ABSTRACT

Context aware services need context of a certain quality to provide their services. The quality of context can be described using the Quality of Context (QoC) concept. There are five parameters, or dimensions, to describe the QoC: precision, freshness, temporal resolution, spatial resolution and probability of correctness. There is a relation between QoC and privacy sensitivity, because if the QoC is higher, the privacy sensitivity is typically higher. To influence the privacy sensitivity, context can be obfuscated in one or more QoC dimensions. We identify different categories of context. For each category the importance of the QoC parameters on the privacy sensitivity is examined. Obfuscating the probability of correctness gives the user plausible deniability and reduces the privacy sensitivity of context in all categories. Furthermore, obfuscating context in the precision dimension reduces the QoC in all categories.

Keywords
Context aware, Quality of Context, Obfuscation, Privacy, Categorization

1. INTRODUCTION

A context aware service is a service which uses context to adopt its behavior to a changing user environment. For example, when the application automatically puts the phone on silence when a user is in a meeting, except for when his pregnant wife calls. Context aware services can be of great use for many people. They are able to aid people in their business, providing them with feedback, information etc, usually without the user explicitly having to make a request.

With these novel type of services, new privacy threats have arrived. How these privacy threats are addressed will have great impact on the success of context aware services. If users experience a few privacy violations, this can lead to distrust in context aware services. As a result of this distrust they could refuse to further use any context aware services. Thus privacy protection is needed for a user to be able to trust a context aware service [1, 2].

Privacy has different meanings to different people in different situations. Therefore, there is not one privacy definition which is widely adopted [3]. Westin [4] stated that privacy is "the claim of individuals, groups or institutions to determine for themselves when, how and to what extent information about themselves is communicated to others". For this paper we use this definition, since it describes the risk of the new context aware services. That is, to what extent can context be used for purposes the user did not intend it to be.

Within the scope of this paper we are concerned about the context on human users. Sheikh et al. [5] state that user context is "Information that describes the situation of a human user either directly or indirectly". We use this definition for user context, because we are concerned about a user using a context aware service and the information that is communicated about this user. This information can either be direct, for example the location of a user, or indirect, for example the bandwidth available to the user's mobile device.

The information that describes a situation of a human can be of varying quality. Quality of Context (QoC) is the concept that describes the quality of this information. Sheikh et al. [5] state that there are several parameters of context which can be used to measure the QoC namely: precision (Pr), freshness (Fr), temporal resolution (TR), spatial resolution (SR) and probability of correctness (PoC).

There is a relation between QoC and privacy sensitivity of context. Our definition of privacy describes that individuals should be able to determine how much information is used and to what extent. When reducing the QoC, by reducing one or more QoC parameters, less information is provided.

This reduction can, for example, be done by using obfuscation techniques. Obfuscation techniques make data contain less information. For example, obfuscation can be achieved by reducing the significant figures of a number.

This results in the question "Which Quality of Context parameters are suitable for obfuscation and how should this obfuscation be done?". To answer this question we first identify different categories of context in section 2. How the QoC concept works is explained in section 3. After that we investigate which obfuscation methods are known for which data type in section 4. Then we make our contribution, by combining the QoC concept with the different categories of context, section 5. This illustrates which QoC parameters of context should be obfuscated. Then we know in which dimension we should obfuscate. To see how to do this, we describe each QoC parameter and explain how obfuscation in this dimension can be done, section 6. In section 7 we show how the gained insights can be used. Related work is discussed in section 8. We conclude with the conclusion in section 9. Future work is discussed in section 10.

2. CONTEXT CATEGORIZATION

There are various ways to categorize context. Here we describe the categorization method which is most suitable for our pur-
poses. The other categorization methods we identified in our re-
search can be found in appendix A.

