Gemeente Amsterdam
Dienst Infrastructuur Verkeer en Vervoer

Towards a multimodal API

Road plan to provide the optimal open data for developers to build a multimodal app

Version 1.0, April 3rd, 2013

Project:	Multimodal API	
Document name:	IVV – Multimodal API – Report	
Version:	1.0	
Authors:	Paul Manwaring	Glimworm
	Jonathan Carter	Glimworm
	Erik Romijn	Solid Links
	Ron van der Lans	Braxwell
	Jasper Soetendal	Braxwell
Date:	April 3rd, 2013	

Contents

This report was commissioned by Gemeente Amsterdam, Dienst Infrastructuur, Verkeer en Vervoer

1 Introduction

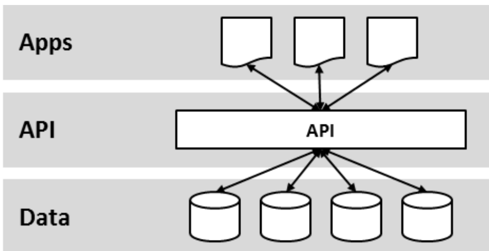
1.1 *Cause for the assignment*

For some time there is a topic within the transport domain of the municipality that has specific political attention: *the multimodal app*.
The multimodal app is an all-aware route planner that combines all different modes of transport in a route planner. It plans a route in advance and/or changes the route in real-time, due to delays and other circumstances. The ideal multimodal app will tell you to park your car at a P+R and travel by public transport in rush hour, will divert you if the tram you planned to catch is delayed or will guide you around construction sites or traffic jams.

Over a year ago, DIVV (Department of Infrastructure, Traffic and Transport of the city of Amsterdam) announced their Open Data motto “We provide the data, you build the apps (*Wij de data, u de apps*)”. They stated that apps should be developed by the open market, the municipality is dedicated to the provision of Open Data.

Until now, there have been no strong initiatives from the open market in the direction of a multimodal app. To encourage such developments, DIVV wants to examine the development of a multimodal API. By bringing data from different modes of transport together in one API the development of such apps are much easier.

A multimodal API (Application Programming Interface) provides an interface for developers to the data they need to build a multimodal route planner. Without an API a developer has to get all the data from all different providers and needs a lot of expertise and work to combine them together. With a multimodal API a developer can connect to one interface, and can get exactly the data he or she needs, in one uniform format.



Objective for this project is to answer the most relevant question regarding this multimodal API:
- Can a multimodal API be built now?
    - What are the requirements on the data for building a multimodal API?
    - Is this data available?
    - Is the quality of this data sufficient?
- What are the next steps towards a multimodal API?

1.2 *Assignment*

Our assignment for this project is to answer these questions, to investigate the potential for DIVV to achieve a multimodal API with the current set of data sources.

The best way to find out if it is possible to build a multimodal API, is to actually try to build one. Our assignment for this project is to:

- Determine objectives and constraints for a multimodal API
- List and research all available data sources
- Define the most likely use cases for a multimodal API
- Build a proof of concept for the multimodal API
- Report on the most important data sources and the lessons learned while building the proof of concept

1.3 The team

For this assignment, a team is composed with a variety of relevant expertise.

- **Glimworm** (Jonathan Carter, Paul Manwaring and their team) has extensive experience on Open Data and the development of applications. They developed the ParkShark API based on the parking data of IVV, which won the Apps for Amsterdam competition in 2012.
- **Solid Links** (Erik Romijn) is a specialist from the Appsterdam network with much experience on the field of API development and public transport data.
- **Braxwell** (Ron van der Lans and Jasper Soetendal) has detailed knowledge on the Open Data strategy of DIVV and the city of Amsterdam. For this project they've focused on the non-technical elements of the assignment.

1.4 Approach

1. **Catalogue of appropriate data sources**
   We've constructed a list of all available data sources that might be suitable for the API. For all databases we've defined both information and technical aspects, such as coverage, frequency, granularity, format and licensing.
2. **Definition of the most likely use cases**
   We've defined five use case scenario's on how the multimodal app may be used. Using the 80/20 rule, these most likely uses cases cover 80% of the expected use:
   a. **Frequent traveller.** A person on a normal route to work (or other regular journey), should be signalled if there is a delay somewhere in journey that can be avoided if they reroute NOW.
   b. **The basic A -> B planner.** This person is starting and ending their journey within Amsterdam (or its nearby surroundings). They may be a resident or a frequent visitor.
   c. **The Amsterdam Visitor.** This person is visiting Amsterdam from another town or city and can travel by car or train into the city. They want to know how best to travel to Amsterdam and how to get from either the train station or parking garage to their final destination.
   d. **The new tourist.** This person is a tourist who arrives probably by Air at Schiphol and is not online yet, they need basic information about how to proceed with transportation within the city, for instance about buying on OV card, and how to get to their initial destination such as a hotel.
   e. **The destination planner.** This is a variation on the A -> B planner, more like an A -> B -> C -> Z planner with multiple stops and destinations on the way.

   More detailed descriptions of these use cases are in the WIKI (https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Usecases)
3. **Technical and functional investigation with proof of concept**
   For each of the use case we've defined what data is needed, and what data source can be used, or what data sources can be combined to provide the described functionality. We've connected to these data sources and we've built a proof of concept for the multimodal API, consisting of two parts: a Part A used for static planning and a Part B for real time disruption monitoring.

4. **Report on the lessons learned and most important data sources**
Intermediate results are discussed with DIVV and the results of the proof of concept will be reported. This document is the main deliverable for this last step.
5. **Leave a digital footprint of what we've done**
By building a proof of concept of the multimodal API, we've left a digital footprint of all challenges we've overcome, issues we 've discovered and code we've created. All this (both in text and code) is available as open data, to be built upon by other developers.

1.5 Results

This document describes the results and focuses on the target groups and targets below:

- **For developers:**
    - Sharing the knowledge gained in the process of setting-up a proof of concept.
- **For data providers (including IVV):**
    - Advising on every level and every aspect where and how data should be approved, and how the current issues effect the developers.
    - Stating what data is still missing to build the multimodal app.
- **For IVV:**
    - Providing insight in what data, time, money and expertise is needed to build the actual API

1.5.1 Contents of document

- In Chapter 2 the most relevant data sources are listed, what data sources are used in the proof of concept and what alternatives are available.
- In Chapter 3 we provide the lessons learned while building the proof of concept, by stating the complexity of a multimodal API and the challenges to build the ultimate multimodal API.
- In Chapter 4 we have described what needs to be improved: what data is still missing and how the available data can be improved.
- In Chapter 5 contains the next steps to be taken towards a multimodal API.

