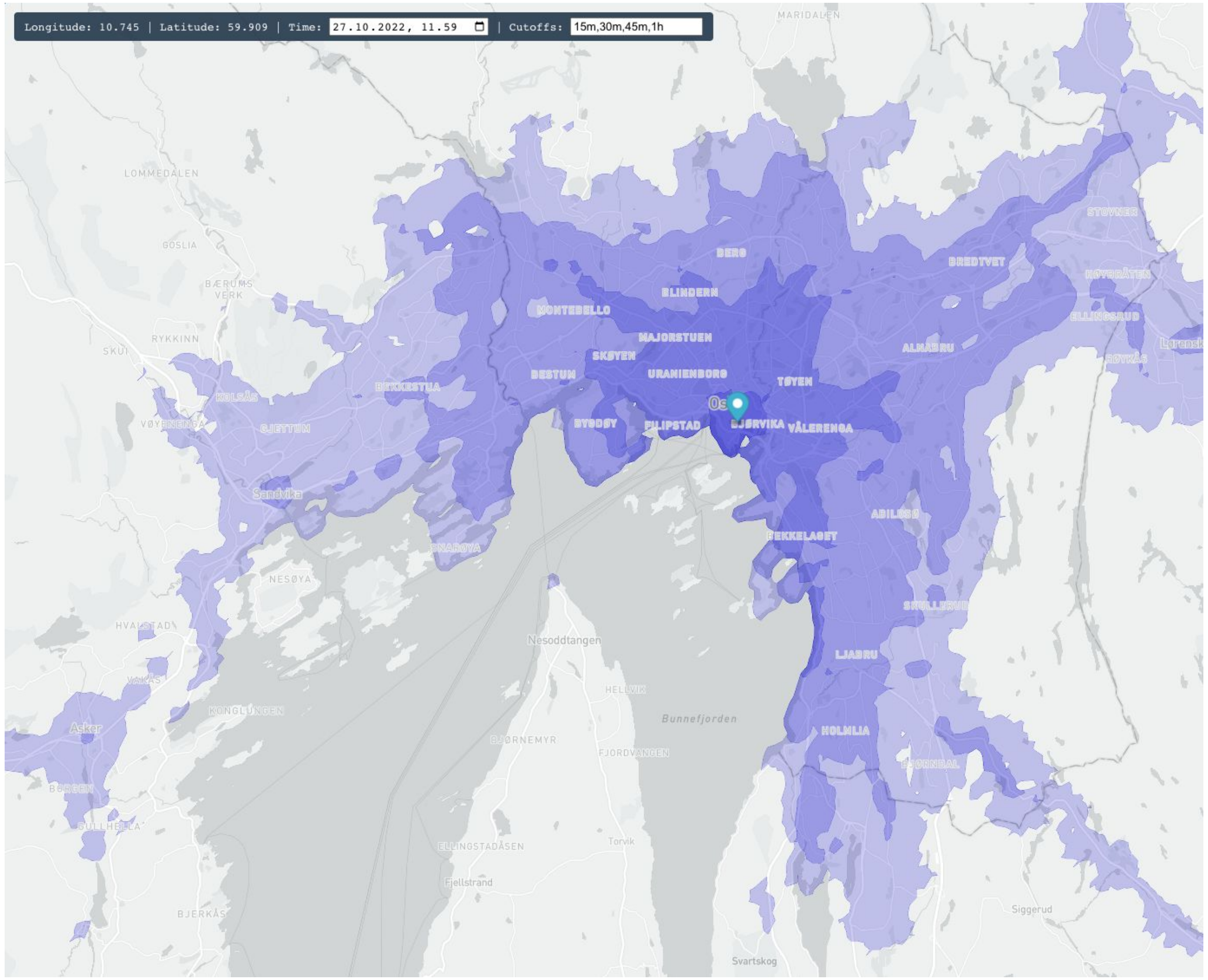


Vector tiles

- Mapbox vector tile format
- Multiple layer types available
  - Stops & stations
  - Rental stations & vehicles
  - Car and bike parking
- Configurable mapping from internal model to tile layer
  - Real-time info
  - Multilingual

