

## Introduction

### Problem Description

Traffic analysis is an import part in road network and mobile network simulations. These simulators require moving subscribers in order to investigate interesting effects in their networks. Moving subscribers can be generated from mobility models which try to describe the mobility of individuals. This work focuses on estimating trajectories for mobility simulations by using mobile subscription data. A trajectory describes the path of a moving object through space as a function of time. Network operators need moving subscribers to evaluate and analyze an existing or a virtual network. Replay scenarios can be created to investigate the current or a new network with a scenario where errors accrued. The trajectory generation process involves several tasks. First useful events have to be filtered in the network. The users path has to be estimated, which involves the start and end position as well as the handover position. Another part of this thesis is to estimate the coverage area for each cell sites. The coverage area is a crucial aspect in trajectory estimation. Trajectories need the location of a subscriber as a function of time. However GSM doesn't expose an accurate position of the subscriber. Therefore a good representation of the coverage area allows to narrow the users position in the network. The problem is to find an approach which creates a good estimation of the coverage area for each cell site.

### Motivation

In the last years mobile network simulations have to adopt to the mobility of the subscribers. Subscribers are not stationary and therefore behavior models have to be defined. These models shall represent the entire subscriber database. Since the beginning of mobile network simulations random walk and manhattan grid approaches were used to enable mobility. The problem is that random walk and manhattan grid rely on statistical data. The statistical input used is derived from household surveys. This surveys however only depict a special moment in time. To evaluate or gather more moments numerous surveys have to be done. Our approach is different, instead of surveys we are using mobile subscriber data. More precise call data records. The motivation behind using mobile subscriber data is that this data represents all users of the mobile network operator. We want to use this data to generate anonymous driving trajectories. The use of call data records allows the generation of trajectories for each day and time of the year. The generated trajectories adopt to daily as well seasonal changes.

### Challenges

The first challenge we have to face is that we don't have a model of the coverage area for all cell sites. Therefor we need a good representation to estimate a coarse position for each subscriber. In order to generate a trajectory the users start and end position are crucial. At first we only have the serving cell and a representation of its serving area. Therefore the position of the user within the cell has to be estimated. For this purpose we are using a combination of population and land use information data.

### Goals

The main goal is to generate trajectories for each mobile subscriber. However due to the design of GSM we can only generate trajectories for mobile users which are in an active call. Additionally we are investigating approaches to annotate the driving trajectories with timing information. In the end we are also examining methods to improve the representation of the serving area with public available data (digital elevation model, land use clutter information,...).

## **Structure**

At first we are discussing state of the art approaches to estimate driving trajectories and mobility in mobile network simulations. Followed by an overview of fundamental concepts and techniques used in this thesis. The third part explains our concept and approach which will be implemented in the succeeding chapter. At last we will present and discuss our results and give a summary of the work.

## **Related Work**

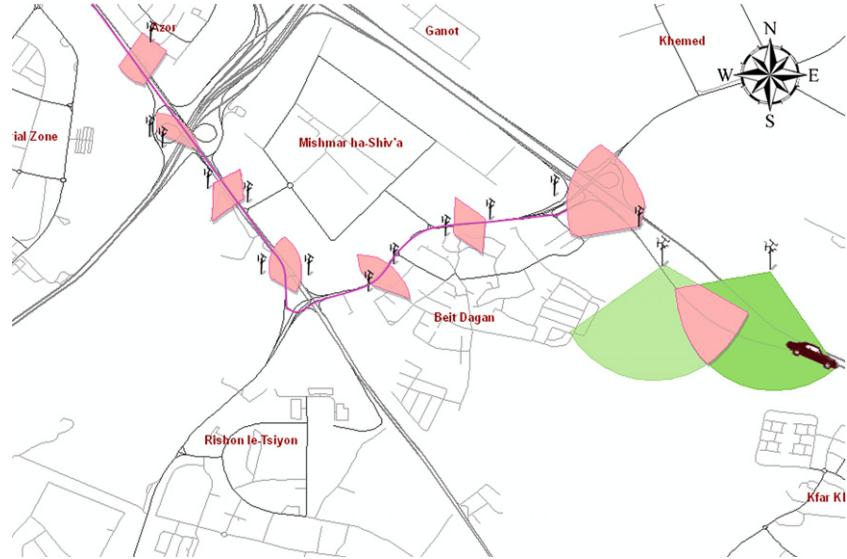
The following chapter presents state-of-the-art approaches and projects related to user mobility and traffic analysis. These projects are used to derive the mobility of mobile subscribers by investigating events in mobile operator networks. They differ from classical behavior analysis in terms of penetration and accuracy. To derive the behavior of the entire population surveys and origin-destination matrices have been used in traffic modeling since the early 1960s. Starting in the late 1990s floating phone data (FPD) has gained interest in traffic estimation and congestion detection. FPD is used due to its high penetration rate, in Austria the market penetration was 159%<sup>1</sup> in 2013. There is a large volume of published studies describing the role of FDP for traffic analysis [@Yim2001][@Qiu2007][@Caceres2008]. The approaches that will be presented here have a higher penetration rate due to the fact that they investigate a whole network whereas surveys can only cover parts of the entire network. Another advantage of these approaches is the adaption to seasonal changes. Surveys only depict the behavior at a certain point in time. However both road and mobile network traffic modeling depend on changes in time. By exploring events in mobile operator networks, the behavior can be analyzed for every point in time.

### **Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel**

To measure the traffic speed and travel time in Israel Bar-Gera [@Bar2007] used a proprietary system by Estimation Ltd. The system works with handover updates to derive the traveled route and traffic speeds. A sequence of locations derived from the handover footprint is matched to road segments which appear to be the most likely on the road network. An example of a handover sequence and its footprints on the road network is shown in Fig. [fig:bar]. Despite the fact that this is a proprietary system it gives a good overview of the capabilities of floating phone data. The work by Bar-Gera shows that travel time estimation with phone data can be a replacement for loop magnetic detectors. He investigated a road segment from January to March 2005 and found that the average absolute relative difference between the two systems is 10.7%. However the whole system has its limitations due to the noise generated by floating phone data. They could reduce the noise by combining travel speeds from subscribers that traveled on the same road segment.

---

<sup>1</sup>Press conference by the Forum Mobilkommunikation [press conference link](#)



[fig:bar]

## Generating Trajectories from Mobile Phone Data

To derive trajectories from mobile subscribers Schlaich et al. [@Schlaich2010a] used location area sequences. Location area update events are issued whenever a mobile subscriber enters a new location area. These events are issued in both connected and idle states. Therefore trajectories can be generated even when no call is ongoing. Due to the small number of location areas in the research area, each location area was represented by a unique character. This allowed storing a location area sequence as a string. The next step was to generate routes in the research area and store the location area characters for each area they traversed. To estimate a trajectory for a mobile subscriber the location area sequence for the subscriber was compared to route sequences that have been generated before. Fig. [fig:schlaichcomp] depicts the comparison of a mobile subscriber sequence and route sequences with high similarity.

A-Data-LAC-sequence “zqnlnjlstscbdhif”	Network routes and resulting network-LAC-sequence
	 “zmqnlnjls cbdhif” similarity: 80 %
	 “zmqnlnjls cbdhif” similarity: 80 %
	 “bcs” similarity: -105 %
	271,000 other network-routes similarities < 80 %

[fig:schlaichcomp]

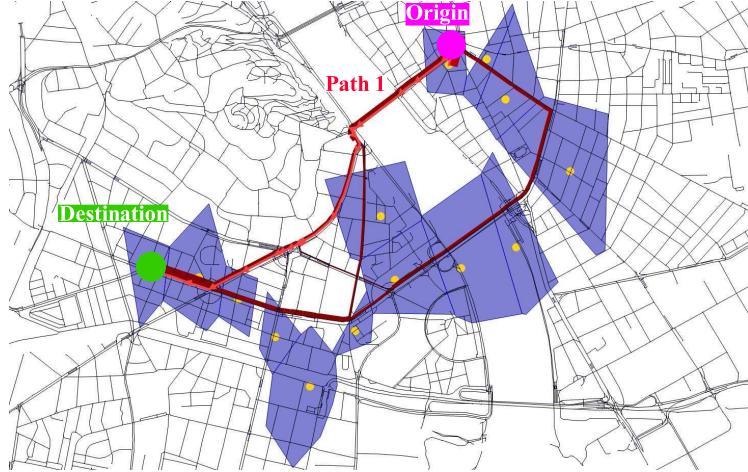
The results showed that this technique works well for longer trips around 20 kilometers because trips can only be estimated if the number of different location areas is greater than 3. Schlaich et al. also stated that the presented approach only generates trajectories for sim cards and not for vehicles. A vehicle can have none, one or more sim cards therefore a vehicle can generate  $0..n$  trajectories where  $n$  is the number of sim cards within the vehicle. Schlaich [@Schlaich2010] later used this technique to observe the route choice behavior in the Stuttgart, Germany area. It has shown that drivers react to traffic news broadcasts and variable message signs (VMS). However the acceptance of a route displayed on a VMS was only between 3 and 17%.

## Route Choice Estimation Based on Cellular Signaling Data

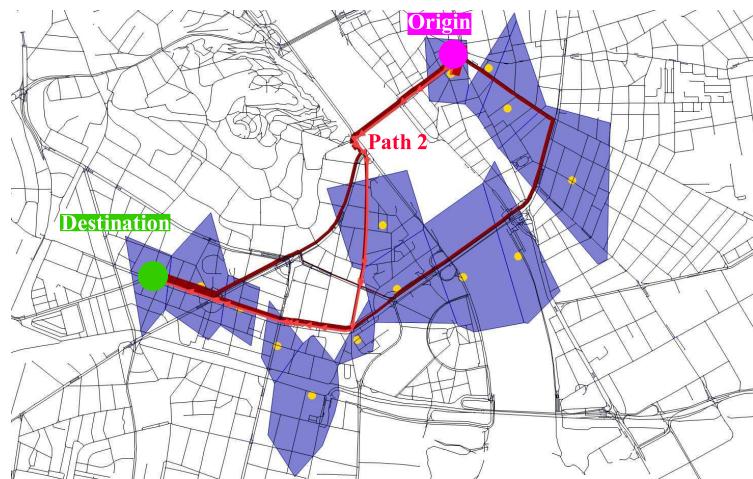
To overcome the use of origin-destination matrices Tettamanti et al. [@Tettamanti2012] used a simulation framework and a route generator to generate trajectories for mobile subscribers. Instead of using location area updates as Schlaich et al. did they used handover updates. Handover updates allow to generate trajectories not only for a higher road network but also for the minor one. Their approach is based on cell area estimation as each handover update reveals the current cell area. Whenever a mobile phone reaches the boundary of the currently connected cell or when another cell has a higher reception then a handover is made to a cell with a better reception. Tettamanti et al. used Voronoi partitioning to calculate the coverage area for each cell site to estimate a coarse user position. The main limitation when using handover updates is that a handover update is only propagated when the mobile phone is in connected mode (during a call). The start and end of the trajectory were set by the centroid of the cell where the call originated and terminated. To derive routes, a traffic modeling simulation framework *VISSIM* was used. VISSIM was used to generate routes between the start and the end of the call. For each of the generated routes, the euclidean distance between each of the cell sites and the generated route was calculated. To compare different routes equation [eq:sumsquare] was used as a metric. For each route  $j$ , the squared sum of all minimum distances  $d_{i,j}$  between the route and the cell site was calculated. The route with the minimum sum was used as a trajectory for the subscriber. In Fig. [fig:tettaroutes] an example for four different routes which were generated by VISSIM can be seen. Tab. [tab:tetta] depicts the results of the above stated equations (see Equation. [eq:sumsquare]). In this example route 4 (see Fig. [fig:tettaroutes](#)) had the smallest squared sum of all four routes.

$$D_j = \sum_{i=1}^m d_{i,j}^2 \quad (1)$$

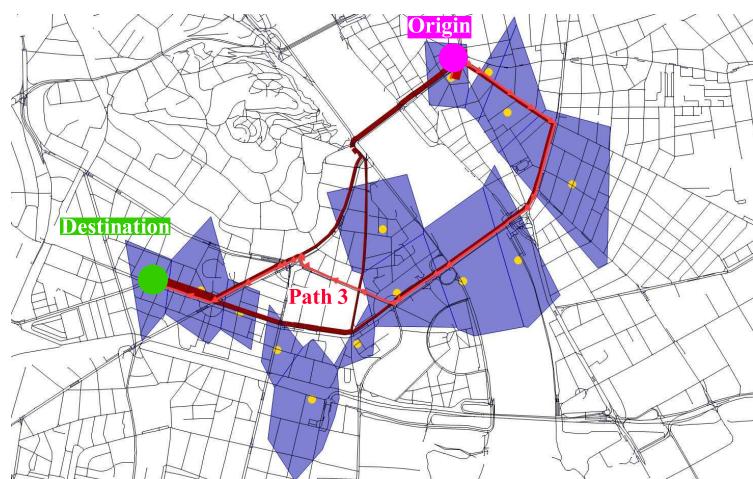
$$\min(D_j), j = 1, 2, \dots, n \quad (2)$$



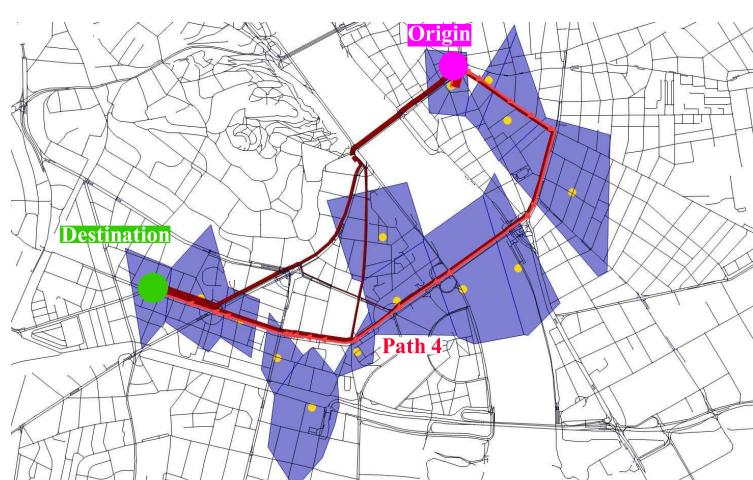
&



(a) first & (b) second



&



(c) third & (d) fourth  
fig:tettaroutes

[h]

Route $j$	$D_j = \sum_{i=1}^m d_{i,j}^2$	Ratio compared to the lowest value
1	5130283	8.2
2	3097741	4.9
3	1356076	2.2
<b>4</b>	<b>627122</b>	1

[tab:tetta]

The research by Tettamanti et al. showed that trajectories can be generated for mobile subscribers by analyzing their handover events. This can be done without using special equipment and can be used to investigate the travel behavior for many subscribers.

## RoadCell – Road Traffic Estimation from Cellular Network Signaling

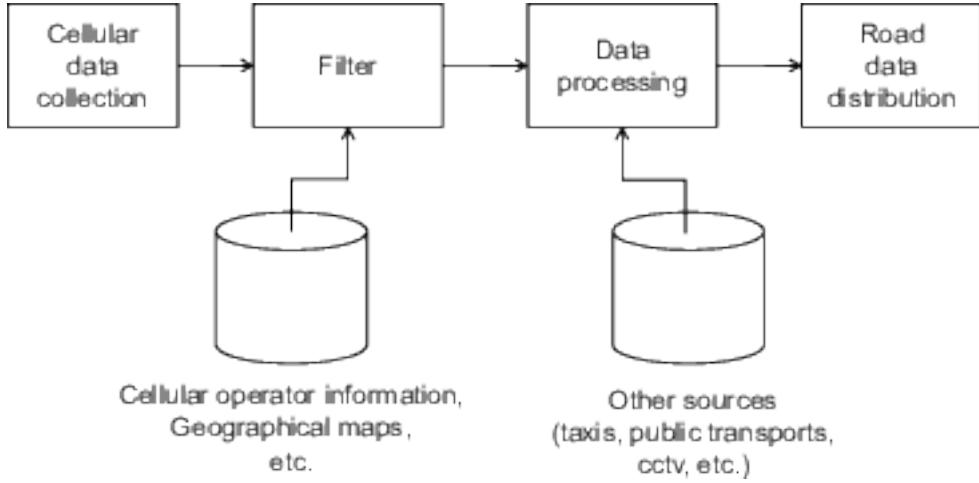
By using cellular network monitoring and therefore mobile subscription data Valerio et al. [@Valerio2009][@Valerio2009] at the FTW Forschungszentrum Telekommunikation Wien GmbH are investigating road networks in Austria. *RoadCell*<sup>2</sup> is using the same mobile subscription data source as our system [@RoadCell2009]. The aim of *RoadCell* is to recognize situations in the traffic flow by analyzing events in the core network of the mobile network operator. Driven by the fact that each road user is also a subscriber of a cellular network, network operator can be seen as another source to gather traffic information. Their motivation is to provide road operators with an inexpensive toolkit to observe the traffic flow without the installment of costly sensors. Thereby it is possible to not only investigate the traffic flow on the higher road network but also on minor roads where the installment of dedicated sensors cost-effective. The idea is to observe changes in network signaling events and extract road conditions for example: drop in the handover rate; abrupt change in the LR update; (c) increase in the number of calls/SMS; change in the number of road users. Figure [fig:raodcellaccident] shows how an accident effects the amount of Routing Area Update (RAU) and Location Area Update (LAU). An accident is indicated by sharp decrease in RAU and LAU events followed by a sharp increase in RAU and LAU events.



[fig:raodcellaccident]

<sup>2</sup>Project website: <http://www.ftw.at/forschung-innovation/projekte/roadcell> last accessed March 12 2014

Figure [fig:roadcell] depicts the system overview of the *RoadCell* project. It can be seen that, despite cellular network data, there are additionally data sources used. In order to map a particular event to a road segment, they are using coverage predictions from the network operator. To support data processing they are using third-party sources like floating car data from taxis and public transports which are more accurate, but have a lower penetration rate.



[fig:roadcell]

During their research found out that a combination of passive and active tracking is best to observe road conditions. Passive tracking is done without the involvement of the mobile station. This approach only utilizes the events captured in the operators core network. While passive tracking offers a high penetration and no increase in network load, it is less accurate than active tracking. Active tracking requires the mobile station and core network to exchange special events that allow the network operator to get an accurate position of the mobile station. This technique is known as Time Difference of Arrival (TDOA). Valerios proposal is to use a combination of both; passive tracking is always used and once the system detects a possible accident active tracking is used for the area in which the accident occurred. This approach improves the accuracy while trying to minimize the additional network load introduced by active tracking.

## Fundamentals

This chapter covers fundamentals that are necessary for a better understanding of the concepts presented in this work. It gives a short introduction into mobile communications systems which our system uses. Mobile communications systems are used in our system to estimate the users coarse location. Furthermore, state of the art analytical mobility models for mobile network simulations are presented. Another section covers the data (provided by A1) our system is using to generate driving trajectories. At last, the process to estimate the coverage area for cell sites with Voronoi tessellation is shown.

