# An Architecture for Acquisition and Provision of Hotspot Coverage Information

**Stephan Lück[1], Michael Scharf[1], Jorge Gil**

**University of Stuttgart, Institute of Communication Networks and Computer Engineering**
**Stuttgart, Germany**
**e-mail: {lueck, scharf}@ikr.uni-stuttgart.de**

*Abstract* - **Cellular mobile networks will be more and more complemented by high speed wireless hotspots. In order to use this additional access technology efficiently, terminals must be able to discover the wireless hotspots located around them. Classical methods based on measurements on the physical network interface are not sufficient because they neither can detect neighbouring hotspots nor are they well-suited for multi-mode interfaces, which can use only one access technology at once. Therefore, we suggest to use a model-based access discovery method, which applies queries to a location-based data model. In this paper, we describe how to collect field strength measurement from mobile terminals, how to convert them to polygons describing the coverage of wireless hotspots, and how to retrieve the coverage information.**

## 1. Introduction

Cellular wireless networks offer an excellent coverage even in rural regions because of their large cell sizes. However, the data rates of cellular networks are low and are insufficient for many data applications. That is why operators started to install additional radio access technology at areas where a large number of potential customers is present – so-called *hotspots*. Hotspots typically use wireless LAN technology based on IEEE 802.11, which offers high data rates, but the cells are much smaller than those of classic cellular networks rendering large coverage impossible. A a result, in may locations the coverage of cellular networks and hotspots overlap. Consequently, mobile terminals need to select the best available access network; they should be *always best connected* [1], [2].

In order to detect access networks, all suitable radio interfaces must be periodically scanned (*measurement-based access discovery*). This has two major drawbacks: First, frequently scanning for new networks consumes a significant amount of energy. Second, measurement-based network discovery can only find networks which are available at the current position. But, for example in a city center, hotspots may be available just 100m from the current position. If the network access there is "better" than at the current position (in terms of available bandwidth, or price), many users might be willing to move the get better Internet access. Providing such information requires that the position of hotspots be stored in databases and can be queried by users, which are interested in finding an optimal network access. In this paper we describe an architecture for such a system.

[1])This work was funded by the German Research Foundation (DFG) through the Center of Excellence (SFB) 627 "Spatial World Models for Mobile Context-Aware Applications"

In related literature, authors also have observed the need of information systems, which can provide terminals with information about the coverage of access networks. The Mirai project [3] proposes to manage user mobility by means of so-called resource servers. These servers have information about the coverage of access networks. In addition, in [4] and [5], the authors found that terminals, which use a certain configuration on their radio interfaces, cannot scan all supported frequency ranges while communicating. Therefore they suggest to supplement scanning procedures with database queries. However, [3] does not address the data acquisition problem and in [4] and [5] a solution for taking and distributing measurement values is discussed but methods of the further processing of these values are not covered.

A system that stores hotspot coverage is only useful if a sufficiently high number of access networks is included and if the stored information is accurate. In principle, there are two sources of information: the network operators and the users. Network operators have a profound knowledge about their access points and network coverage and might be willing to announce this to attract users. The users also get information about hotspots by monitoring their radio interfaces. However, users do not have a global view of the network, i. e., they only know which networks are currently available at their location. Accurately predicting the coverage of hotspots is therefore only possible by collecting measurement from many users. Our goal is an architecture that collects such distributed measurements, processes them and determines the corresponding cell geometries.

Retrieving coverage information must not be too resource-consuming. Thus, the corresponding data records must be compact. Therefore, we convert the field strength distribution of a radio cell to a polygon describing the boundary of the cell. A mobile terminal which queries coverage information then does not obtain field strength values but only the cell location, which is absolutely sufficient to perform access selection decisions and the corresponding data records are small and simple to process.

This paper is structured as follows: In Section 2, we describe the main building blocks of our architecture, which are responsible for data acquisition, data processing, and data storage and retrieval respectively. Section 3 studies more in detail how measurement data has to be processed in order to determine cell geometries and Section 4 shows how the cell geometries are stored and how mobile terminals retrieve the data. Section 5 concludes the paper.