Travel time analysis

- Requested feature from OTP 1
- Two output formats
  - GeoJSON — isochrones
  - GeoTIFF — travel time rasters
- Configurable street & transit modes



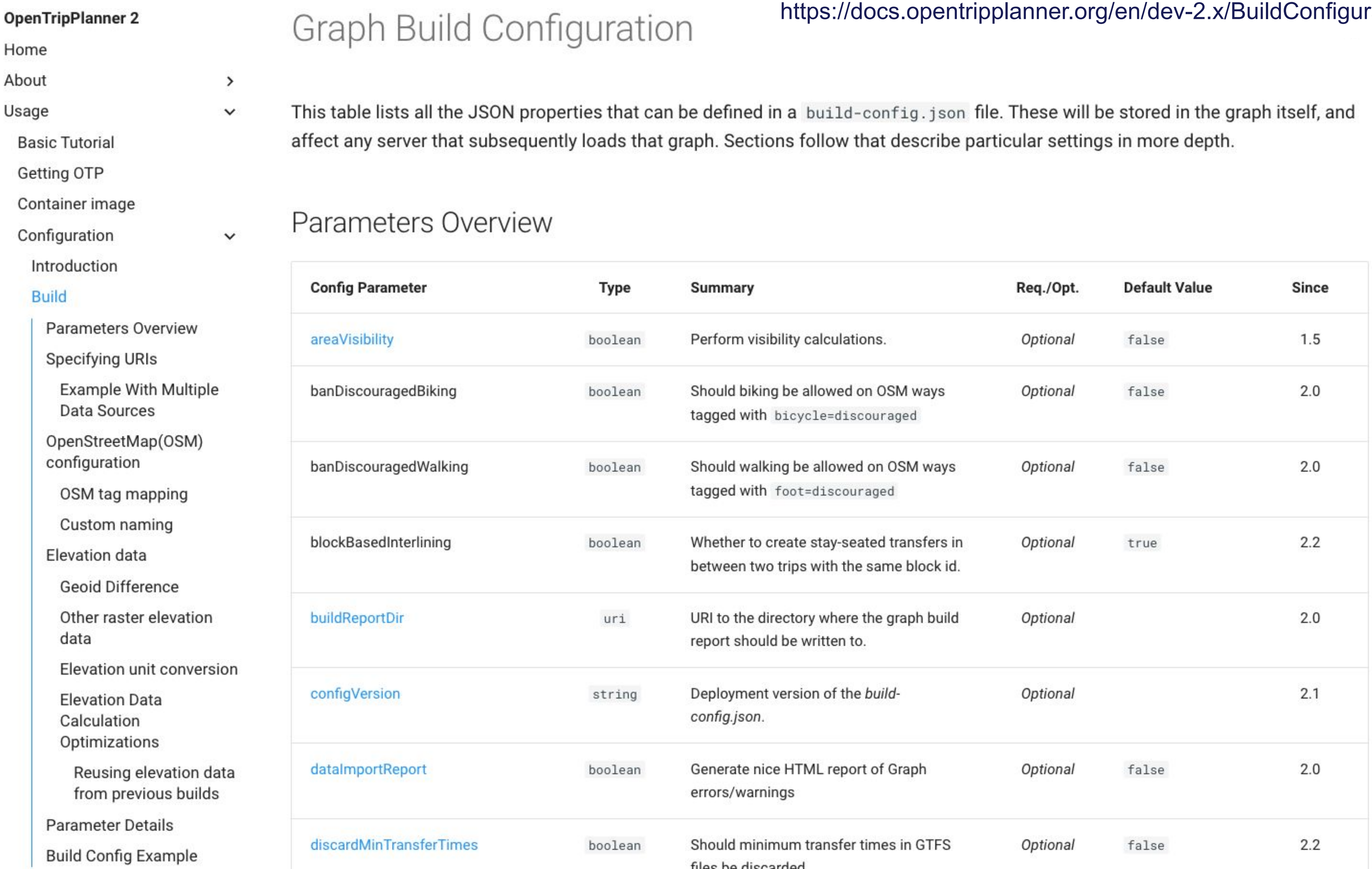
Longitude: 10.745 | Latitude: 59.909 | Time: 27.10.2022, 11.59 | Cutoffs: 15m,30m,45m,1h