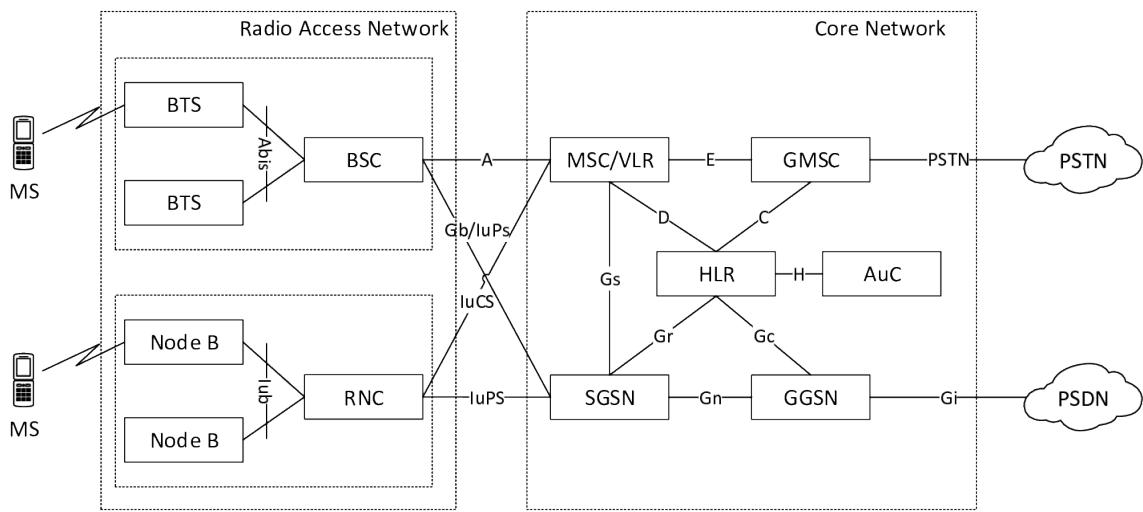
### Mobile Communication Networks - GSM

Mobile communication networks are the basis of our modern communication and interaction. However the beginning of wireless communication began in the late and was shaped by the work of Hughes, Maxwell, Hertz, Tesla and Faraday. In 1982, a new mobile communication network was developed by the Group Spécial Mobile of the CEPT (Conférence Européenne des Administrations des Postes et des Télécommunications). The main goal of GSM was to develop a Europe-wide standard for digital mobile communication. Over the past two centuries, there has been a rapid increase in the use of GSM all over the world. The present section is based on the books *GSM: Switching, Services and Protocols* by Jörg Eberspächer et al. [@Eberspaecher2001] and *Mobilfunknetze und ihre Protokolle 1* by Bernhard Walke [@Walke2001]. Besides GSM, there are other mobile communication currently in use. These systems

are known as 3G (UMTS, HSPA, WCDMA, etc.) and 4G (LTE, LTE Advanced, etc.) networks whereas GSM is a network of the second generation (2G).

## Components - System architecture

GSM consists of a large number of components which are necessary for its operation. However, we will only focus on components that are necessary to understand the functionality of our system. The overview of a GSM network with GPRS capability is visible in Fig. [fig:GSMNetwork]. The various standardized interfaces which are responsible for communication between the components can be seen as well as the different components of the *Radio Access Network RAN* and the *Core Network CN*. The RAN consists of a BTS and BSC in GSM and a Node B and a RNC in GPRS.



[fig:GSMNetwork]

**Mobile Station - MS** Mobile stations are the equipment used by mobile subscribers to access services provided by the network operator. The mobile station consist of two components, first the *Mobile Equipment* and second the *Subscriber Identity Module* (SIM). The SIM grants a mobile subscriber access to the network and allows to initiate or receive calls. For management purposes GSM assigns the following numbers and identities to a MS:

- **IMSI** International Mobile Station Identity
- **TMSI** Temporary Mobile Station Identity
- **MSISDN** Mobile Station International ISDN Number
- **MSRN** Mobile Station Roaming Number

**Base Transceiver Station - BTS** The BTS provides the radio channels for signaling and user traffic data with in a cell. The mobile station establishes a connection with the BTS over the air interface. The BTS consists only of a few parts, the high-frequency equipment (transceiver and receiver) and the signaling and protocol processing unit. The control remains in the BSC.

**Base Station Controller - BSC** A BSC is used to manage one or more BTS. It is directly connected to a MSC. Together with the BTS, the BSC forms the *Base Station Subsystem* (BSS). For example, the BSC executes the handover protocol and switches a MS to a new BTS.

**Mobile Switching Center - MSC** The MSC together with the databases (HLR, VLR) forms the *Mobile Switching System* (MSS). The MSC is responsible for the switching in the network. For example, the MSC performs signal routing, routing path search and service processing. Additionally the MSC has to pay attention to the allocation and administration of radio channel and the mobility of subscribers. A *Public Land Mobile Network* (PLMN) can consist of several MSCs where each is responsible for a dedicated area. Furthermore, the MSC is connected to a *Gateway Mobile Switching Center* (GMSC) which forwards voice traffic between fixed and mobile networks. The GMSC enables the subscriber to call subscribers in different MSCs or PLMNs.

**Visited Location Register - VLR** The VLR stores information of all MS who are located in the serving area of the associated MSC. A VLR can be assigned to one or more MSCs. A MS can be registered in a VLR of its home PLMN or a foreign one when a roaming agreement exists.

**Home Location Register - HLR** Typically, there is one HLR per PLMN and one VLR for each MSC. The HLR stores all permanent and temporary subscriber information for all registered subscribers. Moreover, it stores a coarse current location of each MS. The HLR operates as a central register needed by the MSC for routing of subscribers.

## Radio Resource Management - RR

The GSM standard [@Etsi1994] distinguishes between two modes of operation first idle and second connected mode. The radio resource management is necessary to manage the physical and logical channels.

**Idle Mode** In idle mode there is no dedicated channel assigned to a MS. However, the MS listens on signaling channels (BCCH and CCCH). The higher layers are only informed when a MS reaches the boundaries of a location area.

**Connected Mode** In this mode there are two dedicated channels assigned to the MS, the first one is SACCH and the second one is either FACCH or SDCCH. The radio resource management provides the following service to a connected MS (a full list can be found in ETSI GSM 04.08 [@Etsi1994]):

- Transfer of messages on any data link layer
- Establishment/release of multiframe mode on SDCCH, FACCH or SACCH.
- Automatic cell reselection and handover to maintain the RR-connection

## Handover

A handover is the transfer of an established connection to a new BTS. A handover is necessary for various reasons. First of all, a handover decision is made by the GSM network and not by the MS. As mentioned before the handover protocol is implemented in the BSS. The BSS decides to initiate a handover based on BSS criteria (channel quality, received signal level, distance between MS and BTS) and network criteria (e.g. traffic load of the network). Handovers are only performed when the MS is in connected mode. If the MS is in idle mode and reaches the boundaries of the cell a location area update is performed if the new cell is in a different location area. The GSM standard does not define an algorithm for handover decision. Therefore, network suppliers are responsible for implement them. A basic handover algorithm is specified in appendix A of the ETSI GMS 05.08 [@Etsi19942].

**Intra-Cell-Handover** The channel within the cell will be changed, e.g. when the channel is noisy. The change can either be done to a new frequency or to a new time-slot with the same (old) frequency.

**Inter-Cell/Intra-BSC-Handover** The channel will be changed between two cells within the same BSC.

**Intra-Cell/Intra-MSC-Handover** The connection will be changed between two cells in different BSCs, but both are managed by the same MSC.

**Inter-MS-Handover** The connection will be changed between two cells that are managed by different MSCs.

The first two handovers can be carried out by the BSC if it supports the handover protocol. If this is not the case then also Intra-Cell-Handover have to be carried out by the MSC.

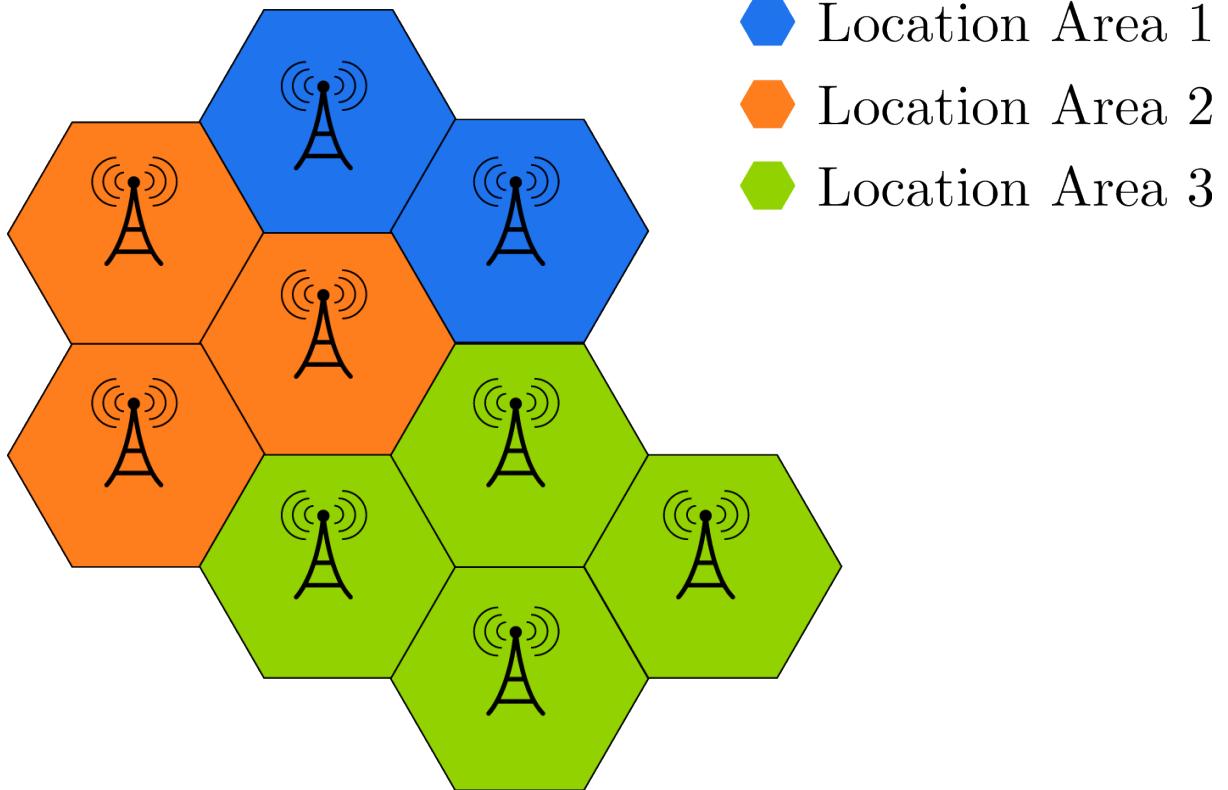
**Ping-pong Handover** A ping-pong handover is an undesirable effect where a handover is made to a neighbor cell and after short time back to the original cell [@Junius1995].

### **Location Area Updates - LAU**

Location area updates are used to ease the localization of mobile subscribers. Therefore, a MS must initiate a location area update once it leaves the location area. The HLR and VLR both store the current location area. A MS constantly measures the received signal strengths of all surrounding BTSs and reports it to the BSS to the one with the highest signal strength. If it connects to a BTS outside of the current location area then the MS initiates a location area update and tells the CN its new location area. There are two cases which can distinguished:

- change within the same VLR
- change to a new VLR

**Location Area - LA** A location area defines the area within the mobile network in which the mobile subscriber is located. Location areas are used to decrease the signaling necessary to locate a mobile subscriber within the network. Without location areas, the PLMN has to initiate a paging request to all BTSs. By using location areas, only the BTS within the location area have to carry out the paging request.



[fig:locationarea]

One or more BTS with the same MSC can be part of one location area, in most cases they are part of the same BSC. Fig. [fig:locationarea] shows that a location area can consist of several BTS. Each location area is uniquely defined by the *Location Area Identity* (LAI). The LAI (see Tab. [tab:lai]) consists of the Mobile Country Code (e.g. Austria 232) which identifies the country, the Mobile Network Code (e.g. A1 1) which identifies the PLMN and the Location Area which identifies the location area within the PLMN. Together with the Cell-ID (CID) the LAI forms the Global Cell-ID (GCID). The size of a location area depends on:

- Cell density
- Voice and data traffic
- Network operator configuration

Mobile Country Code & Mobile Network Code & Location Area Code

3 digits & 2 or 3 digits & 5 digits

tab:lai

**Updates** For its operation the MSC (HLR, VLR) needs to know where the mobile subscriber is located. Therefore, location area updates are required. There are two kinds of location area updates first normal updates and second periodic updates. Normal location area updates (NLAU) happen either when the MS is switched on or off during an IMSI attach/detach or when a MS connects to a new BTS in a different location area. Periodic location area updates (PLAU) are defined by the network operator. Each MS has a PLAU timer for this purpose, whenever the timer expires the MS performs a PLAU.

## Antennas

Antennas enable wireless communication between a receiver and a transmitter. In GSM, they are the direct air interface between the MS and the Radio Access Network (RAN). Antennas can be designed to

transmit or receive signals equally in all directions (omni-directional antennas) or transmit and receive signals in a specific direction (directional or high gain antennas). In GSM networks, a combination of both is used. Omni-directional are mainly used in rural areas to cover a large area whereas directional antennas are used to cover a smaller area with a higher traffic load.

**Sector** Sector antennas are a special use case of directional antennas where several directional antennas are combined and each covering a particular sector such as a  $120^\circ$  horizontal pattern. In GSM the most common installations of sector antennas are  $60^\circ$ ,  $90^\circ$  and  $120^\circ$ . Based on the traffic load, smaller sectors can be used with larger ones on the same tower (e.g. two  $60^\circ$  antennas with two  $120^\circ$  antennas).

**Antenna size** The antenna size or more general its coverage is defined by the antenna gain, the antenna characteristics and the transmit power. In mobile communication networks there are four different types of cells in use that are distinguished by their coverage.

**Macrocell** Provides the primary radio coverage for mobile networks. Macrocell antennas are mounted on ground-based towers, buildings or other existing infrastructure in order to have a clear view over the coverage area.

**Microcell** A microcell offers additional capacity within the coverage of a macrocell. Typically, microcell antennas are mounted at street level. Microcells have lower output power than macrocell and therefore a decreased coverage (e.g. 300 to 1000 meters).

**Picocell** To provide coverage inside buildings or to a high numbers of users. The coverage of a picocell is 200 meters or less.

**Femtocell** A low-powered, coverage is on the order of 10 meters, base station designed for use in a home or small business [@Zhang2011].

## Mobile Network Mobility Simulations

Mobile networks permit its subscribers to move freely within the coverage area. To evaluate and ensure the mobility, mobile network operators need to understand the user mobility. A user mobility can be described by a mobility model. A mobility model describes the user behavior and activity using simulation and analytic models. Simulation models are based on realistic mobility scenarios whereas simplified assumptions about the users movement behavior are the foundation of analytic models. Two major analytic models which are used in mobile network simulation are the *Random Walk* [@Akyildiz2000][@Bettstetter2001][@Bettstetter2002] and the *Manhattan Grid* [@Markoulidakis1997].

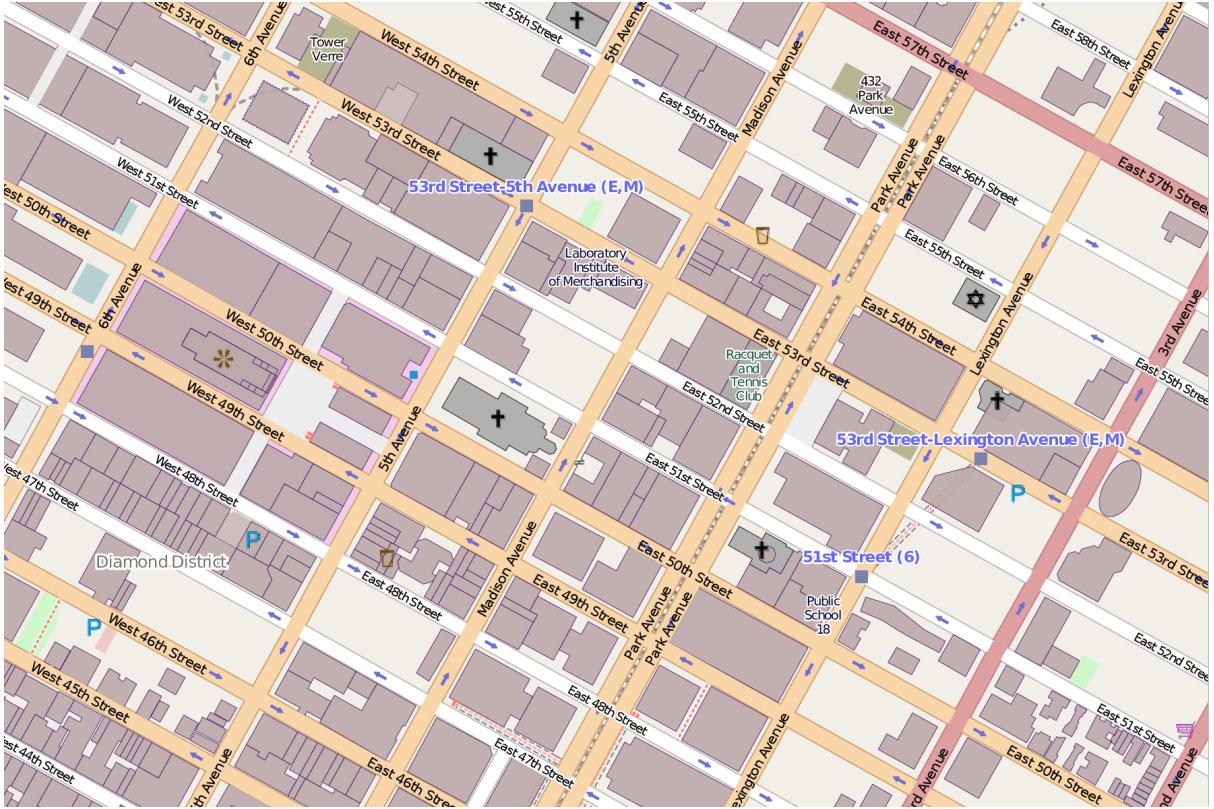
### Random walk

The mobility of entities in nature is unpredictable, therefore a variety of models exists in literature. These models try to describe the mobility for different scenarios. The random walk model is a memory-less system and maintains no record of previous locations and speed. In each iteration dices a new direction and speed within a predefined range [@Camp2002]. An example for a generated random walk is depicted in Fig. [fig:randomwalk]. The speed is limited to the interval  $0..5$  and angle is limited to  $-15..15^\circ$ . The mobility of entities in nature is unpredictable, therefore a variety of models exists in literature. These models try to describe the mobility for different scenarios. The random walk model is a memory-less system and maintains no record of previous locations and speed. In each iteration dices a new direction and speed within a predefined range [@Camp2002]. An example for a generated random walk is depicted in Fig. [fig:randomwalk]. The speed is limited to the interval  $0..5$  and angle is limited to  $-15..15^\circ$ .

```
[ymin=0,xlabel=,ylabel=,title=] table[mark=none];
[fig:randomwalk]
```

## Manhattan Grid

In a Manhattan grid scenario streets are aligned in a square grid. It is named after the district Manhattan in the City of New York. Fig. [fig:manhattan] shows a map of Manhattan and its road network. It can be seen that roads consist of straight lines and each junction forms a right angle. The Manhattan Grid mobility model can work on any road network. At each junction, the logic is executed which decides if the current street shall be used or a different one. The logic can be modeled as a random number generator with a defined probability density function. For example the probability to drive straight can be 60%, turn right 30% and turn left 10%. The Manhattan Grid mobility model is an extension to the random walk model where participants are not allowed to move freely, but rather move on a defined road network.