## 2. Architecture

In this section, we describe our proposed architecture. We highlight three basic aspects of an architecture that must be able to gather and provide hotspot information: *data acquisition*, *data processing*, and *data retrieval*.

### 2.1 Data Acquisition

Concerning data acquisition, we observe two approaches. First, data about radio coverage can be obtained by collecting field strength values measured by mobile terminals. We call this approach *distributed data acquisition*. Second, network planning systems, which cellular network operators might be using could make available field strength values. Because this approach typically uses a single network planning database, we call it *central data acquisition*. Figure 1 illustrates both approaches.

In the rest of this section, we describe the more challenging distributed data acquisition approach in more detail.

The mobile terminal has one or more *radio interfaces* for different radio access network technologies, e. g., both WLAN and GPRS/UMTS. The radio interfaces periodically monitor available access networks and report them to a *communication middleware* running on the terminal. The reported information must include a unique identifier of the access network or access point, for instance the MAC address of an WLAN access point, the measured radio link quality, e. g. given as a field strength or signal to noise ratio, and the access network technology. Furthermore, the terminal must determine its position. For outdoor usage, this can be achieved by a satellite-based positioning system such as (D)GPS or Galileo. Indoors, other positioning methods could be used.

The communication middleware (see Figure 2) converts the measurement values to a standardized format. It is important to compute the values in such a way that they can be compared to each other even when they are generated by different terminals. For WLAN, we use the signal-to-noise ratio in dB as a standardized format.

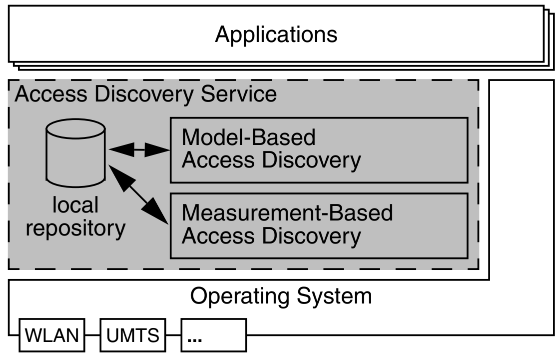


**Figure 2:** Mobile Terminal Software Architecture

After this *preprocessing*, the middleware stores the normalized measurement values, along with the position where they have been observed, in a *local measurement repository*. Note that typically the field strength can be measured much more frequently than the position. For instance, GPS receivers usually perform about one measurement per second whereas in Wireless

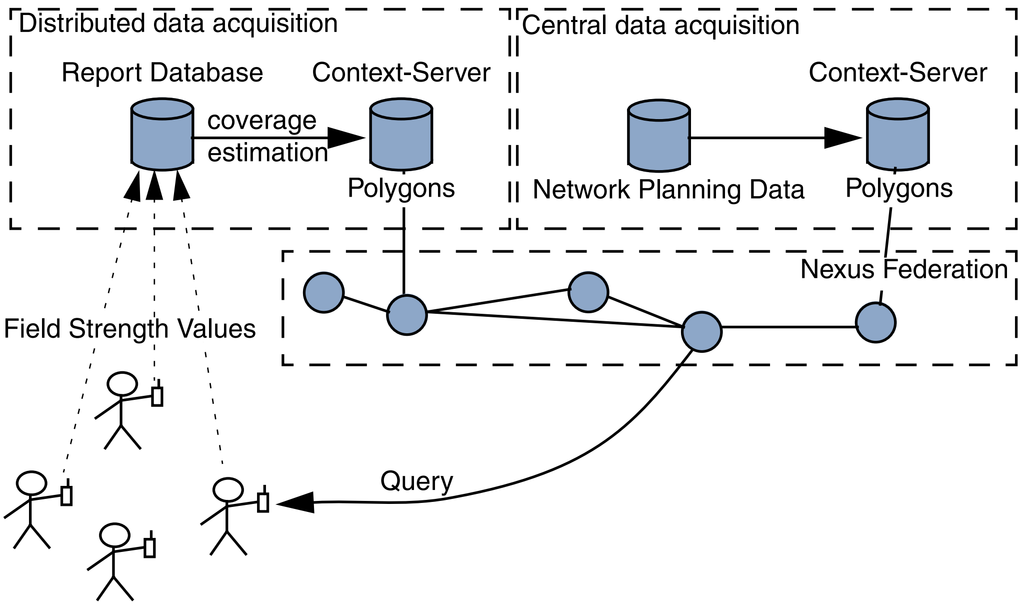


**Figure 1:** Architecture of a system for provision of radio cell coverage information based on the Nexus-Information-System

LANs the field strength typically can be determined ten times per second. Therefore, several values might be stored for one position.