ENTUR

Simplified operations

- Abstracted data sources
  - Local file system
  - HTTPS
  - Cloud storage services
    - GCP storage
    - Azure blob storage
    - AWS S3 (open PR pending somebody using it)
- All input and output paths can be configured
  - Data can be read from or written to an data source
- Improved monitoring support
  - Prometheus endpoint

https://docs.opentripplanner.org/en/dev-2.x/BuildConfiguration/



OpenTripPlanner 2

- Home
- About ›
- Usage ⌄
  - Basic Tutorial
  - Getting OTP
  - Container image
  - Configuration ⌄
    - Introduction
    - Build
      - Parameters Overview
      - Specifying URIs
        - Example With Multiple Data Sources
      - OpenStreetMap(OSM) configuration
        - OSM tag mapping
        - Custom naming
      - Elevation data
        - Geoid Difference
        - Other raster elevation data
        - Elevation unit conversion
        - Elevation Data Calculation Optimizations
          - Reusing elevation data from previous builds
      - Parameter Details
      - Build Config Example

Graph Build Configuration

This table lists all the JSON properties that can be defined in a `build-config.json` file. These will be stored in the graph itself, and affect any server that subsequently loads that graph. Sections follow that describe particular settings in more depth.

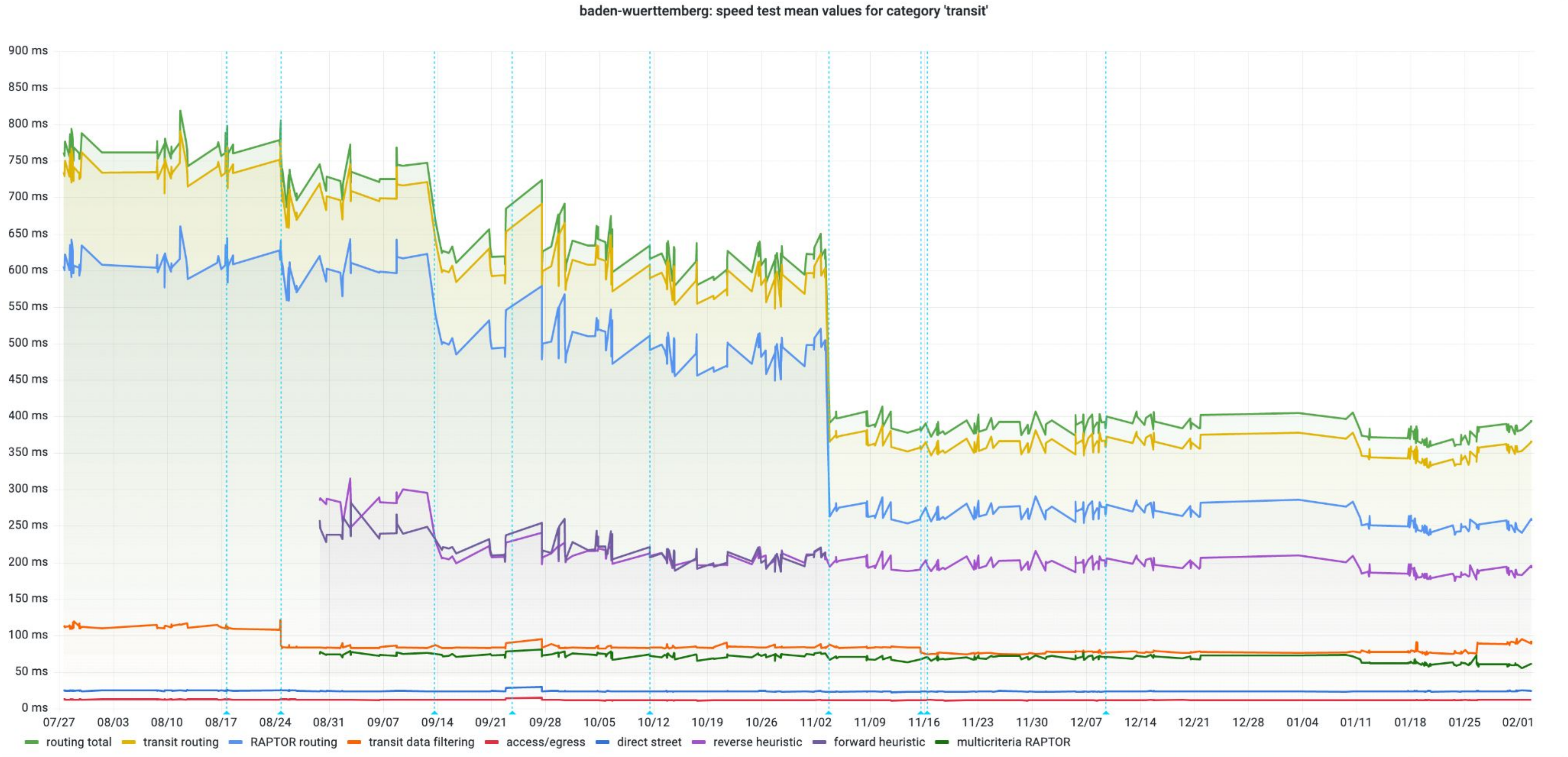
Parameters Overview

Config Parameter	Type	Summary	Req./Opt.	Default Value	Since
areaVisibility	`boolean`	Perform visibility calculations.	*Optional*	`false`	1.5
banDiscouragedBiking	`boolean`	Should biking be allowed on OSM ways tagged with `bicycle=discouraged`	*Optional*	`false`	2.0
banDiscouragedWalking	`boolean`	Should walking be allowed on OSM ways tagged with `foot=discouraged`	*Optional*	`false`	2.0
blockBasedInterlining	`boolean`	Whether to create stay-seated transfers in between two trips with the same block id.	*Optional*	`true`	2.2