Dey et al. [6] propose to divide context into a two-tiered sys-
tem. The first tier consists of four categories, namely: location,
identity, time and activity. The second tier consists of all types
of context, this tier has a relationship with the first tier. Context
can either have a relationship with one primary context or have
multiple. An example of a single relation is a phone number: if
a person’s identity is known, we can retrieve his phone number.
A relation with multiple primary context categories is needed for
e.g. a weather forecast since a time and a location are needed to
predict the weather.

The categories proposed by Dey can be reformulated into four
`W` questions, see table 1. This reformulation must not be taken
too literally. For example, a service requests an email address
than this information is also related to the person’s identity and
thus it is a request from the identity category. Another example
is, who someone’s friends are. This is also related to identity.

<table>
<thead>
<tr>
<th>Dey</th>
<th>W question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>Who?</td>
</tr>
<tr>
<td>Location</td>
<td>Where?</td>
</tr>
<tr>
<td>Time</td>
<td>When?</td>
</tr>
<tr>
<td>Activity</td>
<td>What?</td>
</tr>
</tbody>
</table>

We use this categorization method for two reasons. First, the four
categories reflect four ‘basic’ questions. We expect that for these
questions different QoC parameters are important. Second, the
others can be derived to Dey et al., see Appendix A.

3. MEASURING QUALITY OF CONTEXT

Sheikh et al. [7] state that there are five different parameters
indicating Quality of Context. These parameters are Precision ,
Freshness, Spatial Resolution, Temporal Resolution and Prob-
bility of Correctness. Their paper provides reasoning why other
QoC concepts are not as good as theirs. In our research we did not
identify more QoC concepts than Sheikh et al. [7]. The following
sections explore the QoC parameters.

3.1 Precision

Precision is defined as: `granularity with which context informa-
tion describes a real world situation`.

This thus represents how precise context is. For example, the
GPS coordinate 52.0349426269531, 6.34299325942993 defines
a square on the earth with an edge of less than a meter, while the
GPS coordinate 52.0, 6.3 defines a square with an edge of more
than 15 km [8].

3.2 Freshness

Freshness is defined as: `the time that elapses between the deter-
mination of context information and its delivery to a requester`.
For example, if an activity stopped at May 8th and the informa-
tion is requested on May 10th the freshness is 2 days.

3.3 Spatial Resolution

Spatial resolution is defined as: `the precision with which the phys-
ical area, to which an instance of context information is applica-
tible, is expressed`.

This area can for instance, be a room a building or a city. If there
is less physical space included in the area the spatial resolution is
higher, and vice versa. This is illustrated in figure 1.

3.4 Temporal Resolution

Temporal resolution is defined as: `the period of time to which a
single instance of context information is applicable`.
For example, Chris left his work at 16 : 32 the resolution is in minutes, however
if we only know that he left his work on May 9th the resolution is in days. Parallel to spatial resolution, a temporal resolution in
days is lower than in minutes.

The difference between temporal resolution and freshness is sub-
tle. Freshness is, as said, the time that is elapsed between deter-
mination of the context and delivery to the requester. If some-
thing is measured every 30 seconds the temporal resolution is 30
seconds. When this information is requested later, the freshness
depends on, how much later the information is requested. The
temporal resolution remains 30 seconds. This is illustrated in fig-
ure 2.

3.5 Probability of Correctness

Probability of correctness is defined as: `the probability that an
instance of context accurately represents the corresponding real
world situation, as assessed by the context source, at the time it
was determined`.

For example, the location of a person is re-
quested. This person has just logged into a building with a secu-
ritry pass, but his telephone is located at home. When his location
is requested, it is not 100% sure where he is located.

4. OBfuscATION TECHNIQUES

This section explains different obfuscation techniques. To see
what kinds of obfuscation are possible we first draw a scenario
about Chris and Alice. From this scenario we derive three dif-
f erent ways of obfuscation namely: refusal, falsification and ab-
straction. Since we are mostly interested in abstraction we only
briefly mention refusal and falsification.
Alice and Chris know each other and they are both shopping independently. Alice is calling Chris and she is telling that she is in the city in a bookshop. After that she asks where he currently is. At this point Chris has several options and he will, most likely, choose one depending on his mood and relationship with Alice.