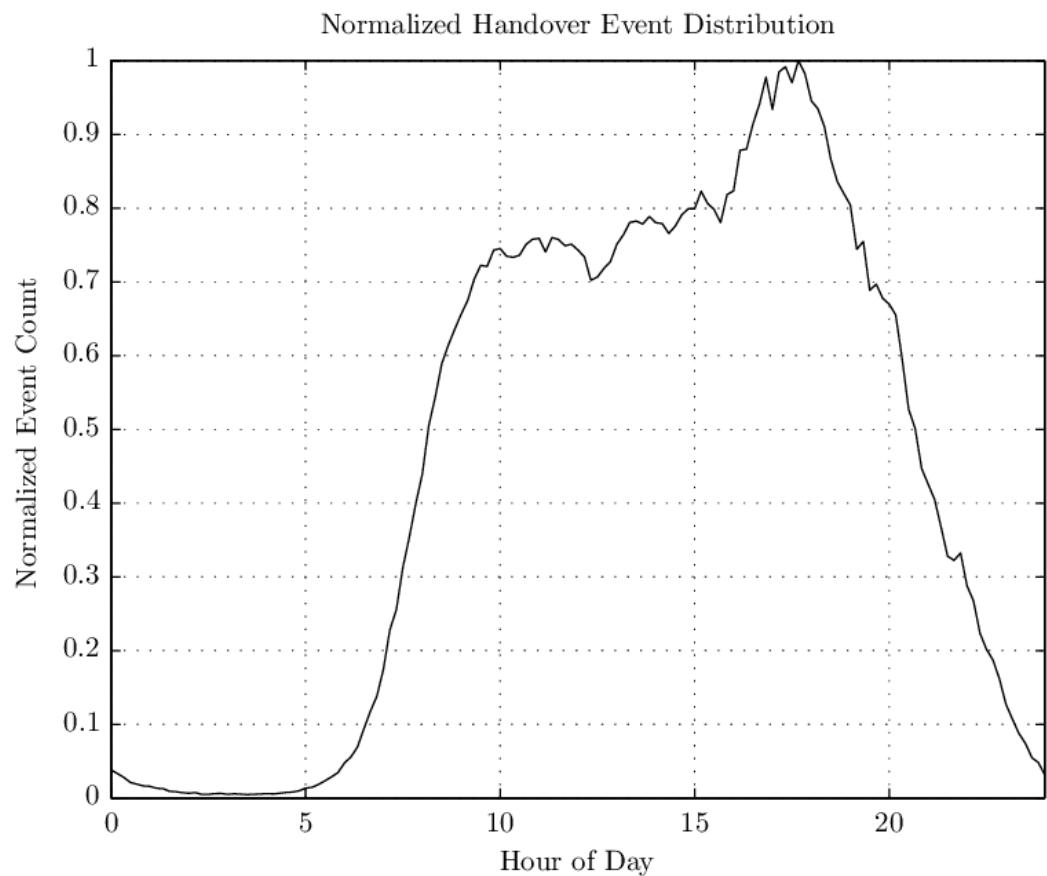


[fig:manhattan]

## A1 Data

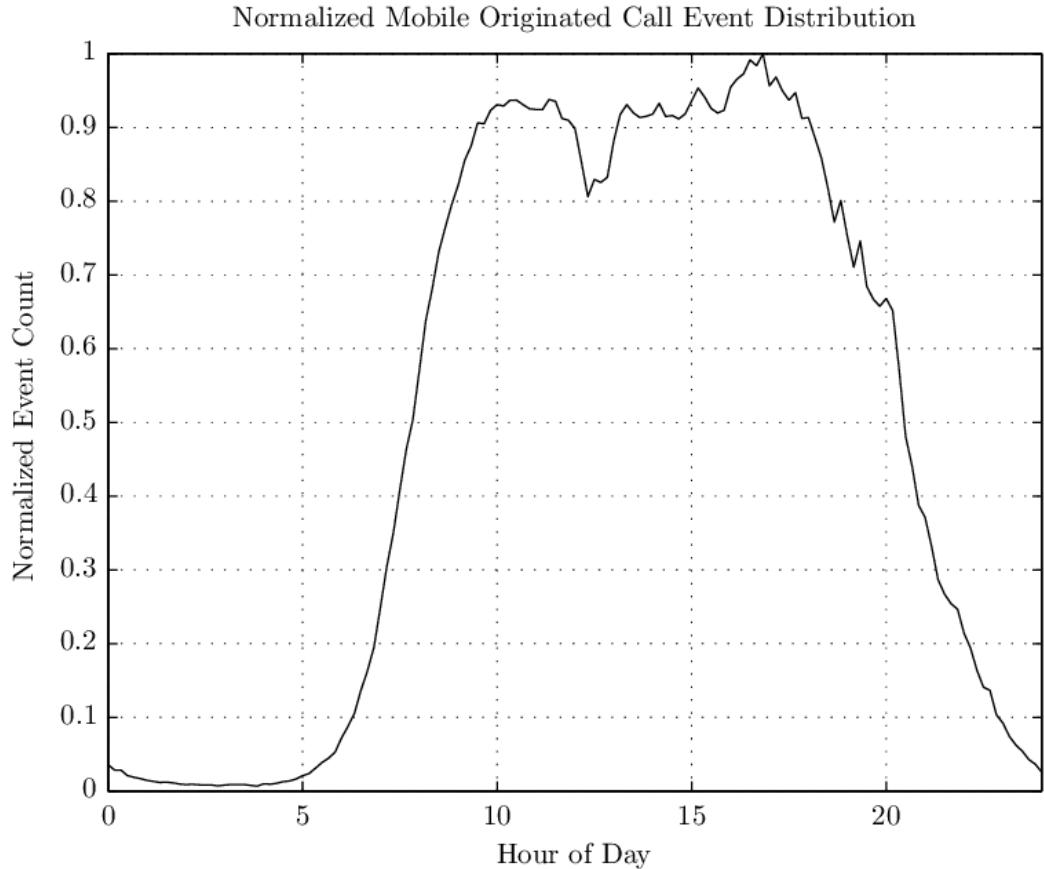
We are using mobile subscription data from A1. A1 is the largest network operator in Austria with 5.3 million subscribers. The used data was recorded on the network of A1 between Monday, November 2010 and Sunday, November 2010. The captured interfaces, events and the data structure of the system will be depicted in the succeeding subsections. Fig. [fig:distsEvents] shows the distribution of events for Monday, November 2010. It can be seen from this that the distribution for handover events is not the same as for mobile originated calls. A strong increase of handover events (see Fig. [fig:distHandover]) takes place at 17 o'clock. This is due to the end of work of subscribers. People are leaving from work and start calling their friends while on their way home. On the other hand, a strong decrease of call activity (see Fig. [fig:distMoc]) takes place between 12 and 13 o'clock where people are having lunch.

[b]



[fig:disthandover]

[b]

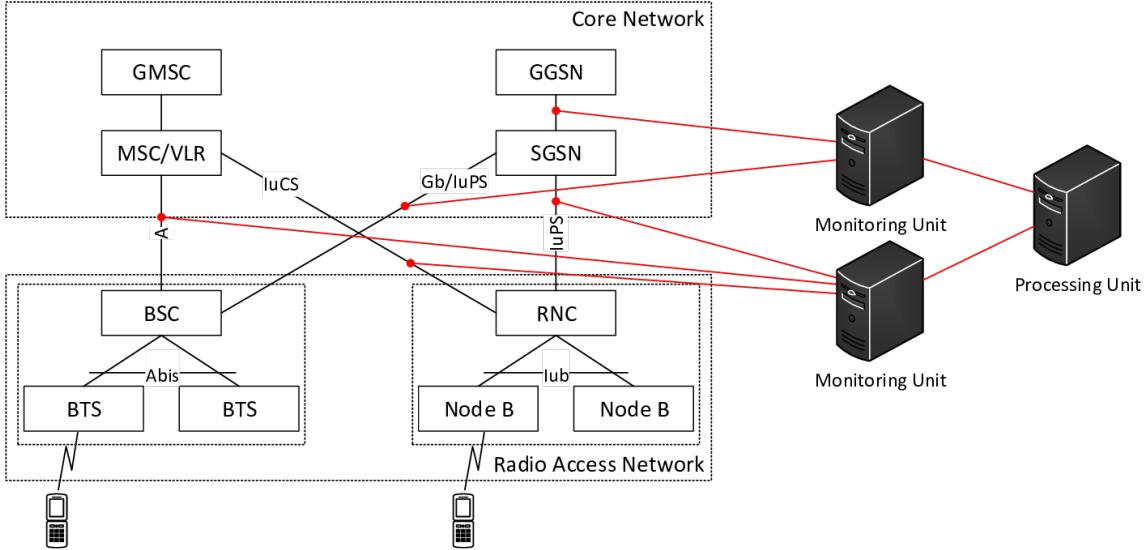


[fig:distmoc]

[fig:distevents]

### System Architecture

Monitoring units need to be installed in the Core Network of A1 to capture useful events. Fig [fig:A1Network] illustrates the system architecture used by A1 to capture events. The architecture is almost the same as for a typical GSM system, the only differences are the monitoring units responsible for capturing and forwarding events. In order to intercept the network, the monitoring units are attached to the main interfaces (A, IuCS, Gb/IuPS, IuPS) of the Core Network. The processing unit is responsible for aggregating events and ensuring anonymity.



[fig:A1Network]

**Interfaces** The monitoring units capture both *Circuit Switched* (CS) and *Packed Switched* (PS) data. Therefore, it intercepts the communication of CS and PS interfaces. The CS interfaces are A and IuCS and they belong to the MSC and enable voice calls and sms. The PS interfaces are the Gb/IuPS and the IuPS. Both are connected to the SGSN and enable subscribers to use packed orientated services (e.g. Internet).

## Data Structure

Each event captured by the monitoring unit is forwarded to the processing unit for further aggregation. An event is encoded into a binary format and then forwarded to the consumer. Each event has a fixed length (88 bytes) and its structure is shown in Fig. [fig:a1structure]. Because not all fields were used by our system, only the ones used are described.

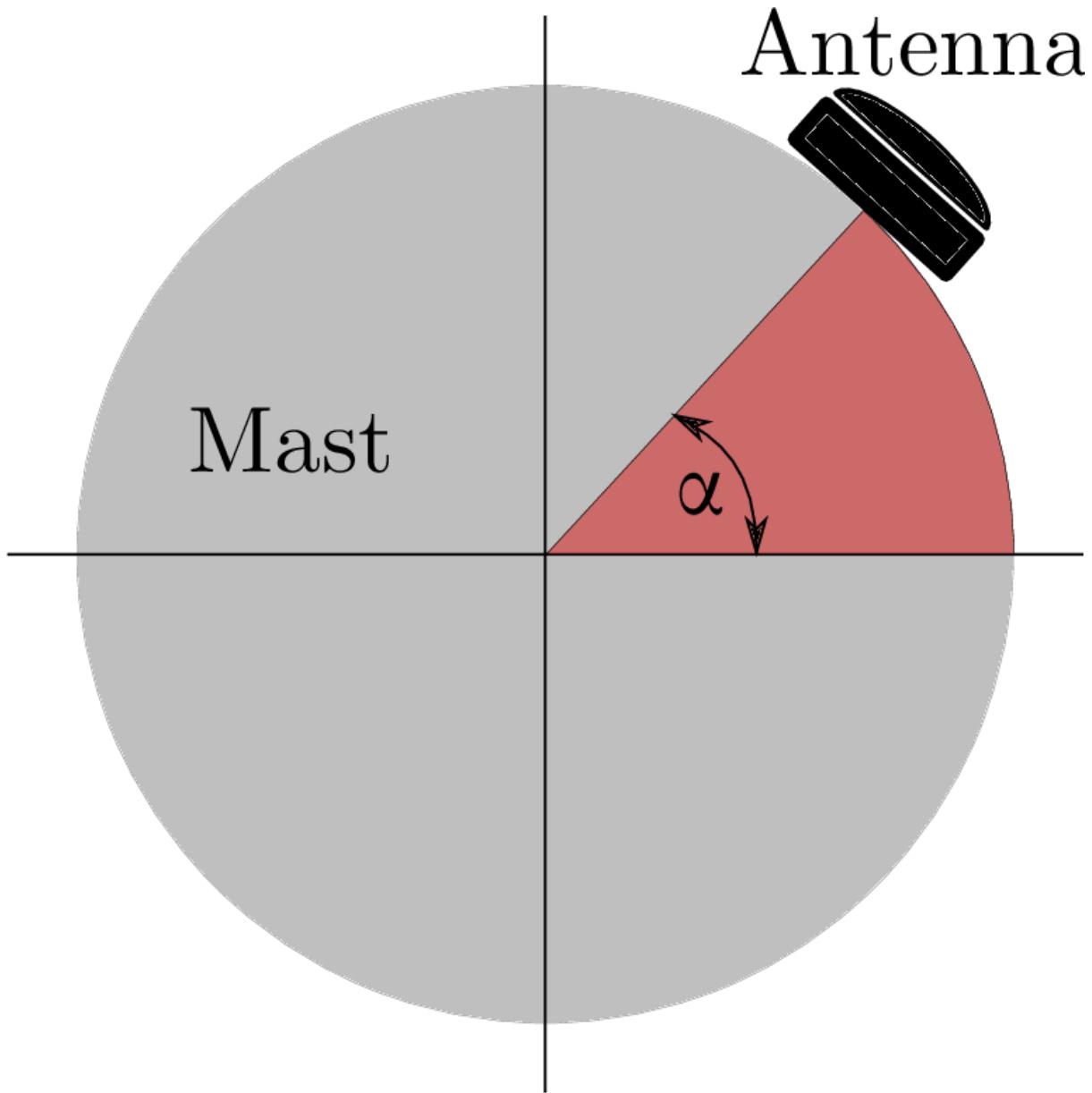
**Anonymous ID** This 32 bytes field contains a unique anonymous identifier of the mobile terminal. It changes every day at midnight, i.e. a device can be followed for maximum 24 hours time span before the assigned anonymous ID changes.

**Timestamp** The timestamp represents the exact time of occurrence of the event (i.e., the instant when the event has been captured by the monitoring unit).

**Latitude and Longitude** These two fields represent the position of the BTS in which the event occurred. They are both encoded as decimal numbers representing the WGS84 coordinates in degrees. The decimal number is encoded according to IEEE 754 floating-point double format@IEEE754.

**Event type** Every event includes a field (i.e., Event type) that indicates which type of event has been detected. All possible signaling events are described in Table [tab:eventtype].

**Angle** This field defines the installment of a sector antenna. It is the angle at which the antenna is mounted on the mast. Fig. [fig:antennaangle] illustrates the mounting of a directional antenna on a mast.



[fig:antennaangle]

[bitwidth=1.1em]

&

& &

fig:a1structure

**Encoding example** The following example illustrates the decoding of the input bit stream.

E4A2E4E263A2B54214A2B28E09AE02C36089219123696993EF9E1DF9D8647F1

496DC7340006E61B4030747CF7F849A84048143577861E680ooooooooooooo5780200

- Anonymous ID  
Encoded Byte Array = 0xE4A2E4E263A2B54214A2B28E09AE02C36089219123696993EF9E1DF9D8647F18
  - Timestamp: Wed Jan 14 12:06:28 CET 2009, in microseconds: 1231931188452123  
Encoded Byte Array = 0x496DC7340006E61B
  - Latitude: 16,45503187  
Encoded Byte Array = 0x4030747CF7F849A8
  - Longitude: 48,15788168  
Encoded Byte Array = 0x4048143577861E68
  - Radius: 1400  
Encoded Byte Array = 0x0000000000000000578
  - Input source: Metawin  
Encoded Byte Array = 0x02
  - Reserved: 0  
Encoded Byte Array = 0x0000000
  - Reserved: 0  
Encoded Byte Array = 0x000000000
  - Event type: Emergency Call  
Encoded Byte Array = 0x23
  - Reserved: 0  
Encoded Byte Array = 0x00
  - Reserved: 0  
Encoded Byte Array = 0x00000
  - Angle: 0  
Encoded Byte Array = 0x00000000000000000000000000000000
  - Speed: 0  
Encoded Byte Array = 0x000000000

## Events

In this subsection, we describe how everyday users activities are visible in the A1 data stream. The events detected by the monitoring units from each terminal depends on the type of terminal. While it is not feasible to cover all events (complete list of events is shown in Tab. [tab:eventtype]), we only describe the ones (mobility related) used in our system.

**Calls** When a terminal establishes a call with another subscriber a Mobile Originated Call event is created. The receiving terminal will create a Mobile Terminated Call event. Whenever a call is ended (e.g. one subscriber hangs up) an A Disconnect event is created. However in our research of the data stream, there was no evidence of an A Disconnect event.

- MOBILE TERMINATED CALL (0x1D): terminal receives a call
  - MOBILE ORIGINATED CALL (0x22): terminal originates a call
  - A DISCONNECT (0x18): The call is closed

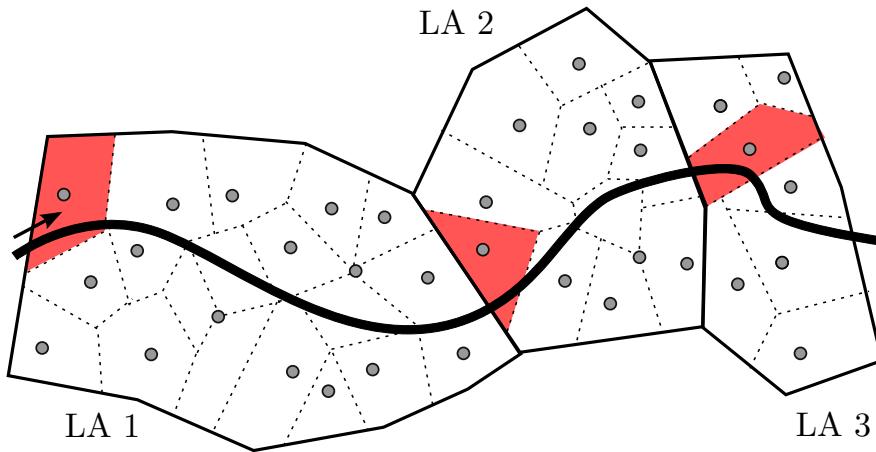
**Location Area Update** Devices are most of the time in “idle” mode where they are switched on but are not involved in any voice/data communication. In this state the device must still be reachable by the network. Thus, when a terminal changes to a cell in a different location area, it sends a Location Area Update message to the core network. This message appears in the data stream as:

- LOCATION UPDATE (ox1F), in case of 2G terminal
- IUCS LOCATION UPDATE NORMAL (ox15), in case of 3G terminal

If a devices do not change its Location Area for a specific time and the PLAU timer expires it sends a keep alive message to the core network.