buildReportDir	`uri`	URI to the directory where the graph build report should be written to.	*Optional*		2.0
configVersion	`string`	Deployment version of the *build-config.json*.	*Optional*		2.1
dataImportReport	`boolean`	Generate nice HTML report of Graph errors/warnings	*Optional*	`false`	2.0
discardMinTransferTimes	`boolean`	Should minimum transfer times in GTFS files be discarded	*Optional*	`false`	2.2

```
//build-config.json
{
  "transitModelTimeZone": "Europe/Brussels",
  "osmCacheDataInMem": "true",
  "osm": [
    {
      "source": "https://download.geofabrik.de/europe/belgium-latest.osm.pbf",
      "osmTagMapping": "germany"
    }
  ],
  "transitFeeds": [
    {
      "type": "gtfs",
      "feedId": "NMBS",
      "source": "http://gtfs.irail.be/nmbs/gtfs/latest.zip"
    },
    {
      "type": "gtfs",
      "feedId": "LIJN",
      "source": "http://gtfs.irail.be/de-lijn/de_lijn-gtfs.zip"
    },
    {
      "type": "gtfs",
      "feedId": "TEC",
      "source": "https://gtfs.irail.be/tec/tec-gtfs.zip"
    },
    {
      "type": "gtfs",
      "feedId": "MIVB",
      "source": "https://gtfs.irail.be/mivb/mivb-gtfs.zip"
    },
  ]
}
```

Future

Performance



Competition neutrality

- New Raptor criteria
- Fixed size bitset for used authority/operator group

- Operator 1, departure from A at 18:00, arrival at B at 21:00
- Operator 2, departure from A at 18:01, arrival at B at 20:59
- => Only operator 2 is showing up

Unified GraphQL API

- Currently two GraphQL APIs and one REST API
  - Deprecate REST API
- New unified GraphQL API
  - One structure
  - Two dialects, GTFS and Transmodel
  - Use translation file go from internal model to API

ENTUR

Useful links

- https://www.opentripplanner.org/
- https://docs.opentripplanner.org/en/dev-2.x/
- https://github.com/opentripplanner/OpenTripPlanner
- https://gitter.im/opentripplanner/OpenTripPlanner
- https://otp-performance.leonard.io/