Chris might not want to tell Alice that he is also shopping, maybe because he is in a hurry or he is getting a gift for Alice. To achieve this he can, for example, tell her that he doesn’t want to tell her where he is, refusal. He can also tell her that he is at home instead of shopping, falsification. Both refusing and falsification can result in unwanted situations. If Chris refuses to tell Alice what he is doing Alice will suspect Chris from withholding private information and if they accidently meet downtown, while shopping, the falsification results in an awkward situation.

Instead of not telling where he is, Chris can also not tell Alice exactly what he is doing. He can for example tell her that he is shopping and not mention the location or he can say that he is outside. Both these things are true, if he is not in a shop, but Alice will not be able to retrieve exactly what he is doing and where he is. This is called abstraction.

In the following two sections we give a brief description of how refusal and falsification can and should be applied. After that, a description of the abstraction methods which we identified in our research is given. These methods are categorized by data type. Next to the data type based abstraction methods we discuss the k-anonymity method, since quite some research has been done in this method. To conclude we discuss possible vulnerabilities.

4.1 Refusal

As seen in the example, Chris could say to Alice that he does not want to answer a question. In practice people almost never say explicitly that they do not want to answer a question. They typically ignore the question or make sure that the question is not asked, polite blocking. For example, when a user doesn’t want to chat on an instant messenger, the user puts his status on invisible or ‘appear offline’. That way no one can see that the user is online, and thus no one will start a chat. This is in contrast with putting the status on “I don’t want to chat”.

Context aware systems can use the same mechanism. They can ignore the question if the user has stated that he does not want to share the requested information. However if the services shared this type of context in the past, it is strange if the service now refuses.

4.2 Falsification

Another option is to decide not to give the correct answer, falsification. The main problem with falsification is that it must be done in a consistent and reasonable way. If we don’t falsify in a consistent way we might reveal the secrets that we try to hide. We want the receiver to believe our falsification, therefore falsification needs to be done in a reasonable way. There are various techniques which can produce consistent lies. One of these techniques is called polyinstantiation [9]. This technique generates cover stories. By doing this, the technique produces consistent lies. These lies therefore do not reveal the secrets. The problem of reasonable lying still exists.

4.3 Abstraction

In the scenario about Chris and Alice we saw examples of abstraction. This section describes abstraction based obfuscation techniques for various data types. We discuss the techniques which we identified in literature. The data types discussed are: boolean, incremental set, numbers and hierarchical trees. For each data type we examine how the obfuscation techniques work. Furthermore, we examine how the QoC parameters are represented in the data type. After doing this we discuss the string data type, for this data type we did not identify any obfuscation method.

4.3.1 Boolean

It is impossible to obfuscate a boolean by abstraction, since the value is either true or false. For example, the question "is Chris at his work?" can only be answered with yes or no. It is not possible to give another, boolean, answer. There are however two things which can be done. We can state that the PoC is lower than 100%, or we can adjust the context value depending on the desired PoC. This, for example, results in the following calculations. Someone requests if Alice is currently at her work. The service is allowed to answer this question with 70% certainty. Alice is currently shopping, and she doesn’t work in a shop. The service should now, with a chance of 70%, choose false as an answer. True must be selected as an answer with the remaining 30% chance. The included probability of correctness must be 70%. This can be seen as falsification rather than abstraction.

4.3.2 Incremental Sets

Incremental sets can be used for values which consist of more than one part, each part adding more detail. For example, someone describes a location as The Netherlands, Enschede or Matenweg. The first part represents the countries name, the second part a city and the third represents a street. To obfuscate such a set we leave out more detailed data.