The content of the local measurement repository, i. e. records of the form (technology, access point ID, normalized quality, position), is uploaded to a service center of the *hotspot coverage provider*. Both for efficiency and privacy reasons, it does not make sense to transmit each individual measurement immediately. Instead, the measurements are preprocessed and anonymised and only uploaded if there is good connectivity (i. e., if there is cheap and/or fast access network available). The hotspot provider collects all uploads and stores them in a *report database*.

## 2.2 Data Processing

All data stored in the report database is then periodically processed by a *coverage prediction* component. In order to predict the cell geometries, the following steps are necessary for each hotspot:

1. To simplify processing, all measured values should be aligned to a grid with a certain granularity. In indoor environments, a value of one meter might be an appropriate choice, while for outdoor usage a larger value might be sufficient.
2. In the optimal case, there should be exhaustive measurements. In practice, however, even if a lot of measurements are available, *interpolation* is required to estimate the field strength at locations where no measurements are available. Furthermore, erroneous data should *filtered* out.
3. The interpolated raster data can then be converted into polygons by a *vectorization* algorithm.
4. Simple vectorization algorithms yield polygons with a rather large number of vertices and edges. In order to efficiently store and transmit polygons, it is highly desirable to have rather simple descriptions of cell boundaries, i. e., polygons with only few vertices. Therefore, a *simplification* algorithm (see Section 3.3) has to be applied to the result of the vectorization. There is a trade-off between simplicity and accuracy.
5. Finally, the obtained polygons can be stored in a *coverage database*.

The coverage database can then be queried by the users. For instance, this can be done by some kind of Web front-end. But, of course, this can also be done automatically by a *handover decision component* within the *communication middleware*.

## 2.3 Data Storage and Retrieval

Our architecture uses the Nexus [6], [7] context-based and location-based information system to store and retrieve hotspot coverage information. The Nexus platform is an open system; multiple providers can make context-based information available. This information is stored in so-called *context servers*. Nexus has a *federation tier*. To retrieve data, a *Nexus client*, which is typically running on a mobile terminal, sends a query to the federation. The federation then relays the query to all context servers which probably can answer this query, collects the responses from the context servers, merges the data to a single response and finally sends the merged data back to the client.

Within our architecture, we realise the coverage database by means of a context server. When a mobile terminal wants to discover the access networks within a certain area, say 100 meter around its current position, it sends a query to the federation. Then the federation forwards the query to the context servers, which are responsible to answer queries on this geographical area and sends the answer back to the terminal. There are two cases to be distinguished. If there is only one server responsible for the given location, the federation just forwards the response. In the more complex case that more than one server provides data, as shown in Figure 1, the federation must be able to merge the data from different servers i.e. concatenating polygons for different cells and merging polygons for the same cell.

Therefore, two functions are necessary in the federation. First, identical cells need to be identified. This can be realized by introducing globally unique cell identifiers into the data records stored in the context servers. These identifiers can quite easily be formed using the technology specific cell identifiers used by almost all radio technology. Second, polygons describing the same radio cell need to be merged. To this end, appropriate merging algorithms need to be investigated.

# 3. Data Processing Algorithms

Field strength values gathered by one or both of the afore mentioned data acquisition approaches is a set containing tuples of coordinates, field strength values, and cell identifiers. Selecting tuples containing a certain cell identifier allows for calculating the geometry of each cell independently from any other cell. Concerning the algorithm that calculates a cell geometry, it is sufficient to consider one cell at once. In the following, we describe the steps we use to calculate polygons, which describe cell geometries.