- IUPS RA PERIODIC UPDATE (ox0C) and IUCS LOC UPD PERIODIC (ox17) in case of 3G terminal (for PS and CS domain respectively)
- LOCATION UPDATE (ox1F) in case of 2G terminals

Fig. [fig:latraversed] illustrates where a location area update (red cell) happens when a subscriber traverses different location areas on its way.



[fig:latraversed]

**Handover Cell Update** When a call is established the terminal is said to be in active “connected” state (see [subsub:connected]). In this state, the network is informed about each change of cell, regardless of the LA to which the cell belongs. A limitation is that the A1 data stream is not able to capture cell changes for 3G terminals (these messages do not reach the core network, but terminate in the radio access network, i.e. they cannot be monitored by the systems). However cell changes for 2G terminals appear as:

- GB CELL CHANGE (ox02), in case of an ongoing data connection,
- HANDOVER CELL UPDATE (ox20), in case of an ongoing call.

#### HEX & \*\* Event Name\*\* & Event Description

ox01 & GB ATTACH & GPRS terminal attaches to the PS

ox02 & GB CELL UPD & GPRS terminal changes cell

ox03 & GB OTHERS & GPRS terminal changes cell but message is unclassified

ox04 & GB RA UPD & GPRS terminal changes RA

ox05 & IUCS PAGING & UMTS terminal is paged in the CS

ox06 & IUPS PAGING & UMTS terminal is paged in the PS

ox07 & IU OTHER TLLI & UMTS terminal changes cell but message is unclassified

oxo8 & IU OTHER TMSI & UMTS terminal changes cell but message is unclassified  
 oxo9 & IUPS ATTACH & UMTS terminal attaches to the PS  
 oxoA & IUPS DETACH & UMTS terminal detaches from the PS  
 oxoB & IUPS RA UPD & UMTS terminal attached to the PS changes RA  
 oxoC & IUPS COMB RA/LA UPD & UMTS terminal attached to both CS and PS changes LA (and therefore RA)  
 oxoD & IUPS COMB RA/LA UPD WITH IMSI ATTACH & UMTS terminal attaches to the PS and the current RA differs from the one stored in the SIM  
 oxoE & IUPS RA PERIODIC UPD & UMTS terminal does not change RA for longer than a timer  
 oxoF & IUCS DETACH & UMTS terminal detaches from the CS  
 oxio & IUCS SMS ORIGINATED & UMTS terminal sends an SMS  
 oxii & IUCS SMS TERMINATED & UMTS terminal receives SMS  
 ox12 & IUCS SETUP & UMTS terminal requests the establishment of a call  
 ox13 & IUCS CONNECT ACK & UMTS terminal establishes a call  
 ox14 & IUCS DISCONNECT & UMTS terminal disconnects a call  
 ox15 & IUCS LOC UPD NORMAL & UMTS terminal attached to the CS changes LA  
 ox16 & IUCS LOC UPD PERIODIC & UMTS terminal does not change LA for longer than a timer  
 ox17 & IUCS LOC UPD & UMTS terminal attaches to the CS and the WITH IMSI ATTACH current LA differs from the one stored in the SIM  
 ox18 & A DISCONNECT & GSM terminal disconnects a call  
 ox19 & A OTHER & unclassified message on A  
 ox1A & PAGING CS & GSM or GPRS terminal is paged in the CS domain  
 ox1B & DETACH & GSM or GPRS terminal detaches from the network  
 ox1C & SMS & GSM terminal sends or receives an SMS  
 ox1D & MOBILE TERMINATING CALL & GSM terminal receives a call  
 ox1E & CM REESTABLISHMENT & NA  
 ox1F & LOCATION UPDATE & GSM or GPRS terminal changes LA or emits a periodic RA/LA update or attaches to the network  
 ox20 & HANDOVER CELL UPDATE & GSM terminal in active state changes cell  
 ox21 & SUPPLEMENTARY SERVICE & GSM terminal request a supplementary service  
 ox22 & MOBILE ORIGINATED CALL & GSM terminal originates a call  
 ox23 & EMERGENCY CALL & GSM terminal establishes an emergency call  
 ox24 & SETUP & GSM terminal request the establishment of a call  
 ox25 & CONNECT ACK & GSM terminal establishes a call  
 ox28 & CLOSURE & NA  
 ox29 & GB PS PAGING & GPRS terminal is paged in the PS domain  
 tab:eventtype

## OpenCoverageMap

OpenCoverageMap was a project and master thesis done by a colleague of mine at the University of Applied Science Upper Austria. Dieter Schlosser [[@Schlosser2012](#)] goal was to create an open network coverage map for mobile networks. Instead of relying on measurements and coverage reports handed out by network operators he was using smartphones that were doing the measurements. This approach, where users are carrying out work, is known as crowd-sourcing [[@Surowiecki2004](#)].

### Purpose

In our project, we are using OpenCoverageMap data to evaluate our approaches. In order to measure the network, OpenCoverageMap captures events similar to the events captured by A1. The OpenCoverageMap application which is installed on the smartphone periodically records the position (using GPS), the phone state as well as the currently connected cell site. Tab. [tab:ocmrecord] shows an example of data that was recorded by the OpenCoverageMap application.

Timestamp	ID	Cell-ID	LAC	Latitude	Longitude	State
-----------	----	---------	-----	----------	-----------	-------

1328693325	57	9884	5502	48.244381047720614.260003995095	2
1328693326	57	9884	5502	48.244305652189614.2599996518319	2
1328693327	57	9884	5502	48.244230936849714.2600018504158	2
1328693328	57	9884	5502	48.244156690010614.2600058762345	2
1328693329	57	9884	5502	48.24408271771314.2600098594458	2

[tab:ocmrecord]

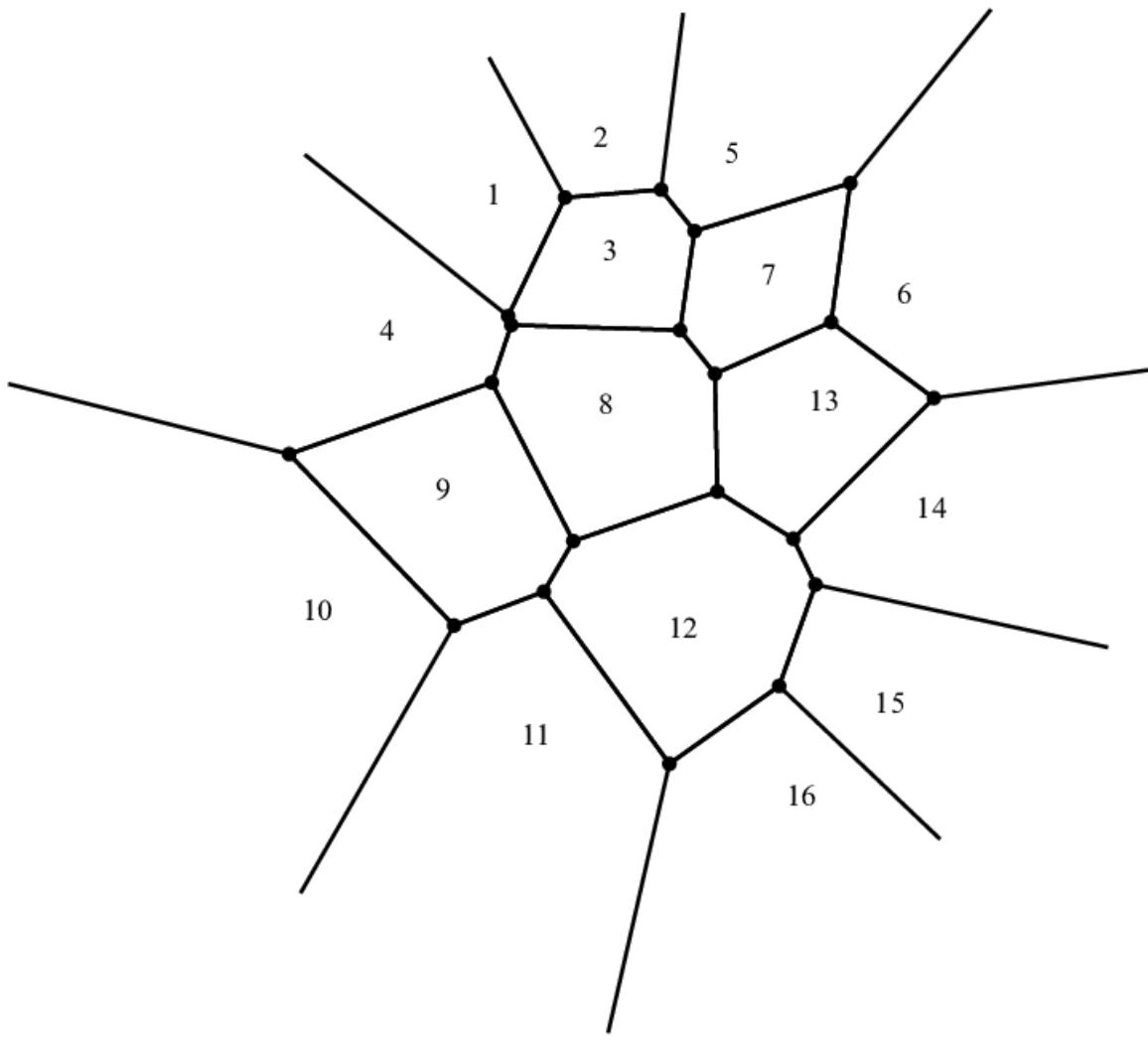
The GPS information allows us to evaluate a route estimated by our system with the actually traveled route (see [sec:route-validation] for more information).

## Data Structure

OpenCoverageMap does not capture events, therefore, our systems needs to covert the OpenCoverageMap data stream to the same format as the A1 data steam. Call establishment and termination events are generated by comparing the timestamps of each record. OpenCoverageMap logs the state of the MS every second. By iterating over the records and comparing the current state with the previous one, it is possible to extract call establishment and termination events. To generate handover events, an iteration over all records is done, each record compares the cell id and lac with the previous one. If both are similar no event is generated. However, if they are different a handover event is generated. After the conversion is done the OpenCoverageMap event stream looks the same as the A1 event stream.

## Voronoi Tessellation

To generate trajectories for each mobile subscriber their coarse location is needed. For this purpose, we first need an estimation of the coverage area of all cell sites. When a mobile subscriber is connected to a cell site we know that his current position must be withing the coverage area of the cell site. A fast approximation of the coverage area can be done with Voronoi diagrams. As mentioned before Tettamanti2012 et al. [@Tettamanti2012] used Voronoi tessellation to estimate the coverage area. Voronoi diagrams are widely used in computer science, e.g. pattern matching, space division, cluster analysis, collision detection, etc. as mentioned by Aurenhammer [@Aurenhammer1991].



[fig:voronoiz]

### Point Tessellation

The following assumptions are based on *Voronoi diagrams—a survey of a fundamental geometric data structure* by Aurenhammer [Aurenhammer1991] *Computational Geometry: Algorithms and Applications* by Mairt de Berg [Berg2000]. Let  $P = \{p_1, p_2, \dots, p_n\}$  be a set of  $n$  distinct points, in our case the location of each cell site. The Voronoi diagram of  $P$  is a subdivision of the plane into  $n$  cells. The property that each Voronoi cell must fulfill is that a point  $q$  lies in the cell respective to a site  $p_i$  only if  $\text{dist}(q, p_i) < \text{dist}(q, p_j)$  for each  $p_j \in P$  where  $i \neq j$ . Each cell defines the coverage area for the  $n$ -th cell site. Moreover each cell is a (possibly unbounded) open convex polygon. Fig. [fig:voronoiz] illustrates a Voronoi diagram for 16 random points in a two dimensional space. It can be seen that the outer cells are not closed, they shape an open convex polygon. In Section [sec:boundaries] we show a method to produce closed polygons.

$$\text{dist}(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (3)$$

## Boundaries

Voronoi diagrams are unbounded which means that the coordinates of vertices can be infinite. Unbounded polygons are undesired in a topology computation. If we would use unbounded polygons for coverage estimation, the area for this cell site would be infinite. A simple approach to eliminate this effect is to clip the Voronoi diagram with a rectangular bounding box. However, this approach is only satisfying if the cell site boundaries shape a rectangle. A better approach is to compute the convex hull over all sites and clip it with the Voronoi diagram.

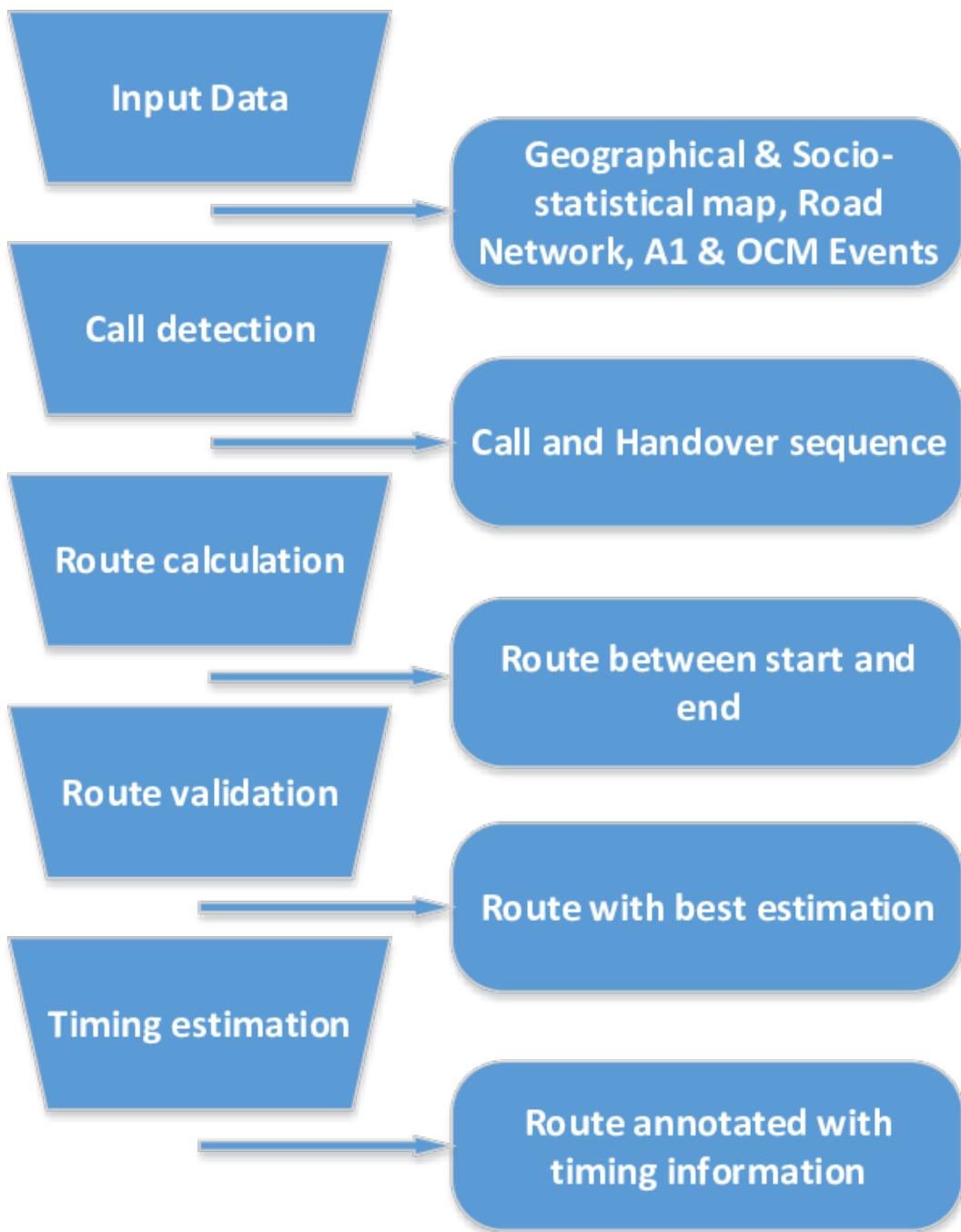
## Cell Tower Segmentation

We mentioned before that a GSM network consists not only of omni directional antennas but also of sectored antennas. The coverage area of each sector is defined by the angle at which the antenna is mounted on the tower. If we would only consider the location of each cell sites, sectors with different angles would have the same Voronoi polygon and, therefore, the same coverage area. In order to get a better representation of the coverage area for each sector, we moved the location of each sector based on its angle  $\phi$ . The definition of the movement can be found in Equation [eq:move]. The constant factor means a maximum movement of 3 meters in either direction.

$$x = x + \frac{\cos(\phi)}{50000}, y = y + \frac{\sin(\phi)}{50000} \quad (4)$$

## Trajectory Generation Pipeline

In this chapter, the pipeline to generate trajectories for mobile subscribers is introduced. The developed pipeline consists of a five-steps model, which is shown in Figure [fig:pipeline]. The concept to generate trajectories is the following: The input data is iteratively used to generate trajectories for each subscriber. The output is a trajectory which consists of the route, traveled by the subscriber as well as timestamps, which defines the point in time when the subscriber was located at this location. Given below each step of the pipeline is described in more detail to better understand how the system operates.



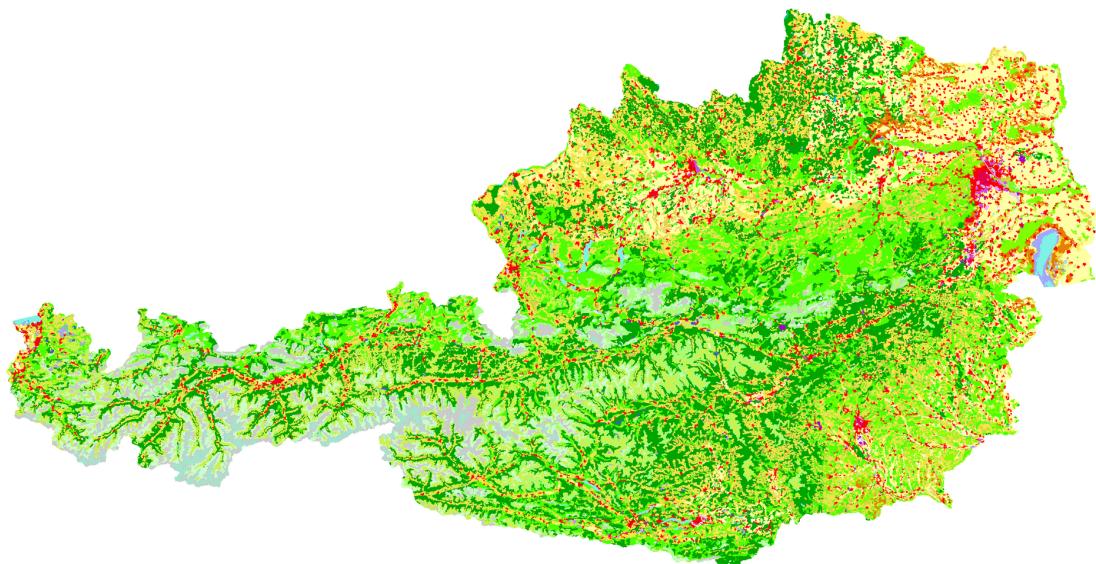
[fig:pipeline]

## **Input Data**

This section focuses on input data that is consumed by the system. The gathering of input data is the first step in the developed pipeline. Moreover input data can be distinguished between four data types, which are used by the system: geographical data, socio-statistical data, mobile subscription data (A1 and OCM), road network data. The different data types are described in the succeeding sections.