For an incremental set the precision parameter is implicit. The precision depends on how many items are returned. The services need to negotiate, in advance, which level of the set corresponds
to which precision level. This is not always obvious. The other parameters are available as meta information to each item in the set and can be obfuscated depending on their data type.

4.3.3 Numbers
This section describes obfuscating techniques which apply to numbers. They have been split up into a section about points with a radius and a section about single numbers.

4.3.3.1 Point with a radius
Ardagna et al. [10] identify three ways of obfuscation a point with a radius namely: obfuscation by enlarging the radius, shifting the center and reducing the radius.

By enlarging the radius the privacy sensitivity of, for example, a location can be reduced. If however the requester knows that the service is simply enlarging the radius, the service can calculate the center of the circle and then know the 'real' location. This can be solved by using the second technique proposed. With this technique the center point is shifted. By doing this, the chance that the users location is somewhere in the circle, is uniformly divided, that is if the new point with a radius includes the users location. This final parameter can be captured with the PoC parameter of the QoC. The third technique proposed is reduction of the radius. This technique results in a smaller chance that the user is actually in the circle, so the QoC decreases. However, if a very precise measurement technique is used the user will stay in the area, even for small radii. For Cell-ID technology this technique might work since Cell-ID's generally have a range varying from 1-3 km in urban areas to 3-20 km in suburban/rural areas [10]. A combination of either increasing or decreasing the radius and shifting the center can also be used to achieve obfuscation.

4.3.3.2 Single numbers
For single numbers, the number of significant figures can be reduced. This results in a lowered QoC, the precision parameter is obfuscated. By reducing the number of significant figures we are creating a grid, in which the real value is positioned somewhere. This technique also is applicable to sets of multiple numbers, for instance a latitude and longitude GPS coordinate combination. By reducing the number of significant figures the precision of the location is reduced. Another application is time, if the time of a day is represented with a number between 0 and 1 the number of significant figures can be lowered to achieve a lower QoC.

4.3.4 Hierarchical tree
In a hierarchical tree, different levels of the tree represent different levels of privacy sensitivity. These levels are typically ordered from less privacy sensitive at the top to more privacy sensitive at the bottom. Wishart et al. [11] identify two kinds of hierarchical trees. The first one exists of valid instances as nodes, see figure 3. This means that when enough leaves are removed, to reach the desired QoC the tree can be send to the requester. The second kind does not have valid instances as nodes, see figure 4. In such a case one must keep in mind never to return more detailed information.

In trees precision is implicit. This parameter is represented in the hierarchy of the tree. Therefore obfuscating this parameter can be done by removing leaves from the tree until the desired precision is achieved. The other QoC parameters are available as meta information and can be obfuscated by selecting the appropriate obfuscation method depending on the data type representing the QoC parameter.

4.3.5 String
In this section we discuss the QoC parameters regarding the string data type. Obfuscating in the precision dimension can, for example, be achieved by removing characters. This however, does not necessarily result in a lower precision. When only a few characters are removed the receiver typically is able to reconstruct the original string. Reconstructing, the original string, can only be done when a limited number of characters are obfuscated.

For obfuscation of a string in the precision dimension it might be a good idea to translate the meaning, of the string, into a hierarchical tree. This tree can be used to derive more abstract representations of the original string. The other QoC parameters are available as meta information and can be obfuscated by selecting the appropriate obfuscation method depending on the data type representing the QoC parameter.
4.4 K-anonymity
K-anonymity is an obfuscation technique used in databases with different people to protect the identity of a single person. The K in K-anonymity stands for the number of persons which must stay indistinguishable from each other when information is requested. The information selecting criteria is called a quasi identifier. How it works is best shown by example. The example is based on Heerde’s thesis [2].