## 3.1 Interpolation and filtering

The algorithms we use for calculating radio cell polygons require so-called raster data. Raster data is characterised by a regular grid of field strength values, which are equidistant in both x and y dimension. Often the grid spacing is too large to apply vectorisation algorithms. Then the grid spacing should be made smaller by generating additional field strength values using an interpolation algorithm. After interpolation, blurring filters, which are known from image processing, can be applied to the raster data. They cause the field strength values to be averaged.

## 3.2 Vectorization

The main goal of our algorithms is to convert raster-data to vector data. This makes data processing within mobile terminals looking for a access network simpler and reduces the amount of data to be transmitted over the wireless network.

In order to vectorise a field strength distribution, we calculate *contour lines* of a field strength distribution. A contour line is the set of points where the field strength has a given constant value. Furthermore, we define a threshold for the field strength values. If a value is above the threshold, it is assumed that a terminal would be able to use the cell; if the value is lower than the terminal could not connect to the cell. Thus we can obtain the boundary of a radio cell by calculating the contour lines corresponding that threshold. Note that the actually used threshold value depends on the considered technology (WLAN or UMTS/GPRS) and on the standardized format (field strength or signal-to-noise-ratio). Also, in reality this threshold cannot be defined exactly, there is more likely a range of field strength where terminals might or might not to be able to communicate. However, due to the limited extent of hotspots, the error by using a single threshold is small.

Calculating contour lines, i.e. vectorising raster data, can be done in a quite straightforward way. The field strength values are interpreted as Z coordinates in a three-dimensional space, whereas the grid spacing defines the corresponding X and Y coordinates. Then for each group of four neighbouring points (x, y, z) a plane surface is calculated by linear approximation. Contour lines within one approximated surface are straight lines. The contour line of the hole grid is obtained by connecting these lines.

## 3.3 Selection and Simplification

In the following, we consider the case of a single polygon only. In practice, it might be necessary to use more complex structures to describe a cell. For example, a cell consisting of areas, which are not connected must be described by shapes built-up of discontiguous polygons – so-called multipolygons.

Vectorization by calculating contour lines may yield multiple contour lines. Therefore, a selection step is required to determine the contour lines to be used in the further processing. In many cases it might be sufficient to consider the contour line that encloses the largest area only.

After the selection step, we obtain a single contour line, which describes the cell boundary. This contour line can easily be converted into a polygon. The only remaining problem is that this polygon may be rather complex so that a high number of points is needed for its definition and a data record representing this polygon may be large. Therefore, we perform a *simplification* step in order to reduce the number of points in order to obtain a more compact data record.

To perform simplification, we consider three different algorithms. A simple distance-based algorithm, an algorithm based on the Euclidean distance, and a slope-based algorithm. All of these algorithms go sequentially along the vertices and remove some of them.

The simple distance-based algorithm places a tolerance rectangle around each vertex. When iterating over the vertices of the polygon, the algorithm removes all vertices but the first one located in a tolerance rectangle; for each vertex P of the polygon the algorithm removes all vertices Q for which

$$|P_x - Q_y| < tolerance_x$$

and

$$|P_y - Q_y| < tolerance_y$$

holds.

The simple distance algorithm can be modified by calculating the Euclidean distance between the vertices. For each vertex *P* of the polygon the algorithm removes all vertices *Q* for which

$$\sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2} < d_{tolerance}$$

holds. As shown in Figure 3, the tolerance rectangles have thus been replaced by tolerance circles.

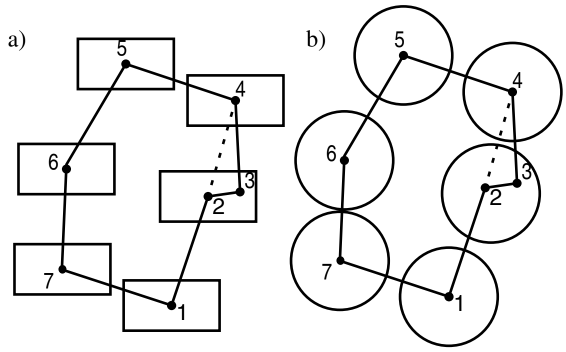


**Figure 3:** Simplification of a polygon by removing the third vertex. a) simple distance-based algorithm b) Euclidean distance-based algorithm.