## Geographical Input Data

As mentioned before GSM does not expose an accurate location of the subscriber. In GSM, only the Cell-ID of the current connected cell is known. However, to calculate a route between the start and end of the call a more accurate location is needed. Because the coverage area of cell can be from as little as  $500m^2$  up to  $50km^2$ , boundaries need to be set in which subscribers can be located. The assumption is that the majority of subscribers will start or end their journey in an urban fabric whereas only a small fraction of subscribers will start or end their journey in open land. In the course of this project the geographic information is derived from data from the CORINE<sup>3</sup> project which was initiated by the European Commission in 1985. The main purpose of this project was to generate a geographic information system for the member states of the European Union. The CORINE land cover (CLC) project is one essential part of the CORINE projects which aim to develop a land cover information system for twelve member states of the European Union (Effective 2006). The CORINE land cover project distinguishes between five main categories and in total 44 land cover classes. Out of the five only two categories are applicable for the purpose of defining boundaries where a subscriber can be located: *artificial surfaces* and *agricultural areas*. The three remaining categories –*forest and semi-natural areas, wetlands* and *water bodies*– can be considered as areas where subscribers are not starting or ending their calls. The CORINE land cover data for Austria can be obtained from the following service<sup>4</sup>. The land cover data is provided in two different data types, first as a raster image with a grid size of 100 meters and second as a Shapefile..Figure [fig:clcaustria] illustrates the land cover map<sup>5</sup> of Austria. The red spots within the image represents urban fabric areas whereas green denotes to forest and semi-natural areas.



[fig:clcaustria]

<sup>3</sup>CORINE: “Coordination of Information on the Environment”; , last accessed on March 11,2014

<sup>4</sup><http://www.data.gv.at/daten/satz/?id=9246f37d-da69-4442-9504-ebd006ao59bb>, last accessed on March 11,2014

<sup>5</sup>Image source: I, last accessed on March 11,2014.

## Socio-statistical Input Data

The second type of input data used in the generation pipeline are population density maps. Population density maps belong to the class of socio-statistical and provide information about the distribution of population in a geographical area. Gallego et al. [@Gallego2010][@Gallego2011] describe several disaggregation methods which can be used to generate a population density map of the European Union. Their approach uses information from the *CORINE* land cover project to derive a population density grid with 100 meters grid spacing. The population density grid is used in the developed system to narrow the location of a user within the current connected cell. This works in conjunction with *CORINE* land cover. A more detailed description of the described process will be presented later. The population density grid can be obtained freely from the European Environment Agency (EEA)<sup>6</sup>. Similar to the *CORINE* land cover data the population density grid is available in two data types.

## A1 Dataset

The mobile subscription data which is provided by A1 is used to generate trajectories for subscribers. To generate trajectories, the developed system is fed with an event stream for a special user. This event stream contains all events which have been captured during one day. As we mentioned before in Section [sec:anonymous] the anonymous user id changes at least every 24 hours. Therefore it is only possible to extract trajectories for one day per subscriber. Each event stream can contain zero, one or  $n$  calls. Every call in the event stream represents a trajectory.

## OCM Dataset

In order to validate our approach we need not only an event stream but also the path traveled by the subscriber. Therefore, the developed system uses an event stream provided by the OpenCoverageMap project. After a conversion to the A1 event format, this event stream can be provided to the developed system. In addition to the A1 event stream, it contains the traveled path which was recorded via GPS on a mobile phone. The GPS tracks allow the system to validate the route finding process as well as the timing estimation. The comparison between the estimated trajectory and the actually traveled route helps to validate the routing. By calculating the speed from the GPS the system can validate how well the timing estimation has been done. More details to the validation process will be provided in Section [sec:routevalidation],[sec:timing-estimation].

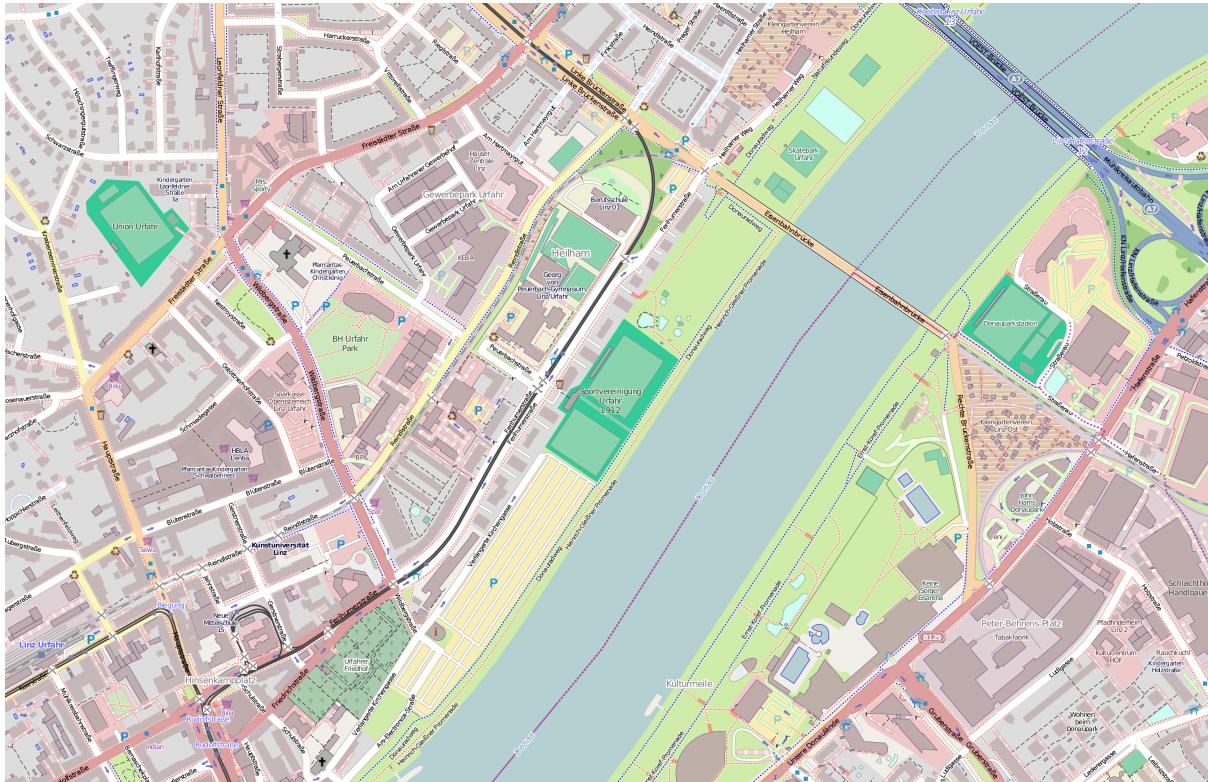
## Road Network

The last input data for the developed system is a representation of the road network. Individuals walk or drive on predefined paths: streets, paths, etc. To generate trajectories, the system needs to have a representation of the road network to assign the path of subscribers on the road network. A freely available and open-source road network is provided by the OpenStreetMap<sup>7</sup> project. The aim of OpenStreetMap is to provide a world wide map which can be used without any royalties. Contributors all over the world are feeding OpenStreetMap with new data. Therefore, OpenStreetMap is very accurate for locations where the number of contributors is high. Figure [fig:maplinz] illustrates a map<sup>8</sup> of Linz, Austria generated by OpenStreetMap. It can be seen that OpenStreetMap provides a large variety of data (e.g. streets, paths, buildings, POI, etc.). The developed system utilizes the road network from OpenStreetMap to estimate a start and end position of the trajectory as well for route generation. Additionally OpenStreetMap is used to validate the timing estimation for subscribers where no GPS path is available which is the case for A1 subscribers.

<sup>6</sup>, last accessed on March 11,2014

<sup>7</sup><http://www.openstreetmap.org/about>, last accessed on March 12, 2014

<sup>8</sup>Image source: <http://render.openstreetmap.org/cgi-bin/export?bbox=14.278879165649414,48.31028478774528,14.302825927734377,48.320590159843626&scale=5754&format=pdf>, last accessed on March 12, 2014



[fig:maplinz]

## Voronoi Coverage

The developed system needs a representation of the coverage area for each cell site. As described before in Section [sec:voronoiites] Voronoi tessellation can be used to calculate an approximate coverage area. However, as Voronoi tessellation only takes the location of the cell site and not the physical characteristics (shadowing, antenna gain, reflections, path loss, etc.) the approximation is not very accurate. To calculate the coverage area –the Voronoi polygon – for each cell site, the system is using the location and the angle of the antenna as input data for the calculation. The angle is needed to separate sector antennas which are mounted on the same cell tower.

## Coverage Planning

Because the developed system is based on handover point estimation, an accurate approximation of the coverage area is needed. Besides Voronoi tessellation, a network planning tool is used. The network planning tool not only takes the location of the cell site but also the physical characteristics into account. The physical characteristics are the following:

- Digital Elevation Model
- Transmit Power
- Path Loss
- Shadowing
- Line Of Sight
- Multipath propagation

By using the physical characteristics, the calculated coverage area is more accurate than the simple approximation done with Voronoi tessellation. Since this system has no access to the physical properties of the operator, it is using freely available information. This information consists of a Digital Elevation Model of the research area and the transmit power of cell sites.

**Digital Elevation Model** The network planning tool is using a Digital Elevation Model to calculate properties such as path loss, shadowing and line of sight. These properties are important for the coverage prediction. The European Environment Agency<sup>9</sup> provides a Digital Elevation Model of Europe. A Digital Elevation Model is aligned in a grid and for each cell of the grid the altitude above sea level is known. The EEA Digital Elevation Model consists of a raster with 25 meters grid spacing. The coverage prediction is limited by the resolution of the underlying Digital Elevation Model. Therefore by using a better model the accuracy of the prediction can be increased.

**Transmit Power** The transmit power defines the power level of the transmitter in the antenna. An increase in transmit power means an increase in coverage as the path loss effect will be minimized. Because the network operator has not provided any information about the transmit power of its BTS the system is using an alternative data source. In Austria the Forum Mobilkommunikation provides a service called *Senderkataster*<sup>10</sup>. The Senderkataster allows to view all broadcast and mobile communication transmitters on a map. In addition to the location it, also depicts the transmitting power. The Senderkataster defines the following four categories for transmit power:

- Category 1: < 15 Watt
- Category 2: 15 - 50 Watt
- Category 3: 50 - 100 Watt
- Category 4: > 100 Watt

By querying the Senderkataster with the research area, the system can get the transmitting power. However, the Senderkataster is a voluntary project and relies on the data provided mobile network operators. This has the disadvantage that every cell site of a network operator is listed in the Senderkataster.

**Data Extraction** The Senderkataster service can be queried with simple HTTP requests (see Listing [lst:senderbbox]). The service takes a bounding box of the research area and returns all transmitters within the bounding box. The bounding box is defined by the four parameters –left, right, bottom, top – EXL, EXR, EXB and EXT. The coordinates are projected in the WIGeoEU projection (EPSG 7397)<sup>11</sup>. After the request was sent to the service, the service delivers all the transmitters in the following format 301377|314876|2057165. The first parameter is the internal ID of the transmitter, second the X coordinate, and lastly the Y coordinate. The ID can then be used to retrieve additional informations such as the transmitting power, the transmitter type and if it is mounted on a house or a tower. To retrieve more information about the transmitter a request (see Listing [lst:transmitterinfo]) to the service with the ID is made. The response to the request is the following: mobilfunk|301377|GSM/UMTS|d|41,48|1. The first parameter describes the transmitter, it can either be a mobile radio network or a broadcast transmitter used for terrestrial radio and television. Second is the ID, the third one depicts the used technology (GSM, GPRS, UMTS, LTE), the fourth indicates if the transmitter is mounted on the roof (d) or a tower (e), followed by the transmit power in Watt and the last parameter shows if other transmitter are using the same tower.

```
http://www.senderkataster.at/functions/getPoints.php5  
?EXL=312790.83396198  
&EXR=315119.16603802
```

<sup>9</sup>Data source: [http://www.eea.europa.eu/data-and-maps/data/ds\\_resolveuid/ca503256de1b4231b029e4145doa8b7b](http://www.eea.europa.eu/data-and-maps/data/ds_resolveuid/ca503256de1b4231b029e4145doa8b7b) last accessed March 19, 2014

<sup>10</sup>Project description: <http://www.senderkataster.at/> last accessed March 19, 2014

<sup>11</sup>Projection string: <http://spatialreference.org/ref/sr-org/7397/html/> last accessed March 19, 2014

&EXB=2056054.8754715  
&EXT=2057801.1245285

<http://www.senderkataster.at/functions/getInfos.php5>  
?TYPE=mobilfunk  
&ID=301377

**Cell Site and Transmitter Position** The latitude and longitude coordinates of the A1 data specify the location of the cell tower. On each cell tower there can be one or more transmitters mounted. The network planning tool is using two databases, one for the cell site (tower position) and the second one for transmitters. Each transmitter is assigned to one cell site whereas a cell site can be connected to one or more transmitters. By querying the A1 data set and only storing unique tuples of latitude and longitude, the system can derive all cell site locations. To get the transmitters for each cell site, the A1 data set is queried for each unique location tuple. A basic transmitter consists of a location, Cell-ID, LAC, power, height and angle. The location is a direct relation to the cell site. The power indicates the transmit power. As there is not information about the installment height of the transmitter it was set to be 20 meters for all transmitters. A higher transmitter height might increase the coverage area for areas shadowed by hills or buildings. The angle defines the mounting angle and indicates in which direction the transmitter is beaming.

**Best Server Plot** All the above mentioned data is fed into the network planning tool to calculate the best server plot that is used as a coverage prediction. A best server plot calculates for each transmitter its path loss effect. The path loss specifies the reduction of power density of an electromagnetic wave as it propagates through space. To calculate the path loss for each transmitter the COST-Hata model is used which extends the urban Okumura-Hata model . The main equations of the COST-Hata model are depicted in Equation [eq:costhata].

[eq:costhata]

$$[eq:costhata] L = 46.3 + 33.9f - 13.82 hB - a(hR) + [44.9 - 6.55 hB] d + C$$

$$[eq:antennacor] a(hR) = (1.1f - 0.7)hR - (1.56f - 0.8)$$

C =

0 dB

3 dB

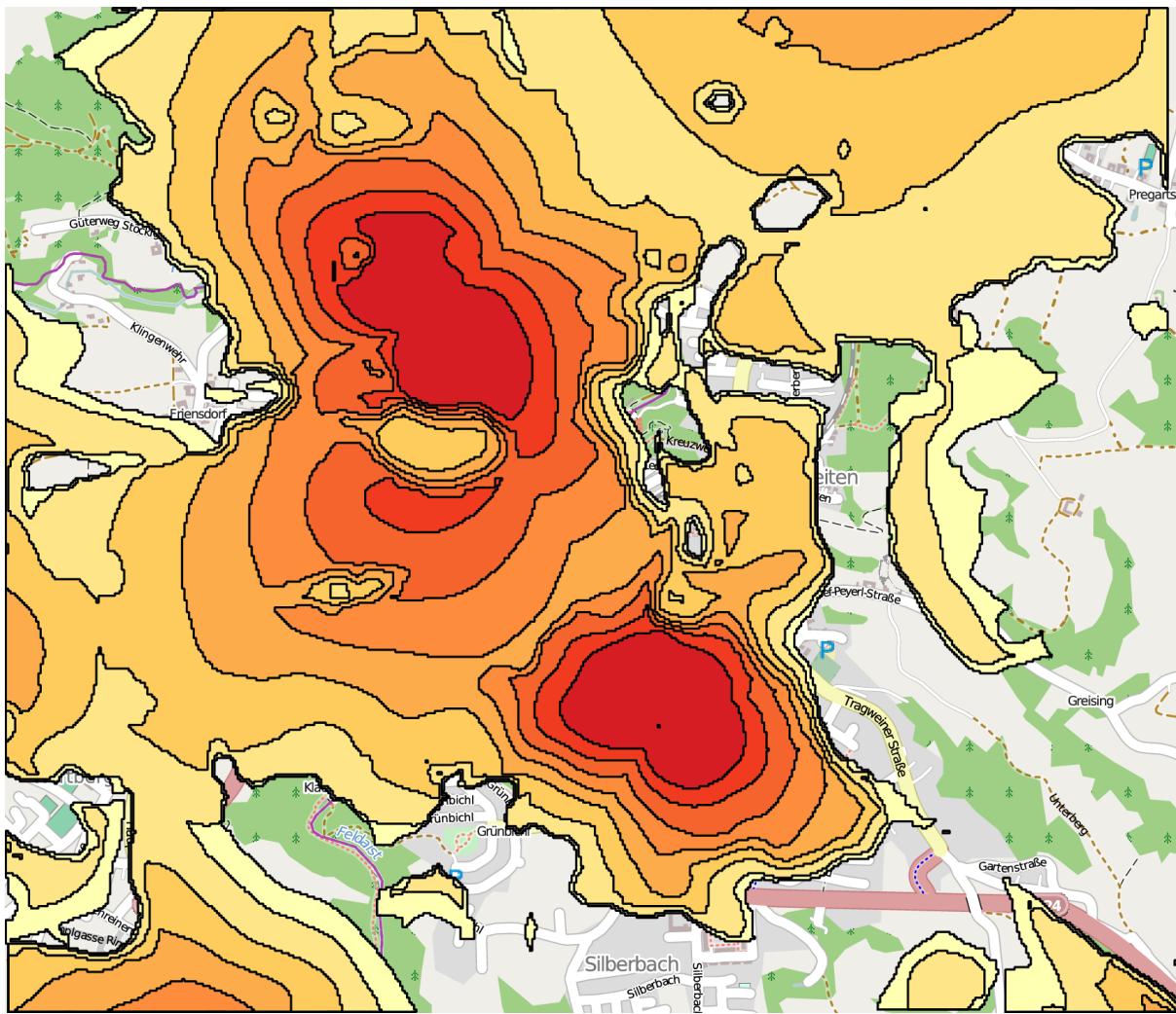
L = Median path loss. Unit: Decibel (dB) f = Frequency of Transmission. Unit: Megahertz (MHz)

hB = Base Station Antenna effective height. Unit: Meter (m) d = Link distance. Unit: Kilometer (km)

hR = Mobile Station Antenna effective height. Unit: Meter (m) a(hR) = Mobile station Antenna height correction factor as described in the Hata Model for Urban Areas..