1.5.2 Digital footprint

This report provides an summary of the results of our effort to create a proof of concept for the multimodal app. Our actual approach, the issues we've found and the challenge we've overcome are documented online and the source code we've created is available as open source. Reuse of all this is highly encouraged.

See: https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report

This document focus on the lessons learned, tips and tricks we've discovered while building a proof of concept. For a full description of our approach, the use case scenarios and the data sources we've investigated and used, see the Wiki-part of the Github (https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki)

2 Most important data sources

In this chapter we will summarize the different data sources we have and haven't used in the proof of concept. For each data source we will define for what purpose it is or can be used and what the known opportunities or limitations are to use it in a multimodal API.

2.1 Data sources used in the proof of concept

2.1.1 MapQuest

Used in proof of concept for:	Route planning for driving
Can it be used in multimodal API?	Yes. The MapQuest Open Directions API Web Service is based on open source data and may be used for free, under the condition that OpenStreetMap and MapQuest is given credit. These conditions are the main reason we choose MapQuest over Google, which provides a similar service, but under more strict conditions (see 'Alternatives').
Is it expected to be working the next three years?	MapQuest is a service of AOL, a big Internet company. However, it is less known and used than similar services from Google. We expect AOL to provide these services for years to come, but no guarantee can be given.

2.1.2 Open Trip Planner

Used in proof of concept for:	Route planning for walking, train, tram, bus, metro
Can it be used in multimodal API?	Yes, Open Trip Planner is an open source initiative which can be used for free.
Is it expected to be working the next three years?	The Dutch Open Trip Planner is set up by the same team that built the OpenOV API. As far as we know this is not a profitable business yet, so the initiative is depending on the open source community. Continuity for the platform to exist for the next three year is not guaranteed. But the source code of OpenTripPlanner is open, so an instance of the service can be deployed and maintained on an own IT-environment.

2.1.3 NS

Used in proof of concept for:	Real time information for train
Can it be used in multimodal API?	Yes, the NS API is available for use for free, although the NS provides the data 'AS IS' and doesn't guarantee any correctness or continuous delivery.
Is it expected to be working the next three years?	The NS didn't make any commitment to guarantee the existence of this service for the years to come. Continuity is not guaranteed.

2.1.4 OpenOV

Used in proof of concept for:	Real time data on Tram, Bus and Metro
Can it be used in multimodal API?	Yes, the OpenOV can be used for free in a multimodal API. Because of the current situation (see below) it is unknown if it is needed to run their open source software on a own IT platform or that they will continue to provide an API service.
Is it expected to be working the next three years?	OpenOV is depending on the open source community, continuity isn't guaranteed. At the same time, the government is working on setting up a ND-OV service (Nationale Public Transport Data). Current contestants for this role are 9292 and OpenOV. They will be required to take and distribute all data and may charge a max of € 1.000/year. If this service is set-up it is expected that the data will be available for the coming years, but it could be in a different form and requiring more effort to get the needed information.

2.1.5 IVV

Used in proof of concept for:	Real-time road traffic information within Amsterdam and information on locations of P+R, parking garages, etc.
Can it be used in multimodal API?	Yes. (See also: "What should be approved")
Is it expected to be working the next three years?	We expect IVV to provide this data (in some open format) for the coming three years.

In addition to the data sources mentioned above, for the example use of the API we have used multiple data sources for looking up geographic locations for starting points and destinations. These services are quite common and will be available for a multimodal API.
In our Proof of Concept we've used OpenStreetMap, 9292, Google Maps and datasets from IVV for this purpose.

2.2 Data sources not used in the proof of concept

Apart from the data sources we've used for the proof of concept, other data sources may provide a viable alternative for use in the multimodal API.

Data source	Alternative to	Notes
MapQuest Commercial	MapQuest Open	MapQuest is the only company that offers an open license and a commercial license. The commercial version uses licensed map data instead of open map data. The commercial license offers a free version with unlimited map views and max. 5.000 routing request per day.
Google Maps	MapQuest Open	Google Maps is a viable alternative to both

		MapQuest and parts of the OpenTripPlanner. However, it's not for free. The free license does not allow the route planner to be used other than route planning displayed on Google Maps and is limited to 2.500 routes a day. Licenses without these limitations start at $ 17.500/year
cloudmade.com, Nokia Navteq, Microsoft Bing, OpenLayers, modest maps, polymaps	MapQuest Open	Multiple other route planning alternatives exists as an alternative to MapQuest. For more detailed information on these alternatives, see the "Available data" page on the WIKI.
TomTom	IVV Real-time traffic data	The real-time traffic data of TomTom is more detailed and has a broader geographical scope than IVV's. However it needs to be purchased from TomTom, and it's most unlikely that this data can be provided for free in an open multimodal API.
Waze	IVV Real-time traffic data	Waze sells their real time traffic data to other parties. They collect this data from the users of their own (free) app. Worldwide, this app has more than 20 million users.
Smart City SDK	Multiple data sources (including OpenOV)	Smart City SDK is an European project which provides a uniform API to multiple (local) data sources. It's still in a test phase and only funded until 2014, but it's a promising initiative to keep an eye on.
9292	Open Trip Planner	At the moment, 9292 provides no open API for route planning on public transport. We don't expect 9292 to allow third parties to use their route planning system anytime soon. Although 9292 has a road map for open data, providing route planning intelligence is not on it. But if they do, eventually, free or paid, it would be a viable alternative to Open Trip Planner.

2.3 Conclusion on data sources to be used

Based on the current and expected availability, the data sources are sufficient for building a multimodal API in the way we will discuss in the next chapter.

While not all data sources we've used in our proof of concept may be used in a multimodal API for free and not all data sources will be guaranteed to be still available in three years, alternatives exists for most of them. And likely there will be more alternatives in the future.
But it's very difficult to get guarantees on the availability of these services in the years to come, especially when it is provided as open data without any kind of Service Level Agreements.

If a multimodal API is going to be built, it should have a modular structure for the data sources to be used. In such a way that when a data provider will discontinue its data delivery, it can be replaced by another. Of course replacing a data provider for another will incur extra efforts and costs, but a modular structure should keep these costs as low as possible.

3 Lessons learned from the proof of concept

3.1 Can the multimodal API be built now?

The most important question needs to be answered first: Can the multimodal API be built with the current available data?
The answer is twofold:

- **NO,** at this moment the ultimate multimodal API, combining all modes of transport, knowledgeable of all current delays and instantly providing alternatives or suggestions for faster routes can't be build.
- **YES,** at this moment we can build a multimodal API which is at least one step ahead of the currently available solutions, combining all modes of transport and monitoring the current delay on the selected route(s).