A different approach is followed by a slope-based algorithm. The slope-based algorithm subsequently visits every vertex of the polygon. At each vertex *P* it compares the slope $s_{P,1}$ of the edge between *P* and the subsequent vertex $Q_1$ with the slope $s_{P,2}$ of the line between *P* and the vertex $Q_2$ following $Q_1$ (see Figure 4). If the slope difference is below a given tolerance value then it removes $Q_1$. Then it compares $s_{P,1}$ with the slope $s_{P,3}$ of the line between P and vertex $Q_3$, which is the vertex following $Q_2$ and so on. Eventually it will find a vertex $Q_n$, for which the difference $|s_{P,n} - s_{P,1}|$ is greater than the tolerance value. Then it sets $P := Q_n$ and performs the same operations on the new vertex *P*.

Simplification by removing vertices not only makes data records representing polygons more compact but also introduces an error because the original polygons and the simplified ones are not identical. Actually, a trade-off between accuracy and the size of data records must be found. Therefore, we introduce two measures

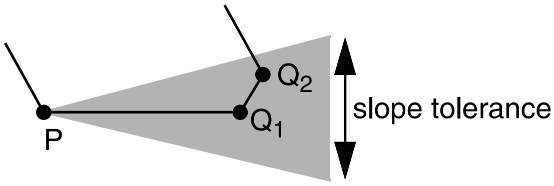


**Figure 4:** slope-based simplification. The vertex $Q_1$ is removed because $Q_2$ is within the extent of tolerance.

of the error introduced by simplification. Figure 5 shows a part of an unsimplified polygon (solid line) and one edge of the corresponding simplified polygon (dotted line), which has been obtained by removing the vertices $p_{i+1}, .. , p_{j-1}$. The first error measure we use is the *mean distance* between the vertices of the original polygon to an edge of the simplified one:

$$E_{dist} = \sum_{m \in M} \frac{1}{j-i-1} \sum_{k = i(m)+1}^{j(m)-1} d_k$$

where $M$ denotes the set of edges of the simplified polygon and $i(m)$ and $j(m)$ stand for the indices of the starting and ending vertex of the corresponding edge respectively.

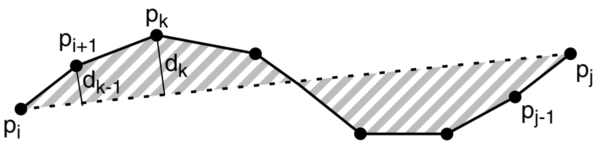


**Figure 5:** Error measures. The solid line is a part of the original polygon, the dotted line is an edge of the simplified one.

The second error measure is the *area* enclosed by the boundaries of the original and the simplified polygon (see Figure 5, hatched area).

We evaluated the three simplification algorithms using a polygon with 39 vertices as shown in Figure 6. These algorithms have been applied to this polygon with augmenting tolerance values. Thus simplified polygons with different numbers of vertices have been obtained. For example, the polygon drawn with dashed lines in Figure 6 has been calculated by reducing the number of vertices to 10 by means of the Euclidean distance-based algorithm.

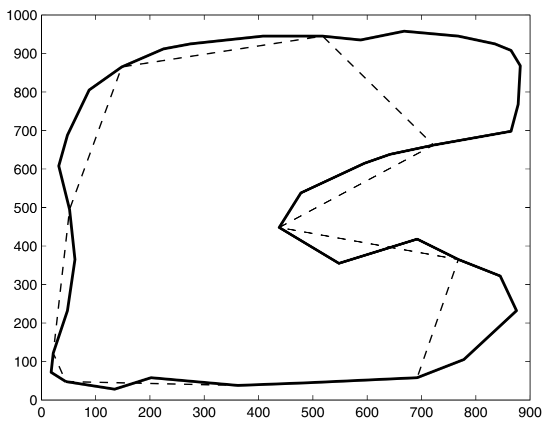


**Figure 6:** Reference polygon and simplified polygon

Table 1 shows the simplification error of the different algorithms determined by both distance-based and areas-based error measures. The distance-based algorithms perform substantially better than the slope-based algorithm. Furthermore, the error increases superproportionally with the number of reduced vertices.