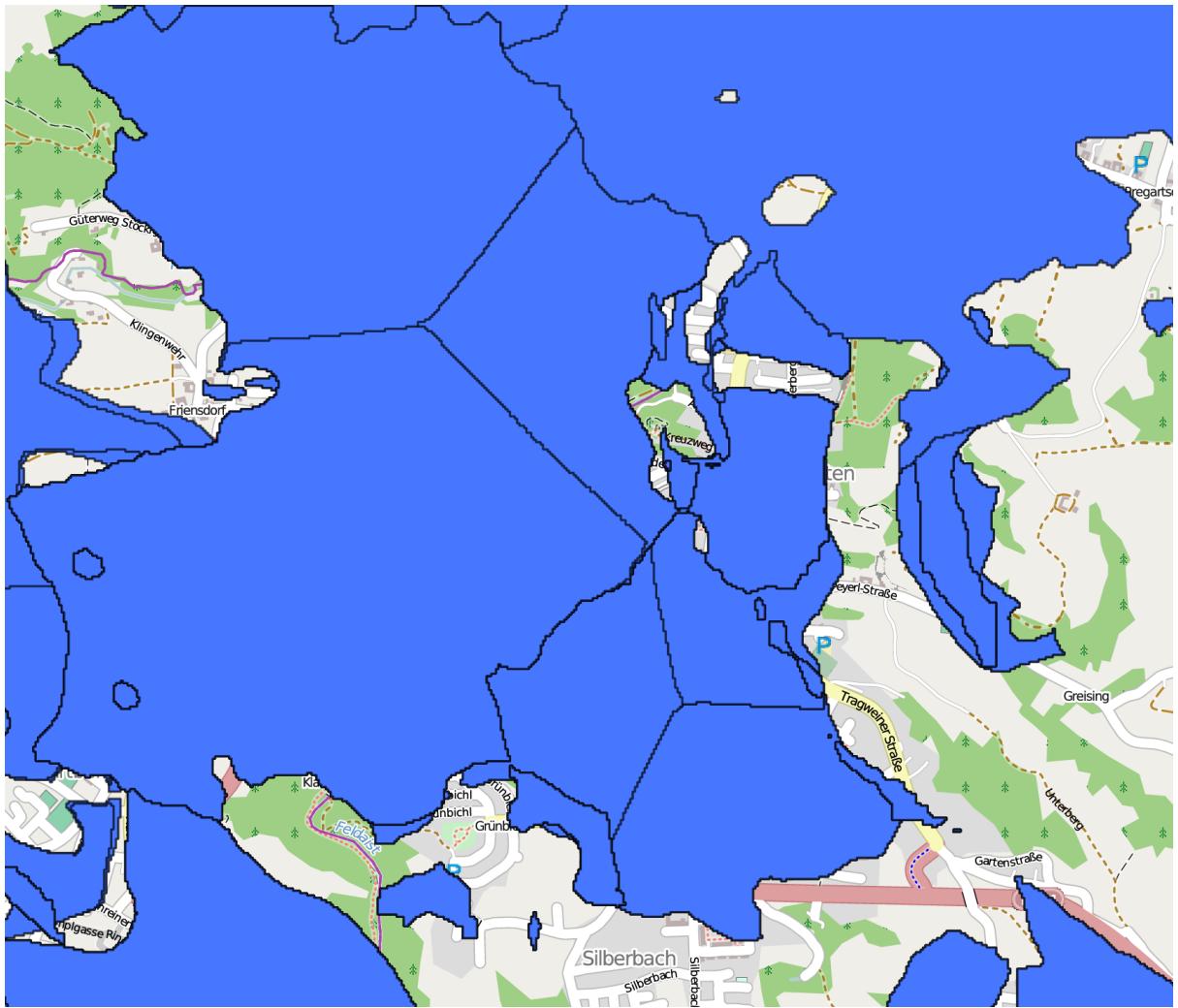
The path loss was calculated for various distance in the interval [0..5000] meters. Therefore, the path loss effect for each transmitter in the calculation area is known. The coverage prediction simply takes the path loss effect and assigns the best server as the transmitter with the least path loss.

[b]



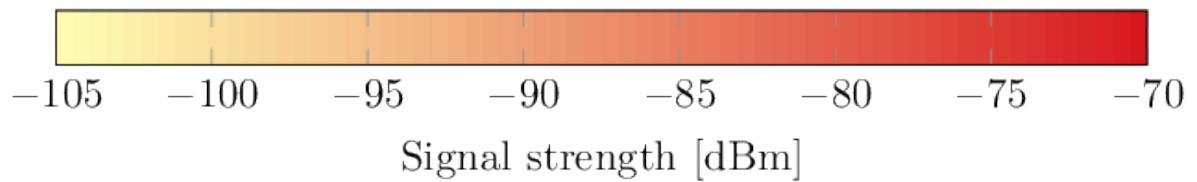
[fig:signallevel]

[b]



[fig:coveragepred]

[b]



[fig:signallevelbar]

[fig:cellarea]

Figure [fig:cellarea] shows two plots different path loss plots for Hagenberg, Austria. The first one Figure [fig:signallevel] depicts the attenuation for each transmitter in the area. The red area indicates a small attenuation whereas yellow indicates a high attenuation. The colorbar in Figure [fig:signallevelbar] indicates the signal strength (dBm) for the above signal level plot. From these attenuation values, a best server plot – coverage prediction – for each transmitter can be derived as shown in Figure [fig:coveragepred]. It can be seen that there is an overlapping area between neighboring transmitters. The system is using the predicted coverage area for start and end point estimation and timing purposes.

## Call Detection

This section describes the process of filtering calls within each event stream. The two used event streams – A1 and OCM – are using different events to signalize call establishment and termination. After this process, the system can generate a trajectory for each call found in the event stream.

### A1 Calls

To signalize the establishment of a call in the A1 event stream either a *Mobile Originated Call* or a *Mobile Terminated Call* event is issued. The first event is used when a mobile station initiates a call with another subscriber. The mobile station of the other subscriber will issue the second event once the call has been established with the first subscriber. When either of the involved subscribers terminates the call, a *A Disconnect* event shall be issued by both subscribers. However, during an investigation of the event stream there was no occurrence of this event. Therefore the developed system is using an approach by which it investigates events that have been issued after a call establishment. If the system recognizes either one of the following events: *SMS*, *Mobile Originated Call*, *Mobile Terminated Call*, *Location Area Update* or *IMSI detach* it generates a call termination event before.

### OCM Calls

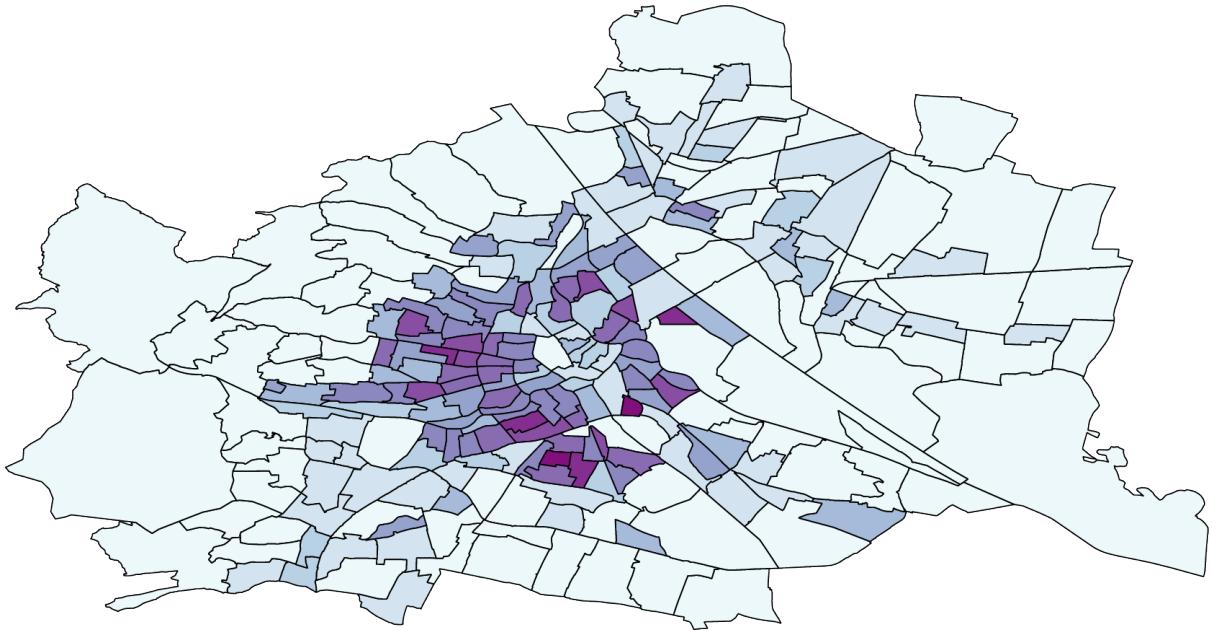
Rather than capturing events in the network, the OpenCoverageMap project periodically stores the state of the mobile phone. The state can either be *idle* or *connected*. In order to detect a call, the system has to find changes in the state. If a state change is detected the corresponding event is issued. A call establishment is detected when the state changes from *idle* to *connected*. In contrast to the call establishment, the call termination is detected when the state changes from *connected* to *idle*. The detection of call terminations is more accurate for OpenCoverageMap than for A1 where no information is available if a call is still ongoing. Because OpenCoverageMap periodically stores the state of the mobile phone whereas in order to detect a call termination for A1 events the system has to recognize a follow-up event.

## Route calculation

The following section presents the process of calculation a route for a mobile subscriber. In general, each call is used to calculate one route which is later transferred to a trajectory. The process consists of several steps which will be described in the succeeding sections.

### Start and End Position Estimation

The first step in the route calculation process is to estimate the subscribers start and end position. As mentioned before GSM only exposes the Cell-ID which only defines a boundary in which the subscriber is located. To narrow this boundaries the system is using geographical input (*CORNINE* land cover) and socio-statistical maps (population density maps). By using *CORNINE* land cover data the system can define areas where it is unlikely that a subscriber will start or end his journey. This step can be parameterized by giving each of the CLC classes a percentage factor. The percentage defines how likely it is that a subscriber starts or ends a call in this class. Our second assumption is that subscribers will more likely start or end their call in a higher populated area within the cell boundaries. More subscribers are located in higher populated areas than in less populated ones. Therefore the system is using population density maps in order to better estimate the start or end position. Figure [fig:popvienna] shows an example for a population density map for Vienna, Austria. In this example it can be seen that the density is higher in the inner districts than in the outer ones.



[fig:popvienna]

**Corine Land Cover Clipping** By clipping and therefore removing unwanted areas the system can narrows the boundaries in which a subscriber is located. To clip the coverage area of a cell the system loads the *Corine* land cover map for the particular area. The next step is to clip the cell coverage area with useful *CORINE* land cover categories such as *artificial surfaces* and *agricultural areas*. This technique known as polygon clipping removes unwanted area –*forest and semi-natural areas, wetlands and water bodies* – from the cell.

**Population Density Estimation** After removing unwanted areas within the cell coverage area the next step is to estimate the subscribers position. This step is the same for start and end positions. By using the population density information of the coverage area the system dices a position based on the population density. Therefore more subscribers will be located in higher populated areas.

**Population Density Clipping** The first step is to load the population density map for the coverage area. A random number generator with a defined probability density function will be created. The probability density function is derived from the population density map. When the system loads the population density map for the coverage area it will get  $n$  polygons with  $n$  different population densities. More generally speaking each area with a unique population density is represented by a polygon. Afterwards the random number generator is used to dice a area of interest. By using the population density as function for the random number generator areas with a hight population density will be picked more often than less populated areas. Once the system picked a polygon based on the random number generator it clips this polygon with the coverage area. This process narrows the boundaries of the subscribers location. Following the subscribers location will be set by dicing a location within the remaining coverage area. Figure [fig:population] illustrates an example for different population densities within the boundaries of one cell. We can see that total population is 315. We calculate a *pdf* for all three population cells based on the ratio of the population and the total population. The calculated *pdf* will be used to estimate the user's start and end position. As an example the probability that the starting point is within the polygon with Population 240 is given by  $240/315$ . Within the polygon "Population 240" the starting point is selected based on a uniform distribution.

[scale=0.70]

[ xlabel= $x$  in meter, ylabel= $y$  in meter, legend pos=outer north east] +[mark=none,draw=black, very thick,dashed] coordinates – cycle; +[mark=none,pattern=crosshatch,area legend,draw=black]

```

coordinates - cycle; +[mark=none,pattern=grid,area legend,draw=black] coordinates - cycle;
+[mark=none,pattern=crosshatch dots,area legend,draw=black] coordinates - cycle;
[fig:population]

```

**Random Point in Coverage Area** Dicing a random location into a rectangular shape is an easy task, however the shape of the remaining coverage area will unlikely be a rectangular. A simple approach is using the bounding box of the coverage area polygon and dicing coordinate pairs for the bounding box as long as one coordinate pair lays within the coverage polygon. This simple approach has one major disadvantage because it is unpredictable. There is no estimation for how long it would take that one coordinate pair is contained in the coverage polygon. A more sophisticated approach is to triangulate the coverage polygon into triangles. Triangulation of convex monotone polygons can be done with algorithms by A. Fournier and D.Y. Montuno [@Fournier1984] or Godfried Toussaint [@Toussaint1984]. The developed system uses the GDAL project [@GDAL] which provides an implementation of polygon triangulation for many platforms and programming languages. Figure [fig:triangulation] depicts the triangulation of a polygon into triangles. The polygon is split into 3 triangles by applying Fourniers algorithm.

```
[scale=0.70]
```

```

[ xlabel=x in meter, ylabel=y in meter, legend pos=outer north east] +[mark=none,draw=black,
very thick,dashed] coordinates - cycle; +[mark=none,pattern=crosshatch,area legend,draw=black]
coordinates - cycle; +[mark=none,pattern=grid,area legend,draw=black] coordinates - cycle;
+[mark=none,pattern=crosshatch dots,area legend,draw=black] coordinates - cycle;
[fig:triangulation]

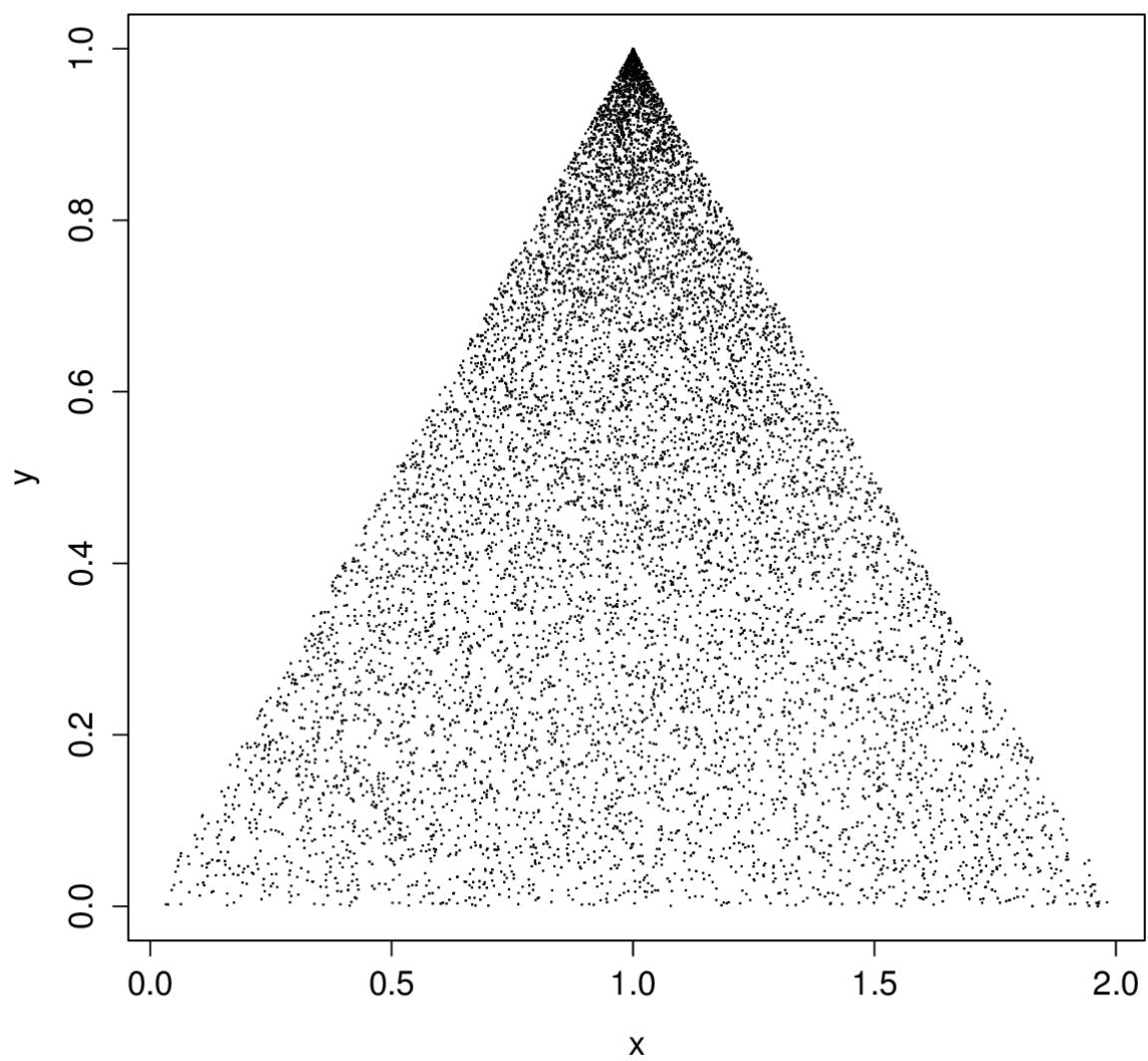
```

As already discussed before the developed system needs to uniformly distribute subscribers location with the polygon. A *pdf* will be created by using the area of each triangle as input. A random number generator will use this *pdf* and randomly assign the triangle in which the subscribers location shall be diced. In Figure [fig:comprandom] two different approaches for random triangle point picking. The first one is picking a point within the boundaries of the triangle whereas the second one is picking a point in a quadrilateral which consist of the triangle and its mirroring. To pick a point in a triangle the first approach is using Equation [eq:randtriangle] where  $a_1$  and  $a_2$  are uniform random numbers in the interval \$,v\_1\$ and  $v_2$  are two vertices's of the triangle. As depicted in Figure [fig:randtriangle] this approach does not create uniform distributed points within the triangle. The second approach is different from the first one as it does not pick points within the triangle but rather in a quadrilateral made of the original triangle and its mirroring. Figure [fig:quadrilateral] shows that this approach creates uniform distributed points. However the points are created in a quadrilateral instead of a triangle this can be overcome by removing points outside of the triangle.