<table>
<thead>
<tr>
<th>Year of Birth</th>
<th>Gender</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>man</td>
<td>Running</td>
</tr>
<tr>
<td>1975</td>
<td>man</td>
<td>Tennis</td>
</tr>
<tr>
<td>1984</td>
<td>woman</td>
<td>Basketball</td>
</tr>
<tr>
<td>1984</td>
<td>woman</td>
<td>Tabletennis</td>
</tr>
<tr>
<td>1983</td>
<td>man</td>
<td>Basketball</td>
</tr>
<tr>
<td>1983</td>
<td>man</td>
<td>Soccer</td>
</tr>
</tbody>
</table>

Table 2: 2-Anonymity example

Table 2 shows a list of year of births and genders with activities. In this example a combination of year of birth and gender is the quasi identifier. When selecting with this quasi identifier we want it to resolve to at least two different activities. With this restriction we can never discover exactly what activity someone is doing. Each selection of year of birth and gender, in the example, results in a minimal number of two activities, so this is a 2-anonymity example.

4.5 Vulnerabilities
The number techniques discussed above suffer from a major drawback. If the information requestor has multiple samples of information, there is a chance that he or she can deduce more information than allowed. For example, a person has a lot of GPS coordinate samples with some radius, within a reasonable time window. That person can take the intersection of those points and deduce a smaller area of where someone is probably located, see figure 5.

With a grid the same problem arises. Consider the situation that someone is on the edge of a grid cell and a service is requesting multiple samples. Since the person is on the edge alternating grid cells can be returned. These alternating grid cells reveal a location line. This most likely occurs because of measurement faults when someone is on the edge, because than the grid can alternate easily between two cells. When the grid cells are relative big to the measurement error the chance of this occurring is relative low. The other possibility of an alternating grid cell is when someone is actually moving.

This problem can be solved by keeping a history of what information has been provided. For example, as long as a person stays within a certain range of the point which is send the first time the same information will be send again. Only after a certain amount of time or ‘major’ location changes the grid or circle must be adjusted. This introduces the problem that services need to keep track of what context has been sent to who and when.

5. ANALYSIS OF QOC PARAMETERS AND CATEGORIES OF CONTEXT
In this section we join the categories of context with the five QoC parameters. We list each context category and reason which QoC parameters are good candidates for obfuscation. The criterion for a parameter to be a good candidate is: when the context is obfuscated in that dimension it typically results in less privacy sensitive context.

For context from other categories than identity we assume the identity is known. Otherwise the information cannot be privacy sensitive. This follows from our definition of privacy. From this we can conclude that identity is the most important category regarding privacy. This results in a hierarchy within the context categories, see figure 6.

After describing which QoC parameters are good obfuscation candidates for a ‘single’ category request we reason about how to combine various categories.

5.1 Identity
If identity is known with high precision and probability, the information typically is very privacy sensitive. This is, because denying becomes impossible and information about individuals typically is more privacy sensitive than information about groups.

There is a relationship between historical information about someone and the legitimacy of this information. For example, a user has been using an email address for several years, it is safe to assume the user still uses the email address. Thus the QoC parameters regarding time are also important for identity. The precision
and PoC parameters however are more important since these imply plausible deniability. The results are summarized in table 3.

In general it is very hard to obfuscate a person’s identity. In many situations it can be derived from other context. For example, if a location is known very precise and the information is new, someone could travel to the location and see who are there. To counter this, the PoC must be obfuscated. If the PoC is not 100% the person involved can always deny being there or doing something at that specific location. If someone travels to the location it could very well be wrong, resulting in the person not being there. If there is an entity that is all knowing, the k-anonymity technique, section 2, can be used to protect someone’s identity.