| vertices | simple distance | | euclidean distance | | slope | |
|---|---|---|---|---|---|---|
| | area | dist | area | dist | area | dist |
| 39 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0.04 | 17 | 0.04 | 17 | 0.05 | 37 |
| 15 | 0.1 | 58 | 0.08 | 65 | 0.1 | 86 |
| 10 | 0.2 | 95 | 0.3 | 120 | 0.3 | 151 |
| 5 | 0.5 | 217 | 0.48 | 237 | 0.8 | 296 |

Table 1: simplification error

# 4. Data Storage and Retrieval

In this section, we describe how mobile terminals can retrieve data stored in context servers.

## 4.1 Data Storage and Representation

Nexus uses the *Augmented World Model (AWM)* [7] to represent context-based and location-based information. This object-oriented data model allows to group of similar types of objects together to one class. For instance, radio cells are modelled as objects of the type "RadioCell". Furthermore, all these objects are derived from the class "SpatialObject". Therefore, they have inherited a *position* and an *extent* attribute. The position should be a point within the radio cell, for example the location of the access point or a base station. To the extent attribute of a radio cell the polygon describing the coverage is assigned. To specify both position and extent the AWML relies on the *Geographic Markup Language (GML)* [8]. At last, all static AWM objects have a unique textual identifier called *Nexus Object Locator (NOL)*. For textual representation of AWM object documents the XML-based *Augmented World Modelling Language (AWML)* [9] is used. When a radio coverage algorithm has determined a new polygon, it generates an appropriate AWML document and uploads it to a context server. Figure 7 shows the AWML specification of a radio cell.

## 4.2 Data retrieval

Mobile terminals use the *Augmented World Query Language* [9] to ask context servers for information about access network data. Among other things, this language allows to specify queries which contain restrictions on the type of the object and the geographic location of the object. For our application, the terminal typically would restrict the query to objects which describe radio cells. In the AWM, such object have the type “RadioCell”. How spatial restrictions are formulated depends on the user’s situation. When the user is looking for a hotspot and would accept to move to get access to a hotspot then the query is restricted to the area of several meters around the current position of the user so that neighbouring hotspots can be discovered. When the user cannot move or does not want to do so then the query is restricted to the current position of the users. Such a query would yield access networks which are available at the user’s current location only. After all, it is also possible to retrieve data about hotspots, which are at a absolutely different location than the user. This allows for scheduling communication activities if the future locations of the users are known or can at least be predicted with a sufficient probability. An example for an AWQL-Query is shown in Figure 8.

## 4.3 Challenges

Concerning our approach to convert filed strength values to cell polygons there are still research challenges affecting both measurement accuracy and privacy.

Because measured field strength values depend on the terminal type and configuration two different terminals might provide different data. This can be circumvented by a careful calibration of all devices. However, since for us it is sufficient to have a rough estimation of

```
<awml:awml>
   <awml:nexusobject>
      <nsas:type><nsas:value>RadioCell</nsas:value></nsas:type>
      <nsas:NOL><nsas:value>...some NOL...</nsas:value></nsas:NOL>
      <nsas:extent>
         <nsas:value>
            <nsas:gml>
               ...GML representation of the cell polygon...
            </nsas:gml>
         </nsas:value>
      </nsas:extent>
      <nsas:pos>
         <nsas:value>
            <nsas:gml>
               ...GML representation of the access point location..
            </nsas:gml>
         </nsas:value>
      </nsas:pos>
   </awml:nexusobject>
</awml:awml>
```

**Figure 7:** An AWML document describing a radio cell

```
<awql:awql>
   <awql:restriction>
      <awql:and>
         <awql:equal>
            <awql:target>nsas:type.nsas:value</awql:target>
            <awql:referenceValue>
               RadioCell
            </awql:referenceValue>
         </awql:equal>
         <awql:intersects>
            <awql:target>nsas:extent.nsas.value</awql:target>
            <awql:referenceValue>
               <nsat:gml>
                  ...GML representation of the area, where
                  the terminal is looking for hotspots...
               </nsat:gml>
            </awql:referenceValue>
         </awql:intersects>
      </awql:and>
   </awql:restriction>
</awql>
```

**Figure 8:** An AWQL-Query for hotspot radio cells within a given area

the boundaries of a hotspot, it is sufficient to use uncalibrated measurement values.