$$p = a_1 * v_1 + (1 - a_1) * a_2 * v_2 \quad (5)$$

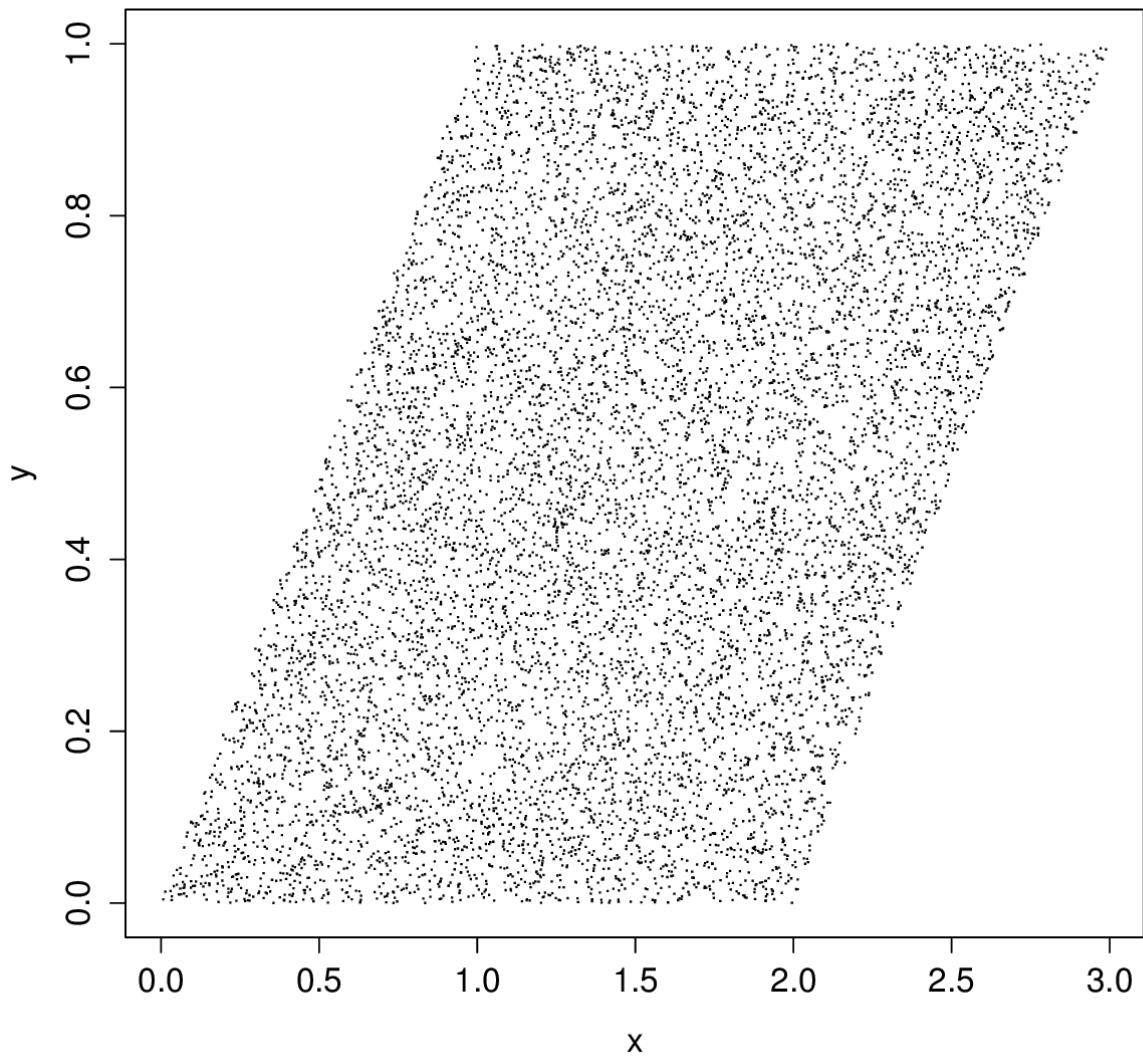
$$p = a_1 * v_1 + a_2 * v_2 \quad (6)$$

[b]



[fig:randtriangle]

[b]



[fig:quadrilateral]

[fig:comprandom]

## Routing

The developed system is using the OpenStreetMap road network and the OSM2PO<sup>12</sup> route engine. After estimating a random start and end position of the subscriber a route between these two points will be calculated with OSM2PO.

**OSM2PO** OSM2PO is a project developer by Carsten Moeller which allows routing on the freely available OpenStreetMap road network. On the first start OSM2PO is generating a graph network out of the road network. The graph network is used internally of OSM2PO and allows a faster route calculation. OSM2PO can calculate the fastest route by using speed limit information or the shortest route with minimum distance.

---

<sup>12</sup>Project description: <http://osm2po.de/>, last accessed on March 18, 2014

**Route Generation** The system first estimates a start and position for the subscriber and calculates the fastest route between these two points. The calculated route contains the geometry of the route, the used roads and the speed limit of each road segment. This information will later be used by the system to validate the route.

## Route Validation

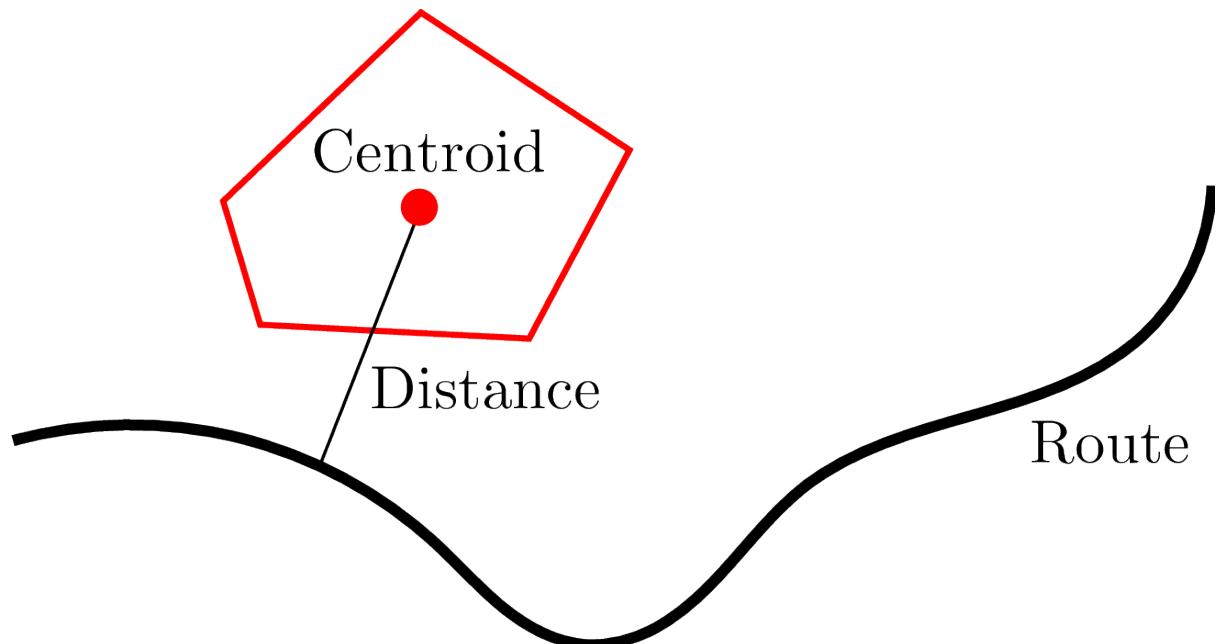
Route validation is necessary to use the best estimation of the traveled route. For each subscriber there will be more than one route generated based on a configuration parameter. The developed system uses two different types of route validation, the first evaluates the route geometry and the second the timing. Each of the validations can be treated separately or in conjunction. By combining the results a better validation can be achieved.

### Geometry Validation

Geometry validation is used to evaluate the estimated route with the subscribers handover sequence. In addition when using events from the OpenCoverageMap the system can validate the estimated route with the actual route based on the GPS information.

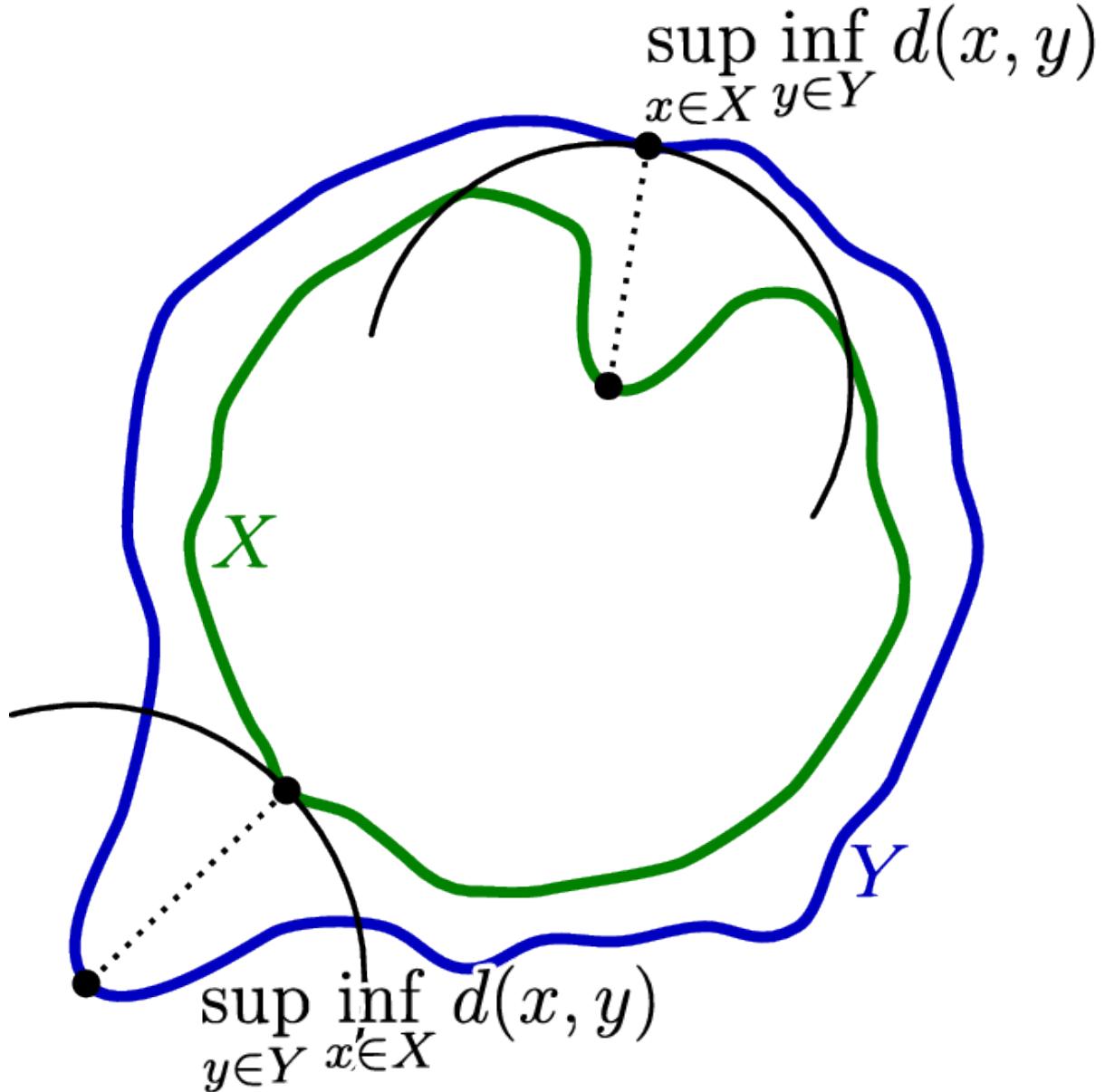
**Squared Sum** The first validation the system is doing for each route is to calculate the squared sum between the cell sites of the handover sequence and the calculated route. To compare different routes Equation [eq:sumsquaremine] was used. This is the same metric Tettamanti et al.[@Tettamanti2010] used for their system. For each calculated route  $j$  the squared sum of all minimum distances  $d_{i,j}$  between the route and the cell site was calculated. Figure [fig:rms] illustrates the minimum distance between the centroid of a cell site and a calculated route.

$$D_j = \sum_{i=1}^m d_{i,j}^2 \quad (7)$$



[fig:rms]

**Hausdorff Distance** If the system is using events from the OpenCoverageMap project the actual route traveled by the subscriber is known by storing GPS information. This allows the system to validate the calculated routes with the actual route. The developed system is using the Hausdorff distance named after Felix Hausdorff [ @Rockafellar1998]. The Hausdorff distance measures how far two subsets of a metric space are from each other, it is the maximum of all the distances from a point in one set to the closest point in the other set. In our system the two sets consists of the calculated and the actual route. Figure [fig:Hausdorffdistancesample] depicts the Hausdorff<sup>13</sup> distance between to paths X and Y. Therefore the system is using the Hausdorff distance to measure the similarity between the calculated and the actual route.



[fig:Hausdorffdistancesample]

### Timing Validation

Besides geometry validation the system is taking the time it needs to actually drive the route into account. When the system generates a trajectory for a subscriber it knows the time when the subscriber

---

<sup>13</sup>Image source: [http://en.wikipedia.org/wiki/File:Hausdorff\\_distance\\_sample.svg](http://en.wikipedia.org/wiki/File:Hausdorff_distance_sample.svg) last accessed on March 16, 2014

initiated and terminated the call. This information is used to extract the journey time. A ratio  $r_t$  between the actual journey time  $t_{actual}$  and the time it takes to travel the estimated route  $t_{route}$  is calculated (see Equation [eq:timeratio]). The ratio gives information if the estimated router is either to fast or to slow. A to fast route can indicate that the route is either too long or that the subscriber was stuck in traffic congestions. On the other hand a slower estimated route can indicate that either the wrong route was chosen or that the subscriber was going faster than the speed limit. Therefore a conjunction of all validations is needed to choose the best approximated route.

$$r_t = \frac{t_{actual}}{t_{route}} \quad (8)$$

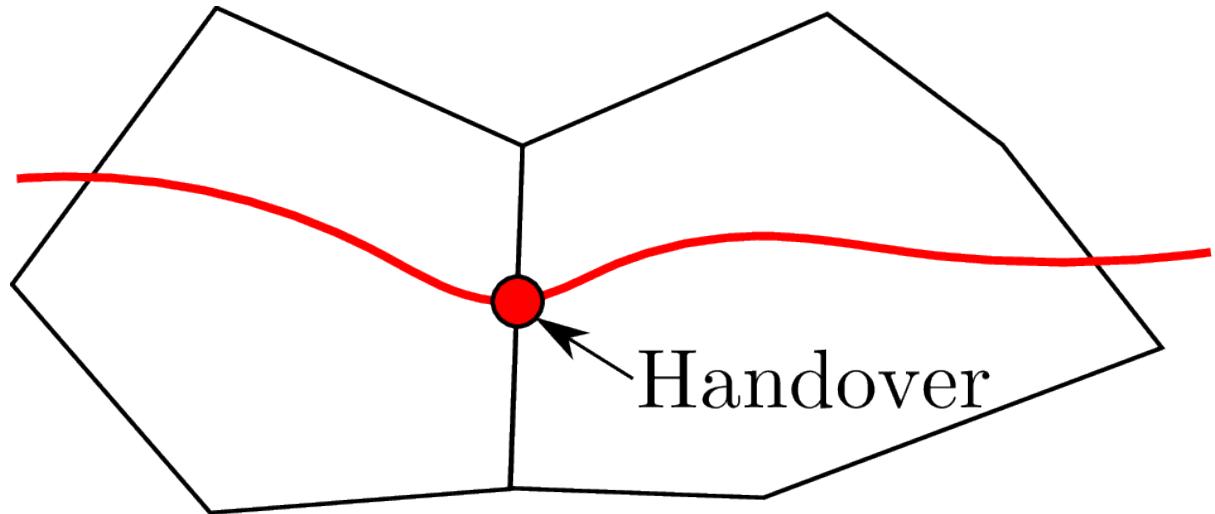
## Timing estimation

Whenever a subscriber is in an active call and moves from one cell to another one a handover event is issued. This event exposes a coarse location and a time stamp. By using this information the system can derive the velocity of the subscriber. The velocity of a subscriber is an important figure for mobility simulation. Because the velocity together with the estimated route describes the mobility of the subscriber. However as we the system only knows the cell in which the handover event originated – cell to which handover was made – an estimation of the actual handover point is needed. A more precise estimation of this handover point results in a better velocity approximation.

### Handover Point Estimation

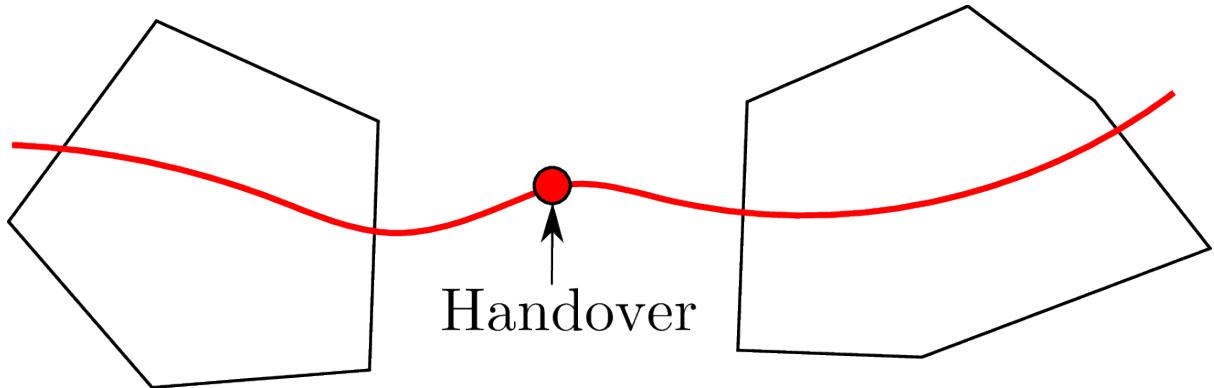
To estimate handover points the system is using two successive handover events. These two events consists of a time stamp, cell-Id and LAC. From the cell-Id and LAC the system can obtain the coverage area of the two cells. During research of the A1 and OCM data we discovered that there are two types of handover. The successive handover are either connected or apart. Connected handover are handover which coverage areas touches or overlaps each other. An apart handover is a handover where the coverage areas are not touching or overlapping.

[b]



[fig:handovertogther]

[b]



[fig:handoverpart]

[fig:handovertypes]

The difference of the two handover types is illustrated in Figure [fig:handovertypes]. For connected handover it is obvious where the handover happened. For this purpose the system is taking the touching or overlapping parts of the coverage areas and the intersection with the estimated route. In contrast for apart handover it is not clear where the handover happened. It could either happen at the boundaries of the first cell or the second cell or in between those. A simplification is to use the middle point between the two cells.