3.2 What a multimodal API can deliver now

For the proof of concept for the multimodal API, we had to decide to split the multimodal route into two parts:

3.2.1 Part A – Static planning using transfer hubs

In this part a route can be planned from A to B. This takes as its input your starting point and destination, and variables as departure or arrival time and desired methods of transport, and calculates one or multiple travel options.

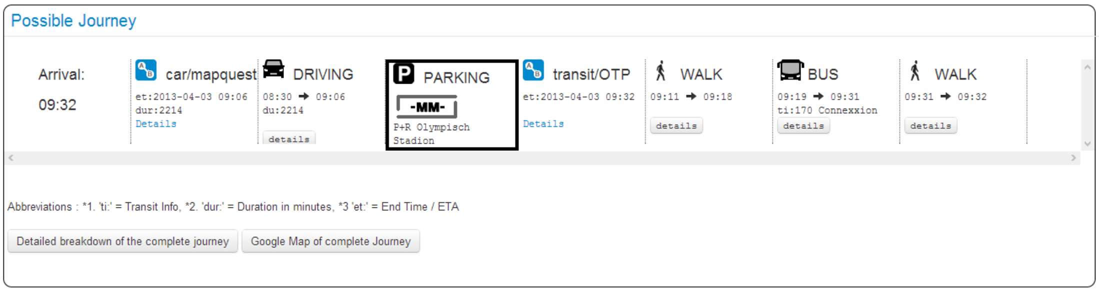
This part of the multimodal API uses four different route planners, and combines its results:

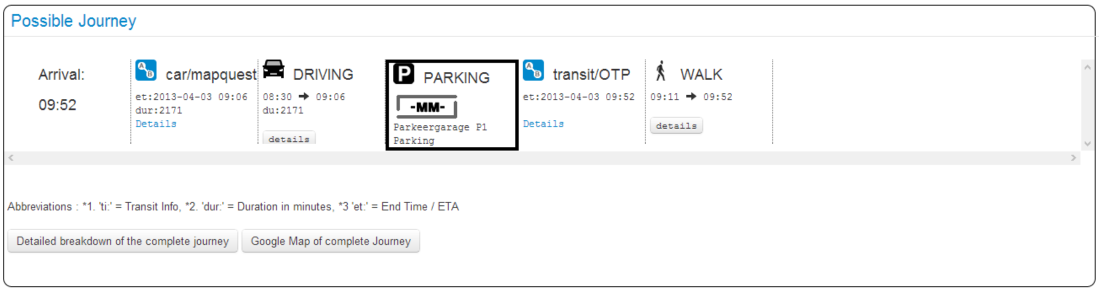
Planning system	Based on data	Covers
MapQuest	OpenStreetMap	Driving
OpenTripPlanner	OpenStreetMap + OpenOV GTFS data	Walking, Train, Tram, Bus, Metro
IVV	Own IVV data	Location of multimodal transfer hubs (P+R, parking garages, etc.)

In its approach, the API uses the concept of 'multimodal transfer hubs'. These are predefined locations which are known to be suitable to transfer from one transport mode to another. For example all P+R locations, all train stations and parking garages are transfer hubs.

The API uses a simple algorithm to calculate all options for a route: First it calculates all 'one transport mode only' routes from A to B, and then it calculates multimodal routes using the transfer hubs for switching between modes of transport. When all options are calculated, it returns all results, sorted on total travel time.

Demo application





For a prototype of how this API can be used, take a look at:
http://divvreport.glimworm.com/example/planning/plan.php

3.2.2 Part B – Real time disruption monitoring

In this part a specified journey is augmented with real time updates in order to see whether the journey is still viable or whether an alternative route needs to be planned. Input for this part of the API is the selected route from Part A, the planner.

This part of the API uses three different monitoring systems and combines its results:

Monitoring system	Based on data	Covers
OpenOV	GOVI data	Tram and Bus
NS	Own NS Data	Train
IVV	Own IVV Data	Driving (Amsterdam only)

The result of the API is a 'yes' or 'no' on whether the route is still viable and in more detail a delay (in minutes) on all parts of the route.

Demo application

The demo application on http://divvreport.glimworm.com/example/planning/plan.php uses real time monitoring, if the journey starts or ends within approximately two hours. Take a look at the demonstration video on http://divvreport.glimworm.com/example/index.php for more info.

3.2.3 Getting Part A and B together

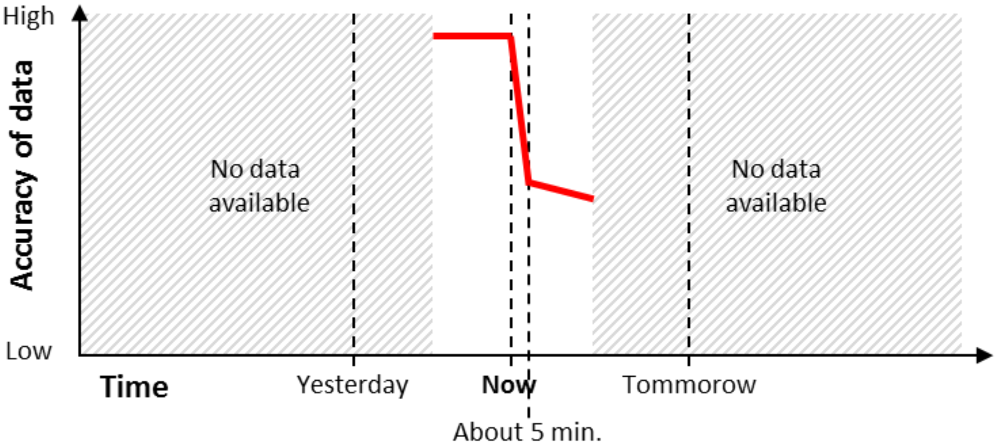
Currently it's not possible to integrate part A and B, they function as two separate modules. The result of Part A (the best route) is input for Part B. But as soon as Part B signals that the current route is delayed or not viable anymore, no direct alternative can be given. Part A should be used again to calculate an alternative, however these planning systems may not be aware of the current situation that caused the delay.

This problem, the difficult and currently impossible combination of planning and real time information is discussed as one of the issues in the next paragraph.

3.3 Why the ultimate multimodal API can't be built

3.3.1 Lack of intelligent prediction on near-future

The data that is provided by the data providers about the near-future is not accurate in all circumstances. Information on the current situation is in most cases correct, but as soon as any sort of prediction is involved, even on the situation in about 5 minutes from now, the accuracy of data is too low to provide intelligent multimodal route planning.



The figure above illustrates the accuracy of data over time in the cases we've studied. In chapter 4 multiple examples will be given on the accuracy of 'real-time' data in the near-future. These examples show that the current predictions are not intelligent enough to provide valuable multimodal routes in all kind of situations, both simple and complex.