The location information is not exact. In this application scenario, it is sufficient to store position information with a granularity of one or even several meters. However, current positioning systems do not necessarily provide a sufficiently high accuracy. Therefore, methods should be developed taking into account the limited accuracy of the position.

The fact that in principle any user of a mobile terminal can enter data into a report database leads to a security problem. Malicious users or organisations can attack the system by inserting wrong filed strength values, which causes the cell polygons to be distorted compromising the usability of the system. How to detect such inconsistencies is currently under investigation in Nexus. One potential solution is to use a reputation system, which allows for branding malicious users. However, when using a reputation system, users cannot remain completely anonymous. But users might not be willing to publish measurement by their real name because user reports can be used to generate traces and user profiles. A solution of this trade-off between user privacy and usability of the system could be that users apply virtual IDs (pseudonyms) [10] instead of their real names.

## 5. Conclusions

We have described an architecture, which allows mobile terminals to detect access networks located around their current position by querying a context-based information system. We argue that this model-based approach of network discovery complements the classical measurement-based approach insofar as neighbouring hotspots can be detected and terminals that can use only one access technology simultaneously can discover access networks of different technology. Data retrieval is based on the Nexus platform so that multiple providers of coverage information could be supported. We have described a method to convert individual field strength values to polygons, which describe the coverage of radio cells.

# References


[1] Eva Gustafsson and Annika Jonsson: *"Always Best Connected"*, IEEE Wireless Communications February 2003, pp 49–55

[2] Gustavo Carneiro, José Ruela, Manuel Ricardo, *"Cross-Layer Design in 4G Wireless Terminals"*, IEEE Wireless Communications, April 2004, pp 7–13

[3] Masugi Inoue, Khaled Mahmud, Homare Murakami, Mikio Hasegawa, Hiroyuki Morikawa, *"Novel Out-of-Band Signaling for Seamless Interworking Between Heterogeneous Networks"*, IEEE Wireless Communications, April 2004, pp 56–63

[4] Matthias Siebert, Marc Schinnenburg, Matthias Lott, Stephan Göbbels, *"Location Aided Handover Support for Next Generation System Integration"*, The 5th European Wireless Conference, February 2004, pp 195-202

[5] Matthias Siebert, Marc Schinnenburg, Matthias Lott, *"Enhanced Measurement Procedures for Vertical Handover in Heterogeneous Wireless Systems"*, The 14th IEEE International Symposium On Personal, Indoor And Mobile Radio Communications, September 2003, pp 166–171

[6] Center Of Excellence *"Spatial World Models for Mobile Context-Aware Applications"*, Universität Stuttgart, `http://www.nexus.uni-stuttgart.de`

[7] Daniela Nicklas, Matthias Großmann, Thomas Schwarz, Steffen Volz, Bernhard Mitschang, *"A Model-Based, Open Architecture for Mobile, Spatially Aware Applications."*, Proceedings of the 7th International Symposium, SSTD 2001, Redondo Beach, CA, USA, July 2001, p 117

[8] Simon Cox, Paul Daisey, Ron Lake, Clemens Portele, Arliss Whiteside, *"Geography Markup Language (GML3.0)"*, January 2003, `http://www.opengeospatial.org/specs/?page=specs`

[9] Martin Bauer, Frank Dürr, Jan Geiger, Matthias Grossmann, Nicola Hönle Jean Joswig, Daniela Nicklas, Thomas Schwarz, *"Information Management and Exchange in the Nexus Platform"*, Technical Report, Universität Stuttgart, April 2004, `http://www.nexus.uni-stuttgart.de`

[10] Christian Hauser, *"Mobility Management Meets Privacy - the Failure of Existing Proposals and a New, Future-Proof Approach",* Proceedings of the Second International Workshop on Mobility Management & Wireless Access Protocols, October 2004, pp 122–124