## Velocity

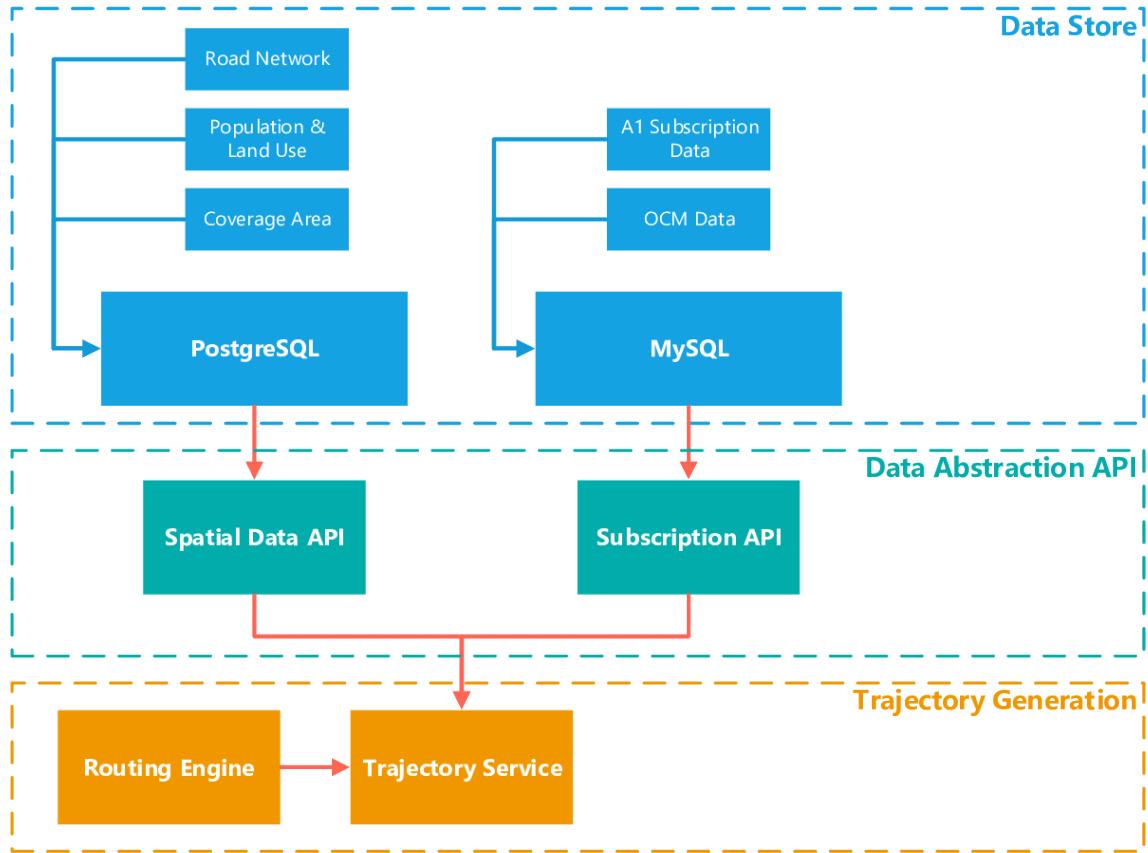
After the handover points have been estimated the next step is to enrich this information with the timestamps when the handover occurred. As mentioned before in Section [subsec:events] each event contains a precise timestamp when the event was captured by the underlying system. This information together with an estimated route and handover point allows to derive the velocity of the driver. The system knows the time difference between two handover events as well as the distance traveled. The traveled distance is the distance between two successive handover points. By cutting the whole route with the two handover points the system can obtain the route between the two handover points and therefore the actual distance between those. The velocity can then be calculated by dividing the distance  $s$  between the handover over the time difference  $t$  between the handover  $v = \frac{s}{t}$ . The calculated velocity represents the average distance between the handover.

## Implementation

The following chapter will present the actually implemented system. The implementation is based on Chapter [cha:concepts] where the concepts for the system have been presented. Through out this chapter all parts of the system and their responsibilities will be explained.

## System Overview

The developer system consists of a set of individual systems. These systems can be grouped in three groups. The first is responsible for storing information such as: mobile subscription data, population and land use information, road network and cell coverage areas. The second is a data abstraction layer which is used to aggregate the data and present to interested systems via an API. The last group is responsible for processing and generating trajectories. It is using the other systems via the API provided by the data abstraction layer. A brief overview of the implemented system is illustrated in Figure [fig:systemoverview].



[fig:systemoverview]

It depicts the data store which consists of two different databases, the first is responsible for storing geographic information and the second for storing subscription information from A1 and OCM.

## Data Store

The data store is responsible for storing data, this data either is either geographic information or subscription data. For geographic data the system is using PostgreSQL with the PostGIS extension. PostGIS extends PostgreSQL with support for spatial data structures and topological queries [@Obe2011]. Therefore, it allows the system to store the OpenStreetMap road network, population and land use information, the coverage area and the location of each cell site. PostGIS allows the use of advanced spatial queries as intersections and clipping which is later needed by the system. Subscription data is stored in a separate MySQL database. It is storing events captured by the A1 system and the OpenCoverageMap project. The database consists of two tables, one for A1 and the second one for OpenCoverageMap.

## Data Abstraction API

We first started to develop a system with direct access to the database via the JDBC driver for PostgreSQL<sup>14</sup> and MySQL<sup>15</sup>. But this approach was not sufficient as we experienced a slow data access rate with larger queries. Therefore, we decided to build an abstraction layer which encapsulates the access of the database from the system. The abstraction layer is built with Node.js – a JavaScript framework – and it provides a HTTP API. More details will be provided in Section [sec:architecture].

<sup>14</sup>PostgreSQL JDBC driver: <http://jdbc.postgresql.org/>, last accessed March, 28 2014

<sup>15</sup>MySQL JDBC driver: <https://dev.mysql.com/downloads/connector/j/>, last accessed March, 28 2014

## Trajectory Generation

The trajectory generation group consists of two systems. First the trajectory service which is generating trajectories for subscribers and second the routing engine which is responsible for calculating routes between a start and end point.

## Architecture

After a brief overview has been given in the previous section, here the underlying system will be explained in more detail. It will also cover the process of importing the needed data into the database and how this data can be retrieved from the database. Moreover the used data and the communication between the subsystems will be described.

### PostgreSQL Database

To store geographic information PostgreSQL together with the PostGIS extension is used. PostGIS extends PostgreSQL in a way that it follows the Simple Features for SQL specification from the Open Geospatial Consortium. The Simple Features are defined in ISO 19125 [[@ISO19125](#)][[@ISO191252](#)]. The developed system is using population and land use information stored in the database. This data is needed to estimate a subscribers start and end position within the boundaries of the cell coverage area. In addition to raster data (population and land use) the database also stores the OpenStreetMap road network. Figure [fig:pgstructure] depicts the internal structure of the PostgreSQL database. The table *at2po4pgr* contains the OpenStreetMap road network, *landuseraster* land use information and *densityraster* population information.

at_2po_4pgr		landuse_raster	density_raster
PK	id	PK	rid
		rast	rast
		filename	filename
osm_id			
osm_name			
osm_meta			
osm_source_id			
osm_target_id			
clazz			
flags			
source			
target			
km			
kmh			
cost			
reverse_coste			
x1			
y1			
x2			
y2			
geom_way			

[fig:pgstructure]

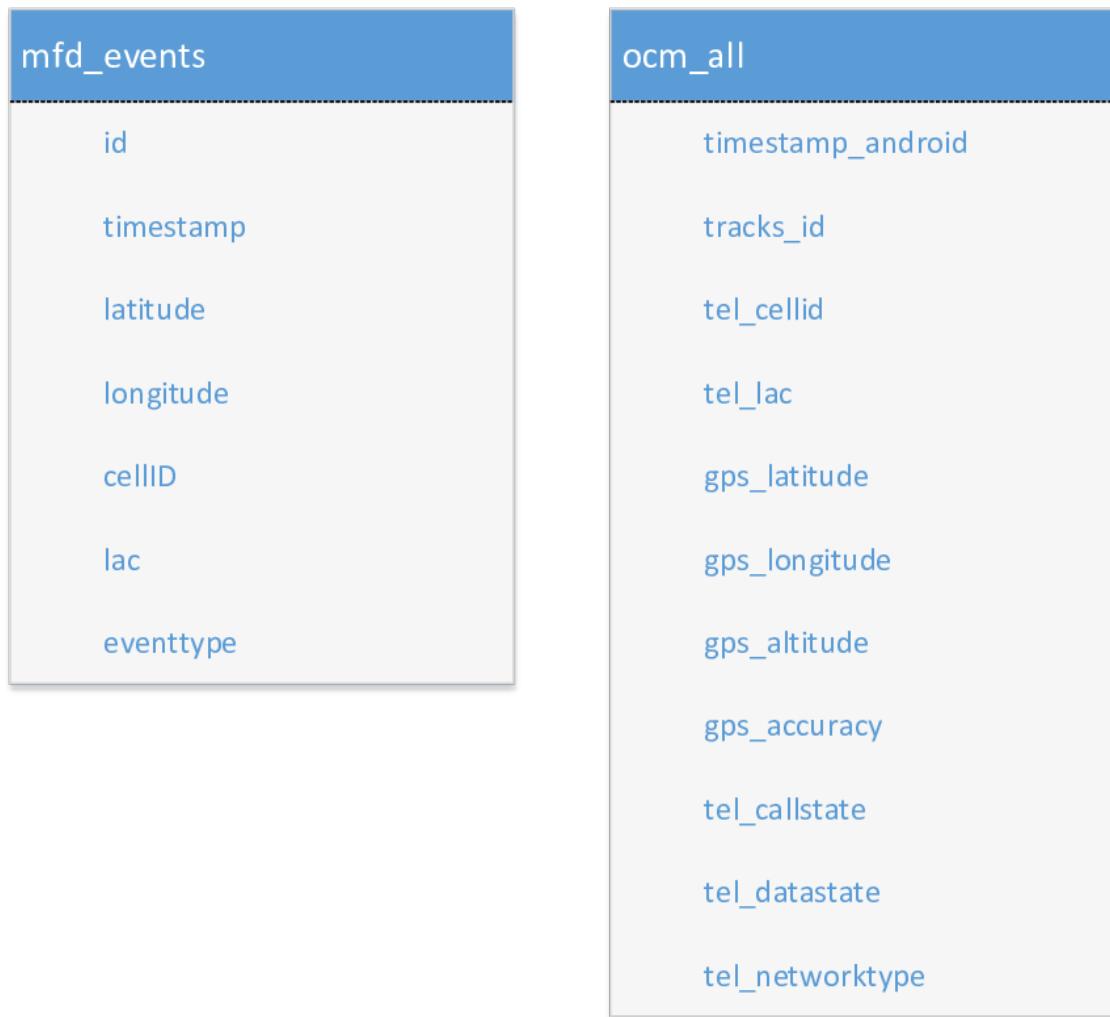
**Import Data** This section describes the process to import both raster and the a road network into the database. Raster data is simply an image where each pixel corresponds to a coordinate in world space. A raster image defines a grid spacing which defines how far in meters pixel are away. A grid spacing of 20 meters means that each pixel covers an area of 400 square meters. Usually environmental and demographic geographic information, is stored as a raster image. On the other hand road networks consists of a vector of points. The road network is stored in the database as Simple Features.

**Raster** To import raster data into the PostgreSQL database the tool `raster2pgsql` was used. Listing [lst:raster2pgsql] illustrates how to import the data. The parameter `s` defines the spatial reference system of the raster image. The grid spacing of the raster image is defined by parameter `t` and is in this case 100 by 100 meters. The tool converts the raster image into an SQL file which can be inserted into the database. To insert the created into the database the second command is used.

**Road Network** Our system is using the routing engine `osm2po` which operates on the OpenStreetMap road network. `Osm2po` can convert the OpenStreetMap road network to a PostgreSQL compatible SQL file. Listing [lst:osm2pgsql] shows the step to import an OpenStreetMap road network – in this case from Austria – into the PostgreSQL database.

## MySQL Database

The developed system is using data from A1 and the OpenCoverageMap project. Because this data is not related with geographic information a second database is used to store this data. This information is stored in two separate tables because the structure of A1 and the OpenCoverageMap is different. Figure [fig:mysqlstructure] depicts the structure of the two tables, the first `mfdevents` stores the A1 data and the second `ocmall` stores information from the OpenCoverageMap project.



[fig:mysqlstructure]

**Import A1** To import the A1 data set into the MySQL, the binary event stream needs to be converted first. The conversion takes the binary event stream and converts its to CSV file. Each row in the CSV file represents an event. Later the created CSV file will be imported into the database similar to Listing [lst:textmysql].

**Convert** The conversion is carried out by a Java program which takes the binary data stream as input and outputs a CSV file. First a Java `InputStream` from the input file will be created. This input stream will be passed to a `DataInputStream`. As illustrated in Listing [lst:parsestream] a loop iterates over the data input stream and parses the binary information according to the A1 data structure. At last at the end of each iteration the parsed information will be print to the standard output.

```
// create file input stream
InputStream is = new FileInputStream("a1.data");
// create new data input stream
DataInputStream dis = new DataInputStream(is);
while (dis.available() != 0){
    byte[] anon_id = new byte[32];
    dis.read(anon_id);
```

```

        timestamp=dis.readInt();
        timestamp_ms=dis.readInt();
        lac=dis.readInt();
        cell_id=dist.readInt();
        latitude=dis.readDouble();
        longitude=dis.readDouble();
        radius= dis.readLong();
        dis.skipBytes(12); //skip reserved fields
        event_type = dis.readShort();
        dis.skipBytes(3); //skip reserved fields
        angle_x=dis.readInt();
        angle_y=dis.readInt();
        speed=dis.readInt();
        //only print useful fields
        System.out.println(Hex.encodeHexString(anon_id) +";"+ timestamp +" ; "+ lac +" ; "+cell_id + " ; "+ radius);
    }

LOAD DATA INFILE 'data.txt' INTO TABLE tbl_name
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n';

```

**Import OpenCoverageMap** The data we got provided by the OpenCoverageMap project was already a CSV file and can be imported into the database with an SQL statement similar to the one in Listing [lst:textmysql].

## Node.js Data API

To encapsulate the querying of the database from the trajectory service a Node.js API was implemented. It provides a HTTP API which gives access to the data stored in the PostgreSQL and MySQL database. The data API is using the following modules, which can be installed through the script provided in Listing [lst:nodemodules]:

- **express**<sup>16</sup> is a web application for Node.js and provides our system with web request routing in a RESTful manner
- **pg**<sup>17</sup> is a PostgreSQL driver for Node.js
- **mysql**<sup>18</sup> is a MySQL driver for Node.js

```

var pg = require('pg');
var conString = "postgres://postgres:1234@localhost/postgres";
var mysql = require('mysql');
var connection = mysql.createConnection({
  host: 'localhost',
  user: 'mysql',
  password: '1234'
});
var express = require('express'),
  app = express.createServer();
app.get('/mysql', function (req, res) {
  connection.connect();
  connection.query('SELECT 1 + 1 AS result', function (err, rows, fields) {
    res.send(rows[0].solution);
  });
}

```

<sup>16</sup>Project website with more information about express: <http://expressjs.com>, last accessed March 28, 2014

<sup>17</sup>More information about the PostgreSQL driver pg: <https://github.com/brianc/node-postgres>, last accessed March 28, 2014

<sup>18</sup>Project website with more information and a tutorial about mysql: <https://github.com/felixge/node-mysql>, last accessed March 28, 2014

```

    });
    connection.end();
});
app.get('/postgres', function (req, res) {
  var client = new pg.Client(conString);
  client.connect(function (err) {
    client.query('SELECT NOW() AS time', function (err, result) {
      res.send(result.rows[0].time);
    });
  });
});
app.listen(3000);

```

**Features** The Node.js provides several features to the trajectory service. It enables the trajectory service to access the two database via an uniform API. This API hides the implementation from the using program. However, to increase performance the API was split into two parts. The first API resides on the same server as the PostgreSQL and handles geographic request whereas the second one handles subscription data and resides on the MySQL server.

**MySQL API** provides access to mobile subscription data for both the A1 and OpenCoverageMap data. In addition, the route traveled by a subscriber can be retrieved from the OpenCoverageMap data. This route consists of a timestamp and the latitude and longitude coordinate. All the returned data is encoded as JSON that eases the parsing. Listing [lst:mysqlapireq] depicts the request that can be made to the database. The first one retrieves events from A1 for the specified user, the second one retrieves events from OpenCoverageMap and the last one get the route traveled by the OpenCoverageMap user.

```

GET /mfdevents?id=USER_ID
GET /ocmraw?id=USER_ID
GET /ocmroute?id=USER_ID

```

**PostgreSQL API** enables the requesting program to access functions of the PostgreSQL database. It provides access to raster data as population and land use information as well as to retrieve a random point within a specified area based on population and land use information (see Listing [pgapireq]). The bounds parameter is encoded as a list of coordinate – separated by a comma – specifying a bounding box (xmin,ymin,xmax,ymax). In comparison the geom parameter is encoded as Well-known text (WKT) which allows defining more advanced areas as polygons and multi polygons (POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))).

```

GET /populationBounds?bounds=BOUNDARIES
GET /landuseBounds?bounds=BOUNDARIES
GET /randomgeom?geom=GEOMETRY&num=NUMBER_OF_POINTS

```

## Routing Engine

To calculate routes on the OpenStreetMap road network, the system is using the osm2po routing engine. This engine can either be used as a standalone web server with a defined API or be integrated into a Java application. In our case the engine was integrated into the trajectory service as a library. The benefit of this approach is, that the system has better access to the graph file used by the engine. The graph file is a reduced road network optimized for routing. By accessing the graph file, the trajectory service can retrieve metadata for the calculated route such as speed limits and length of the roads. This information can later be used to validate the calculated route and the timing information. Listing [lst:osm2po] illustrates how a route between Linz and Hagenberg im Mühlkreis is calculated. The result is an array with segment IDs that can then be used to retrieve the geometry and the speed limits of the segment from the graph file.

```

File graphFile = new File("at_2po_4pgr.gph");
Graph graph = new Graph(graphFile);
DefaultRouter router = new DefaultRouter();
//Linz, Austria
int sourceId = graph.findClosestVertexId(48.306f, 14.285f);
//Hagenberg im Mühlkreis, Austria
int targetId = graph.findClosestVertexId(48.368f, 14.513f);
Properties params = new Properties();
params.setProperty("findShortestPath", "false");
//calculate path between source and target
int[] path = router.findPath(
    graph, sourceId, targetId, Float.MAX_VALUE, params);

```

## Trajectory Service

The trajectory service is using the above mentioned data and APIs to generate trajectories for each subscriber. By using different data sources it can estimate trajectories for different data providers in our case mobile network operators. In the current state trajectories can be generated for A1 subscriber and OpenCoverageMap participants. To fulfill its task the trajectory service needs besides the events from the network operator a coverage prediction for each cell site. The trajectory service is using the following data:

- Mobile subscription data
- Population and land use information
- Cell site coverage prediction
- OpenStreetMap road network
- osm2po route engine

**Convert Events** The structure of events used in the trajectory generator was inspired by the structure used in the A1 event stream. To integrate other data sources such as the OpenCoverageMap a converted needs to be implemented. The objective of the converter is to transform the event structure to the one used by A1 events. The OpenCoverageMap doesn't use the same event structure as A1. Rather than issuing events like call establish / termination or hand over it periodically stores the state of the mobile station. Therefore, this information needs to be transformed to an event structure. The steps to convert the OpenCoverageMap data to the A1 structure are the following: First remove all data where no call is active. Second iterate over this data and check for gaps between succeeding timestamps. Gaps greater than 10 seconds indicate that a call was either established or terminated. Afterwards iterate over the data between two gaps. Create a call establish and termination event at the beginning and end of the list. If there is a difference between the cell-id and location area of two succeeding entries add a hand over event to the current call. The result will be a list of events with the same structure as the ones captured by A1.

## Trajectory Service

The generation of subscriber trajectories involves a number of steps which have to be carried. In this section, an overview of all processes involved in the generation will be explained. In general, the processes can be divided into 4 categories: *data retrieval, processing, validation, presentation*.

### Start and End Position

### Route Validation

### Geometry

**Timing**

**Timing Information**

**Handover Prediction**

## **Results**

**Routing**

**Timing**

**Rural**

**Cluster**

**Cell**

**Hybrid**

**Urban**

**Cluster**

**Cell**

**Hybrid**

**Highway**

**Cluster**

**Cell**

**Hybrid**

**Handover Positions**

**Discussion**

**Summary**