3.3.2 Lack of educated guess for future situations

Apart from the lack of intelligent prediction on the near-future, which is based on the current information, there is currently no open data for prediction or expectations for travel times based on historic information. For any time frame more than two hours from now, there's no open data source for any kind of prediction, expectation or educated guess.
This kind of information is essential to the multimodal route planner. A planner should be aware of regular rush hours, it should know at least something on the chance of getting a parking place and it could be aware of the effect of the predicted weather on the traffic.

For example, a multimodal route planner suggesting to park your car at a P+R-station at noon tomorrow, while it's known to be full every morning at 08:00, doesn't add any value to the user. And if you're planning a trip to Amsterdam at Tuesday afternoon 17:00, the multimodal route planner should take the regular traffic jams into account.

Currently there's no open data source available for this kind of information.

3.3.3 Not all data is available or open

Even on the 'regular' traffic data, not all information is available or open. At this moment, there are no data sources on:

- Real time data on the GVB Metro's
- Real time traffic (on highways and non-Hoofdnet Auto)
- Costs of tram and bus (including all 'kortingskaarten')
- Parking availability (both on streets and in parking garages)

3.3.4 Planning and real time information are two worlds apart

In the current situation, planning a route and monitoring the real time traffic information are two worlds apart. There's no open planner that takes the real time information into account in its planning. And there's no open real time traffic information that provides alternative routes on delay.

In the current proof of concept, we are using the 'route planning intelligence' of existing route planners, MapQuest and OpenTripPlanner. At the moment, these planners and other alternatives are not aware of the real time information.

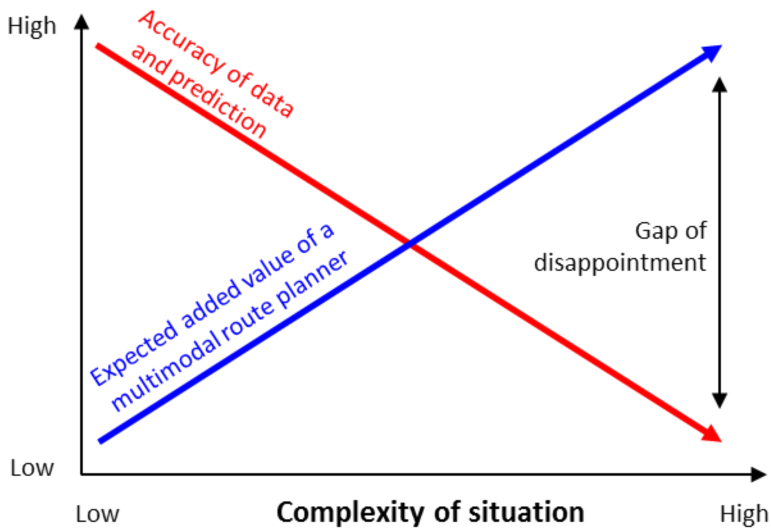
So at this time, it is not possible to integrate planning and real time traffic information in such a way that alternative routes can be given based on the current traffic situation. To solve this problem, two approaches exist:

- Existing route planners (such as MapQuest, Google Maps, OpenTripPlanner, 9292, etc.) should take care of this themselves, and use real time data in their route planning.
- The multimodal API should not rely on the 'route planning intelligence' of the current route planners, but build this intelligence itself. This is an extremely difficult and costly approach.

3.4 Why building a multimodal API is difficult

3.4.1 The "Gap of Disappointment"

The "Gap of Disappointment" is related to the lack of intelligence in the prediction of all transport modes. While the complexity of the situation increases (when the weather is bad, the highways are crowded or transport hubs are busy or shut down), the accuracy of the data and predictions decrease and the expected added value of a multimodal route planner (from a user's perspective) increases. This leaves a 'gap of disappointment' which is extremely difficult to overcome.



In other words, while everything is going smooth, the multimodal route planner will plan correctly. But these are not the situations when the real value of the multimodal route planner will be tested. The multimodal route planner is expected to deliver its added value in the most complex situations, when it's the most difficult to provide accurate information and alternatives. (Think about snow, closed stations, diverted routes, traffic jams, big events, etc.) As soon as the multimodal route planner will disappoint the user in these situations, he or she will ignore the planner and just use the car instead next time.

To bridge this "Gap of Disappointment" it is necessary to provide accurate predictions in the most complex situation, which is extremely difficult and costly.

3.4.2 Different formats are used by different parties

There are literally dozens of different data formats used by the data sources we've investigated for this project. Although platforms like GOVI and OpenOV have done a respectful amount of standardization, to build a multimodal API still a lot of different data formats need to be understood, translated and combined.

In the WIKI (https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Available-data) all data sources and their data formats we've used in the projects are listed. For the proof of concept we were able to combine all these source into one API, so with some effort it is possible to build the multimodal API. But with all the different data sources, maintaining and extending a multimodal API is complex; Every time a data provider needs to be added, or a current data provider changes its data format, the multimodal API needs to be adjusted to it.

3.4.3 No uniformity/standard on roads, routes and public transport stops

By definition, a multimodal API needs to combine the data from different data sources into one uniform format. All these data sources model the same reality: the same roads, the same stops and the same vehicles. However, the data sources don't use or exchange unique identifiers to identify these objects between the multiple data sources.

- There are no unique identifiers for public transport stops. Or there are but they're not used in all data sources. For example OpenTripPlanner and OpenOV use the same id's, but 9292 uses different id's in their app and website. Based on the name of a stop and its geographic location we need to make a 'best guess' to match these stops from different data sources.

- There are no unique identifiers for public transport routes, or not used by all data sources. Imagine System A suggests to take Bus 123 leaving for 'Amstelveen' at Schiphol Airport at 08:00. To find out if its currently delayed, we need to search System B for a bus with the same departure time, place and headsign to check its current status. This bus is not identified with an unique id both System A and System B know.
- There are no unique identifiers for roads. Some systems provide a street name, some provide its geo location and some provide both.

In all cases mentioned above it is feasible to make a 'best guess' to combine the data of different data sources. However, this is prone to errors which may be overcome if the systems are using a shared set of unique identifiers.

4 What should be improved

4.1 Data needed

4.1.1 Road traffic delay

There is currently not enough data available to take the current road traffic situation into account for route planning and delay monitoring.

- The free version of Google Maps doesn't take the current traffic condition into account for route planning. Their paid services, Google Maps for Businesses (starting from $17.500/year), does.
- IVV provides some information on the current travel times, but this data has some limitations:
  - It only provides information on the most used travel routes in Amsterdam. So current delay can only be calculated if exactly those routes are travelled. If only a part of a route is used, the current delay is unknown.
  - No alternative routes can be given based on this information, because the traffic conditions of the other/adjacent streets are unknown.
  - The data is provided with geo locations, but these aren't exactly matched to the real roads. Mapping needs to be done to match it with the real roads, which is prone to errors. (The CitySDK project, which is in test phase right now, provides the IVV data as well and has solved this mapping issue.)
- Both Google and IVV don't provided any data on expected travel times.