<table>
<thead>
<tr>
<th>QoC</th>
<th>Imp</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>++</td>
<td>Information about a specific person is more privacy sensitive, for that person, than information about groups</td>
</tr>
<tr>
<td>Fr</td>
<td>+</td>
<td>Old information can be changed</td>
</tr>
<tr>
<td>TR</td>
<td>+</td>
<td>Information which was applicable for a long time probably still is</td>
</tr>
<tr>
<td>SR</td>
<td>- -</td>
<td>When requesting identity information the location does not matter. If location would have been important, that category would also be requested</td>
</tr>
<tr>
<td>PoC</td>
<td>++</td>
<td>A less than 100% PoC provides the user with plausible deniability</td>
</tr>
</tbody>
</table>

### Table 4: Location

<table>
<thead>
<tr>
<th>QoC</th>
<th>Imp</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>++</td>
<td>If decreased less information is given, thus privacy sensitivity decreases</td>
</tr>
<tr>
<td>Fr</td>
<td>-</td>
<td>Location information has nothing to do with time so this parameter is not important</td>
</tr>
<tr>
<td>TR</td>
<td>-</td>
<td>Location information has nothing to do with time so this parameter is not important</td>
</tr>
<tr>
<td>SR</td>
<td>++</td>
<td>If decreased less information is provided, this parameter is closely related to the precision parameter</td>
</tr>
<tr>
<td>PoC</td>
<td>++</td>
<td>A less than 100% PoC provides the user with plausible deniability</td>
</tr>
</tbody>
</table>

### Table 5: Time

<table>
<thead>
<tr>
<th>QoC</th>
<th>Imp</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>++</td>
<td>If decreased less information is given, thus privacy sensitivity decreases</td>
</tr>
<tr>
<td>Fr</td>
<td>+</td>
<td>If information is older it is, more often than not, less privacy sensitive</td>
</tr>
<tr>
<td>TR</td>
<td>++</td>
<td>A higher resolution typically reveals more information and thus is more privacy sensitive</td>
</tr>
<tr>
<td>SR</td>
<td>-</td>
<td>Does not influence time information and therefore not a good obfuscation candidate</td>
</tr>
<tr>
<td>PoC</td>
<td>++</td>
<td>A less than 100% PoC provides the user with plausible deniability</td>
</tr>
</tbody>
</table>

### Table 6: Activity

<table>
<thead>
<tr>
<th>QoC</th>
<th>Imp</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr</td>
<td>++</td>
<td>If the activity is more abstract it reveals less information about the user and thus is less privacy sensitive</td>
</tr>
<tr>
<td>Fr</td>
<td>+</td>
<td>If the measurement is from a few days ago the QoC typically is lower than if the information was measured only a moment ago</td>
</tr>
<tr>
<td>TR</td>
<td>+</td>
<td>If the context was applicable a moment ago, there is a good chance that the activity is still going on</td>
</tr>
<tr>
<td>SR</td>
<td>-</td>
<td>When requesting only activity information the location does not matter, if location would have been important information from that category would also be requested</td>
</tr>
<tr>
<td>PoC</td>
<td>++</td>
<td>A less than 100% PoC provides the user with plausible deniability</td>
</tr>
</tbody>
</table>

### 5.2 Location

For location information three parameters are good candidates for obfuscation namely: precision, spatial resolution and PoC. A relationship exists between spatial resolution and precision, because if the spatial resolution is lower the precision is also typically lower, and vice versa. These parameters are important for privacy sensitivity since if the location is known more precise, more facts can be derived from it. The time parameters are not important since if time is important the request would also be from the time category. The results are summarized in table 4.

### 5.3 Time

For time related context three parameters are good candidates for obfuscation namely: precision, temporal resolution and PoC. A relation exists between precision and temporal resolution for time concerned context. This is because if the temporal resolution is higher the precision is typically also higher and vice versa.

Freshness can be a good obfuscation candidate, users typically have the opinion that new information is more privacy sensitive compared to old data. This is, because the information has more potential to be used for purposes the user didn’t intend. For example, the context information can be used for advertising. It is however not always a good obfuscation candidate since sometimes the requested information remains privacy sensitive, even if it is old. The results are summarized in table 5.
5.4 Activity
For activity, besides precision and PoC, freshness and temporal resolution are good obfuscation candidates. Freshness is a good candidate, because if the activity was going on a moment ago it is likely it still is. Temporal resolution is a good candidate, because if the context was applicable a few moments ago, it is likely that the activity is still going on. The results are summarized in table 6.