Using commercial data

TomTom has detailed information on both the current situation and predicted travel times, but these need to be purchased[1]. The terms and conditions for this purchased data won't allow this data to be provided for free in an open multimodal API. To know exactly how this data can be used in a multimodal API would require further discussion with TomTom.

4.1.2 Parking information

There are currently no data sources on expected times for finding a parking spot, parking your car and finding a parking meter. Without this data, the multimodal API will often recommend a car-only route as the fastest route, not taking this extra parking time into account.

This data should be available for parking on the street as well as for parking in parking garages and P+R locations.

For the multimodal API to be realistic, this kind of parking information needs to be added. And it needs to be intelligent enough to know the difference between Buitenveldert and de Jordaan, and between Tuesday evening and Sunday morning. It should know that a specific P+R location tends to be full from 9:00 to 17:00.

[1] We've requested an price indication for this data at TomTom. At the time of publication these costs are still unknown.

4.1.3 Route and location information

There is currently no or only limited data[2] on the exact routes of public transport. All stops are known, but the routes between the stops are unknown. For busses this data would be helpful, especially when you want to predict travel times based on the current traffic.

There is no data on the exact location of public transport vehicles. It's known at what time they've left their last stop and it's known at what time they are expected at the next stop, but we don't know their location in between. Again, for predictions, this data will be helpful.

If this data is already available to the transport companies, it would be very helpful to open this data. If this data isn't available yet, it might be too complex and costly to collect this data in relation to the expected benefits.

4.1.4 Transfer hubs

In our proof of concept, we have introduced the concept of the 'Multimodal Transfer Hubs'. These are locations where travellers can easily switch between modes of transport. The reason for creating multimodal transfer hubs is to reflect the reality that although you could use almost any location to switch from car to public transport by using on street parking, it is not desirable for both the traveller and the city of Amsterdam to leave their cars all over the city and then switching on to the tram or bus network. The multimodal transfer hub is a specially created place for people to switch from one mode of transport. For example, P+R locations are multimodal transfer hubs, as well as all train stations, parking garages and bigger parking spots.

It would be very useful if IVV, or any other data provider, could provide a data set with all preferred multimodal transfer hubs.

4.2 Areas for improvement

4.2.1 Prediction

As we stated before intelligent prediction is of great importance for a well-functioning multimodal API. In its current state, the data provided by the data providers lacks this intelligent prediction. For public transport, as soon as any delay occurs, the accuracy of short-term predictions (less than two hours) is insufficient to provide valuable multimodal planning. For longer term prediction (two hours or more), no data is available.

This may result in a multimodal API that is too optimistic on travel times, both on road travel as for public transport. Taking the "Gap of Disappointment" into account, this will lead to unsatisfied users.

In paragraph 4.3 we will illustrate this with some real life examples. The current predictions turn out to be too naive for reliable planning in the reality of a big city like Amsterdam.

To solve the lack of intelligent prediction, there are mainly two options:
- Better prediction can be done by data providers themselves. The data providers should be aware of the real life examples of their predictions and should improve their algorithms to be more realistic.
- Better prediction can be left to market. In this case historical data on travel times should be available to the market for free. Creating a system that can provide these intelligent predictions is a complex

[2] The exact route of a public transport is optional in the GTFS-format. The GVB does provide this data, including bus lanes, but it is not real time (so regular and ad-hoc diversions are not included). Other companies, don't provide this data at all. For example EBS and HTM just draw direct lines between stops.

and costly task. However, for road traffic this is already be done by TomTom. For public transport a similar approach may be feasible.

4.2.2 Standardization of unique identifiers

On public roads, public transport stops, lines and vehicles unique identifiers should be used by all data providers. If all providers are using the same identifiers, that can be linked to each other, setting up a multimodal API would be less complex.

It would prevent errors that currently arise from the 'best guess'-algorithms to identify the same bus, road or route between two different systems.

4.2.3 Train arrival times

The NS API, which we have successfully used in our proof of concept, provides all planned and realized departure times of their trains. However, the expected *arrival* time is not available in the API. This causes inaccuracies in case a train arrives 5 minutes behind schedule, but is scheduled to wait at the destination station for 10 minutes. It will depart on time, but arrived late.

Adding train arrival times to the NS API would be a relevant improvement.

4.2.4 Parking

IVV provides a JSON-file with locations of parking garages, P+R and taxi stops. This is helpful information, but some improvements can be made:

- Information on opening hours and tariffs could be added.
- Information on whether there are parking spots for scooters or motor bikes could be added.
- For cross-referencing to other data sources, the address field should be split up in street, house number and suffix.
- (Links to) real time availability feeds could be added.
- Future prediction or historic data could be added.

4.2.5 Planning should take real time data into account

As we stated before in paragraph 3.3.4, route planners should take real time data into account. Of all available open route planners, only the NS is using real time information to calculate delays and suggesting alternatives for their own trains. All other planner use static data only.

Even the number of commercial route planners that combine planning and real time information is quite limited. TomTom does a great job on actual route planning on road travel, but this planner can't be used by third parties.

In paragraph 3.3.4 we already stated that this problem can be solved by the improving the existing route planners, or by building the 'route planning intelligence' by yourself or by a third party. The latter will probably require effort and money out of the scope of the multimodal API.

4.3 Real life examples

4.3.1 Delays for vehicle turnaround

More details: see https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Incorrect-delays-for-vehicle-turnaround

Trams and buses typically drive back and forth between the two endpoints on a public transport line until they are taken out of service. This means that if a vehicle builds up a long delay on a trip, the reverse trip for the same vehicle will most likely also be delayed. Although predictable, this second delay is currently not taken into account.

Delays for trips at stops close to the beginning of a line are now unknown until very briefly before the trip starts. So the status of a trip from Amsterdam CS to anywhere with a tram is uncertain until the time the tram leaves.

Example

- Tram A is running from IJburg to Amsterdam CS. It is planned to arrive at 12:15, and it's departure for the return trip, from Amsterdam CS to IJburg is planned at 12:17.
- Tram A suffers from a delay on its way to Amsterdam CS. Its expected (delayed) arrival time is 12:25.
- In the real time data, the departure of Tram A from Amsterdam CS to is still predicted at 12:17, although it is quite sure that this tram won't depart on time.

4.3.2 Delays not catching up

More details: https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Overoptimistic-catch-up-for-delays

When a public transport vehicle is delayed, GOVI expects that it will slowly catch up, due to extra room in the scheduling. The algorithm used is very simplistic: it assumes that between every two stops, the delay will decrease with 30% of the scheduled time between stops.

In the reality of Amsterdam, it's exactly the other way around. Particularly in rush hour on frequently running lines, a delayed trip may only get delayed more. The issue is that boarding becomes much slower as the vehicle becomes fuller - and there will be more passengers waiting at stops as the delay gets larger.

While the 'catching up algorithm' may apply to quieter services outside the city limits, this approach is too far from reality in Amsterdam.

Example

At 16:45, a tram line 1 between Nieuwezijdse Kolk and Dam is delayed 4 minutes. It is predicted that the tram will arrive on time, at 16:53, at the 1e Con. Huygensstraat.

At 16:54, nine minutes later, the tram is expected to arrive 4 minutes late at the 1e Con. Huygensstraat.

4.3.3 Stops and stopareas

More details: https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Available-data#wiki-Stops_and_stopareas

A stop in OV data context is a single stop in a single location. This means that a stop as seen by travellers typically consists of two actual stops: one for each direction. This can become even more complex in transfers: Station Lelylaan for example, has two metro stops, two tram stops, and one bus stop (which is used for both directions).

There are two ways of translating between a single stop and the series of stops a traveller is referring to: matching by name, and grouping into a stoparea. In the example of Lelyaan, all these stops have the same name, making the matching simple. However, this is incomplete, as sometimes multiple stops with the same name are actually quite far away from each other (For example, “Keizersgracht” can refer to “Keizersgracht/Vijzelstraat” or “Keizersgracht/Leidsestraat” a 6 minute walk difference). So this needs to be combined with checking how close they are to each other. In addition, stops at crossings may have different names, like Bilderdijkstraat for tram 17 and Kinkerstraat for tram 12. These stops are very close and commonly used to transfer, but have different names.

The more reliable method is using a stoparea. This is a group of stops, defined by the data owner in the static planning data, which would usually be considered one stop by a traveller. For example, Connexxion has the stoparea "_schns_", which is actually 9 different stops, all at Schiphol Centrum (NS station). Unfortunately, the GVB static planning data does not include these areas. In addition, this might not work across transport companies, as they could be using different stoparea codes.

A method of last resort might be to match them on geographical proximity only. However, this might lead to false positives.

4.3.4 Known complex situation around Amsterdam CS

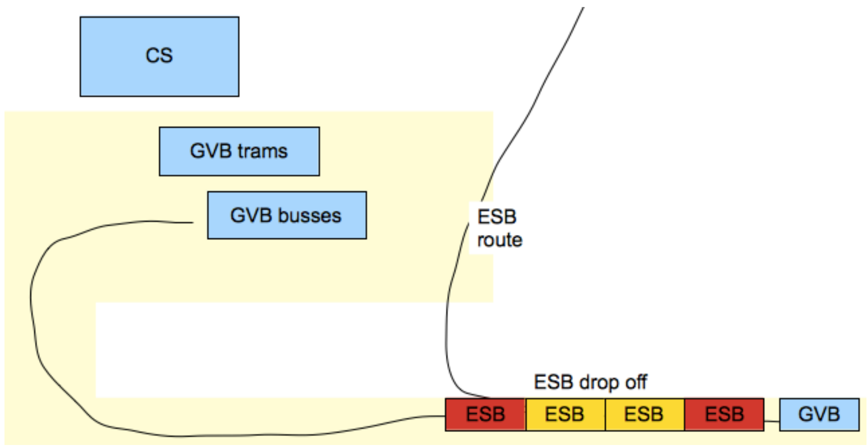
More details: https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/The-bus-drop-off-issue-in-front-of-cs-which-highly-affects-the-data

At Amsterdam Central Station, there is a temporary drop-off for EBS busses at the Prins Hendrikkade. While the EBS busses are dropping off passengers, the busses queue up and the GVB busses suffer minutes of delay.
The problem is that everybody who frequently takes one of these bus lines is aware of this delay, but it is not taken into account in the scheduled or predicted arrival times. Leading to route suggestions that aren't feasible in reality.

Example

	
At Johan v. Hasseltweg, bus line 33 to Central Station has a delay of 3 minutes. However, the bus is expected to arrive at Central Station on time (8:50).	At 8:52 the bus is still queued up behind 6 EBS busses, which have a temporary drop-off point just before the bus stop for GVB. (See figure below) So all passengers in the GVB bus know they will arrive at least 5 minutes late at their destination, but the real time data is still predicting an

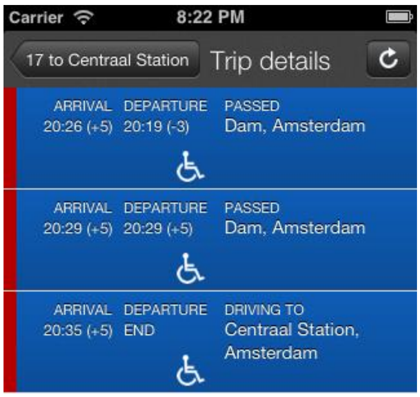
	immediate drop-off. Each minute the real time data adds one minute to the arrival time.



This is a known situation and causes delay day after day. Even if this is due to temporary construction works, this should be part of the schedule or predictions in the real time data. Without doing this, the real time predictions are too optimistic and transfers are suggested that prove to be impossible.

4.3.5 The time travelling tram

More details: https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Timetravel



According to the real time data, tram 17 to Centraal Station arrived at the Dam at 20:26. Magically, the departure time for that same tram was at 20:19.

This is a known problem at GOVI, and they are working on resolving this. Since GOVI does take data quality very seriously, we expect this to be resolved soon.

4.3.6 More examples in the WIKI

In the WIKI at https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report/wiki/Examples-of-incorrect-data there are seven more real life examples to be found.

5 Towards a multimodal API

5.1 Current status

The table below summarizes the current status on the multimodal API. (Appendix A “Required & available data” provides a more detailed overview)
For static planning all data is available, on real time planning and monitoring improvements are needed. The ultimate multimodal API is not possible at the moment.

	Planning		Monitoring	Ultimate API
	Static	Real time		
Public transport	OK	Imp.	Imp.	X
Motorized traffic	OK	Imp.[1]	Imp.[1]	X
Bike	OK	Room for Imp.	Room for Imp.	X
Walking	OK	Room for Imp.	Room for Imp.	Imp.

OK	OK, ready to use
Room for Imp.	Room for improvement
Imp.	Improvements needed
X	Not possible at the moment

[1] Only in the unexpected case that commercial data (for example from TomTom) can be used in an open multimodal API, this would be an ‘OK’.

5.2 Scenarios towards a multimodal API

5.2.1 Option 1. Provide the current proof of concept AS IS to developers to further build upon

This scenario is the end result of the current proof of concept. All source code and documentation is freely available on GitHub. GitHub is a platform for developers to jointly work on software, both open source and closed software.

A next step for this approach would be to set-up some kind of communication or event to encourage developers to use and improve the proof of concept into a real API, apps or other applications.

5.2.2 Option 2. Provide a real API based on the current functionality

Without adding any functionality to the suggested functionality, and realising only the easy, feasible improvements suggested in this document, the current proof of concept may be ‘upgraded’ to a real API.

As we have concluded that the current functionality is not an ultimate multimodal API, we have to realise that it still provides added value to the current situation. Providing a helpful multimodal API that cover the first steps towards the ultimate multimodal API is a viable option for DIVV.

5.2.3 Option 3. Best effort approach

The third and last viable option is to take a best effort approach to building a multimodal API. This includes realizing all improvements suggested in this document as far as possible, and adding as much data as data providers can be stimulated to provide.

In this approach a planning algorithm will be realised that takes real time information into account. And based on available historic data a predictions algorithm will be developed to calculate expected travel times based on time and day of week.

This approach will bring the multimodal API to a promising next level, but it won't be the ultimate multimodal API. The accuracy in complex traffic conditions will still be low (see: gap of disappointment) and a social component (agenda matching, social interaction, etc.) will be lacking.

5.2.4 Option 4 – The ultimate multimodal API

As we have concluded before, the ultimate multimodal API can't be built at this moment.
A first requirement would be to get all the needed data we have mentioned in this document and realize all suggested improvements. But even then it is too complex at this moment to calculate or estimate what it would cost to provide accurate predictions at complex situations. This situation is simply too far from where we are now.

5.3 Indication of related costs

Based on the options in the previous paragraph, we can provide a very high level indication of related costs for the first three options. These figures are only an indication, a rough estimate. To be more exact, more research would be required.

Scenario	Development	Hosting	Maintenance/year
Option 1 – AS IS	-	n/a	n/a
Option 2 – Current functionality	50 – 100k	*[1]	25% of dev.
Option 3 – Intermediate solution	100 – 500k	*[1]	25% of dev.

Hosting costs

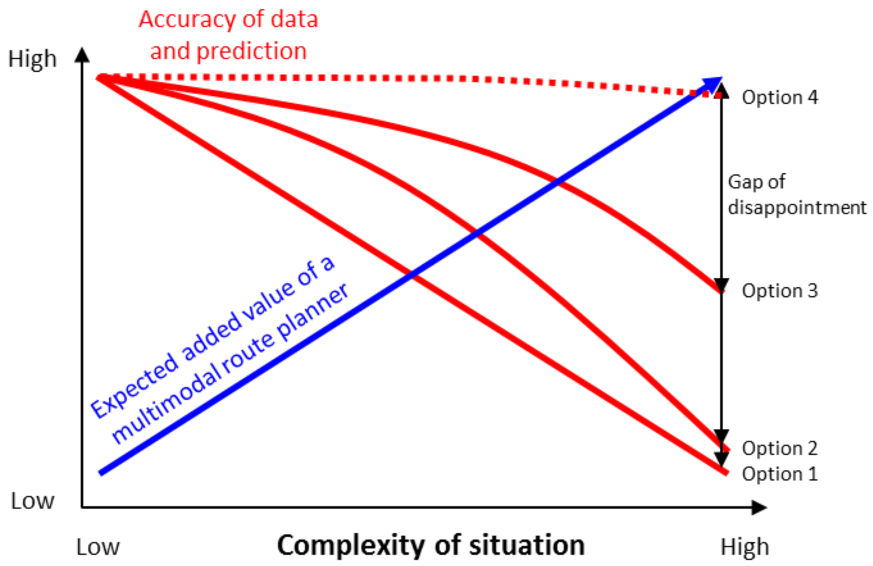
*[1] The required cost for hosting a multimodal API are very hard to predict, and would require more research, both on the expected technical structure of the API and the expected use of it.
If only a development environment is needed, in which case heavy users are expected to run their own hardware, these cost may be quite low (starting from 10k/year). If high performance hosting is needed, to provide decent service levels to heavy users, even in extreme (traffic) conditions, hosting costs may meet or exceed development costs.

Gap of disappointment

The 'gap of disappointment', as we have discussed in paragraph 3.4.1 has to be taken into account in all options. Only in case of an ultimate modal API, the gap of disappointment would be non-existent, as the API will provide accurate data and predictions in all cases, including the most complex situations.

In option 2 and option 3 the accuracy of data and predictions will increase, but still a gap of disappointment will be existing in complex situations.

The figure below shows a schematic view of how the 'gap of disappointment' will change based on the selected option.



5.4 *Next steps*

Based on everything described in this document, the following next steps can be identified:

5.4.1 Improving and extending available data sources of third parties

In this document we have suggested what data is needed to open and how available data can be improved. IVV can play an important role on this.

Next steps

- Contact and motivate data providers to provide the needed data that is still missing.
- Contact and motivate data providers to realize the improvements suggested.
- Include specific data requirements in future concessions to transport companies and other contractors.

5.4.2 Providing data, source code and/or a multimodal API

According to IVV's motto "We provide the data, you build the apps (*Wij de data, u de apps*)", providing open data to developers is key to IVV. It is important to IVV to have a clear vision to in what format this data should be provided, and what role it should and shouldn't play towards developers.

Whether or not IVV, or local government in general, should provide a multimodal API to developers, is a political question which we can't answer. We are, however, quite sure that:

a) IVV itself is not the right party to build and maintain a multimodal API (from a technical perspective)
b) IVV itself shouldn't be a solitaire project owner/contractee to building the ultimate multimodal API (from a national perspective)
c) IVV can play an important role as 'k*wartiermaker*', facilitator and/or sponsor towards the realization of the multimodal API.

Next steps should be taken to investigate what the best approach is to facilitate or sponsor an initiative to realise the multimodal API. In other words, the 'Open Data Game' should be played, creating a setting where all stakeholders are involved and defining demand & supply.

For example, it might be worth investigating if a '*stichting*' can be set up with a target to build and maintain a multimodal API.

Next steps

- Define what data IVV can open and/or improve itself (based on this document)
- Provide this data with clear metadata and documentation
- Investigate what is the best way for IVV to facilitate or sponsor the realization of a multimodal API (Playing the 'Open Data Game').
    - Define demand and supply of involved stakeholders
    - Measure the excitement/interest for developers to build upon the multimodal API
- Decide what role IVV should play in the realization of the multimodal API
    - 'Off the field', as a supporter/motivator
    - 'Along the line', as a facilitator/sponsor
    - 'On the field', as a key player/commissioning authority

5.4.3 Encourage development of applications

Whatever role IVV is playing in the development of the multimodal API, it is important that IVV encourages developers to use the data or API they provide to them. The so called 'warm approach' to open data provisioning, to involve developers in the process of providing data, into the challenges of the city and the opportunities that exist.

- Encourage/motivate developers to use the source code or API provided
- Encourage/motivate developers to develop multimodal apps
- Involve Appsterdam and/or other communities of developers in these activities
- Organise a 'IVV Open Data Café'-like event targeted on developers

Appendix A – Required & available data

In the table below we have listed all information on modes of transport, its current availability and whether this is needed for the use case scenarios we have defined.

Scenarios

- A. **Frequent traveller**
- B. **The basic A -> B planner**
- C. **The Amsterdam Visitor**
- D. **The new tourist**
- E. **The destination planner (A -> B -> C -> Z)**

			Data needed for scenarios				
Mode of transport	**Type of info**	**Data availability**	**A**	**B**	**C**	**D**	**E**
All public Transport	Haltes	OK	R	R	R	R	R
	Routes	OK	R	R	R	R	R
	Costs	X	-	R	R	R	R
	Planning/navigation	Imp.	O	R	R	R	R
	Delay	Imp.	R	R/O	R/O	O	R/O
	Location	X	-	-	-	-	-
Motorized traffic	Planning/navigation	OK	O	R	R	-	R
	Delay	Imp.	R	R/O	R/O	-	R/O
	Costs	X	-	R	R	-	R
	Parking	Imp.	-	R	R	-	R
	Rental	X	O	O	O	-	O
Bike	Planning/navigation	OK	O	R	R	-	R
	Delay	Room for Imp.	O	R/O	R/O	-	R/O
	Parking	OK	-	O	O	-	O
	Rental	X	O	O	O	-	O
Walking	Planning/navigation	OK	O	R	R	R	R
	Delay	Room for Imp.	-	R/O	R/O	-	R/O
Taxi	Planning/navigation	OK	O	R	R	R	R
	Location	X	-	-	-	R	-
	Delay	Imp.	R	R/O	R/O	-	R/O
	Costs	OK	O	R	R	R	R
	Availability	X	O	R/O	R/O	O	R/O
Points of Interest	Location	OK	-	O	O	R	O
Transport hubs		OK	-	O	O	-	O
Tourist information		OK	-	-	-	R	-

Data availability	
OK	OK, ready to use
Room for Imp.	Room for improvement
Imp.	Improvements needed
X	Not possible at the moment

Data needed for scenarios	
R	Required
R/O	Required, but no show stopper
O	Optional
-	Not needed

Appendix B – Digital Footprint

In the process of setting up the proof of concept for a multimodal API, we left a digital footprint which is available at github: https://github.com/DIVV-Amsterdam/DIVV-multi-modal-api-report

The Github consists of two main parts: the source code and the WIKI-documentation. The source code can be used to see what we did, and build upon yourself. The WIKI documents what we did, why we did it and how we did it.

Below is the table of contents for the WIKI, which is a great resource for developers for building a multimodal app.

Op weg naar de multimodale API

Hoe IVV open data beschikbaar kan stellen om ontwikkelaars de multimodale app te laten ontwikkelen

Al geruime tijd heeft de gemeentelijke politieke bijzondere aandacht voor de *multimodale app.* Een alwetende routeplanner die alle vervoermiddelen met elkaar combineert, en steeds aan de hand van de actuele gegevens de juiste routes en vervoermiddelen adviseert. Van ochtendspits naar P+R, van tram naar OV-fiets.

Sinds DIVV ruim een jaar geleden haar motto "Wij de data, u de apps" aankondigde is er weliswaar veel open data beschikbaar gekomen, maar zijn er vanuit de markt nog geen sterke initiatieven in de richting van een multimodale app ontstaan. Daarom heeft DIVV aan een aantal experts gevraagd om aan de hand van een proof of concept vast te stellen of een multimodale API deze ontwikkeling kan versnellen. Door middel van zo'n multimodale API worden alle beschikbare databronnen in één uniforme interface aan ontwikkelaars beschikbaar gesteld, waardoor zij eenvoudiger toepassingen kunnen ontwikkelen.

Uit deze proof of concept blijkt dat door een beperkt aantal open databronnen te combineren, het op dit moment al mogelijk is om een eerste concept van een multimodale routeplanner via een API aan te bieden. Aan de hand van een start- en eindpunt en de gewenste datum en vertrektijd, worden alle mogelijke multimodale routes berekend en in volgorde van totale reistijd teruggegeven. Ook kan hierbij de actuele situatie weergegeven worden, inclusief eventuele vertraging.

De routeplanning vooraf en de actuele informatie zijn door de beperkingen van de databronnen strikt gescheiden. De routeplanner kan voor vertrek gebruikt worden en signaleert evt. vertragingen tijdens de reis, maar kan op dat moment geen alternatieven suggereren. Ook het ontbreken van informatie (bij. over files en beschikbare parkeergelegenheid) en de onnauwkeurigheid van bestaande informatie (bijv. voorspellingen van het OV) beperken de huidige bruikbaarheid van de multimodale API.

De resultaten van de proof of concept wijzen wel uit dat een multimodale API een belangrijke volgende stap kan zijn in de ontwikkeling van de multimodale app. Daarnaast wordt ook vastgesteld dat er nog een lange weg is te gaan is op weg naar de ultieme multimodale API, die onder *alle* omstandigheden de reiziger van de juiste reisinformatie kan voorzien.

Een belangrijk gegeven hierbij is dat hoe complexer een verkeerssituatie wordt (slecht weer, afgesloten stations, drukke wegen, etc.), hoe hoger de verwachtingen zijn van gebruikers van een multimodale app. Maar juist in deze complexe situaties is het zeer moeilijk om goede en betrouwbare adviezen te geven. Op dit moment schiet de beschikbare data op een aantal belangrijke aspecten te kort:

- De korte termijnvoorspellingen (minder dan een uur vooruit) zijn nog niet voldoende betrouwbaar om goede reisadviezen te geven.
- Lange termijnvoorspellingen (van een uur tot weken vooruit) zijn essentieel voor goede reisadviezen, maar zijn in het geheel nog niet beschikbaar.
- Er is nog geen goede integratie tussen reisplanners en de actuele gegevens. De huidige open routeplanners nemen de actuele situatie nog niet mee in hun reisadviezen.

In het rapport worden deze en andere verbeterpunten uitgebreid toegelicht en voorzien van 'real life' voorbeelden. Daarbij worden concrete aanbevelingen gedaan welke stappen IVV en andere dataleveranciers kunnen doen om dichter bij het ideaal van de multimodale reisplanner te komen.