5.5 Multiple context category requests
We saw which QoC parameters are good candidates for obfuscating one specific category of context. In practice however, context is often requested from more than one category. To determine which QoC parameters are good obfuscation candidates we combine the different tables, each time picking the highest value. For example, we combine a request about identity and location, by also requesting location the spatial resolution parameter also gets a good candidate for obfuscation.

6. HOW TO OBFUSCATE CONTEXT
Obfuscation can be done in different QoC dimensions. In the previous section we saw which QoC parameters are good obfuscation candidates for each context category. In this section we outline how obfuscating context in different QoC dimensions can be done.

6.1 Precision
Which methods are applicable for obfuscation in the precision dimension, highly depends on the data type. Methods for obfuscating context of different data types have been mentioned in section 4.

6.2 Freshness
The freshness parameter can be obfuscated in three different ways. The first one is by using an older sample. This results in older information and thus obfuscation. The second way is to just state that the information has a different freshness than known. This last method however only works if the requesting service cannot retrieve that it is just stated that the information is older. Finally if the freshness is represented as a number, obfuscation methods identified in section 4 can be applied.

6.3 Temporal resolution
The main idea to obfuscate the temporal resolution is, to make the time window to which the context applies bigger. By doing this one cannot retrieve the specific moment the context occurred. This can be done by combining sensor information, to either make an average of a time window or count them together. For example, the average temperature of a building during a week need to be averaged. If the TR is represented as a number significant figures can be removed to achieve obfuscation.

6.4 Spatial resolution
Spatial resolution can be lowered by combining multiple sensor information. By combining this information the precise knowledge is gone and thus the information becomes less privacy sensitive. For example, when the number of people in a building is requested and the service knows how many people there are in every room. The service can count all the different rooms in the building and return the total, with the a spatial resolution of the building. This in contrast with returning the number of people in all rooms separately.

6.5 Probability of Correctness
The PoC can be altered in two ways. The first and simplest way is by just stating that the probability is lower than known. The other way is by selecting ‘wrong’ data with a certain chance. By doing this the PoC decreases and thus the privacy sensitivity decreases.

6.6 Relation between PoC and the other parameters
By obfuscating a QoC parameter other than PoC, typically the PoC parameter increases. When this obfuscation is done, ranges typically get bigger. This results in an increased chance that the real value lays within the range and in that case, a higher PoC is obtained.

7. USE CASE: CARCOACH
CarCOACH [12] is a system that can be used to monitor the driving skills of a driver. The system provides feedback to the driver about how he or she is performing. The driver can use these comments to improve his driving skills. Driving errors that are identified are for example: turning without indicating direction, erratic steering, accelerating and breaking abruptly.

In our case, we consider that the system logs all these events. These logged events can be dispersed to several organizations. The government can, for example, use the information to decide if a public campaign for specific driving errors is needed, or not. The leasing company can use the gathered information to make a bill or identify car abuse.

We use the following data: table 7 for the error data, table 8 for sensor data.

<table>
<thead>
<tr>
<th>id/Attribute</th>
<th>Time</th>
<th>Error</th>
<th>Steer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16:32:23</td>
<td>No blinking</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>16:40:24</td>
<td>Abrupt breaking</td>
<td>-45</td>
</tr>
<tr>
<td>3</td>
<td>16:45:25</td>
<td>No blinking</td>
<td>40</td>
</tr>
</tbody>
</table>

We first describe some requests that can be made, after that we evaluate how the dimension selection and obfuscation has worked.

7.1 Context requests
7.1.1 Speeding
The government is interested in where people, with a carcoach, exceeded 120 km/h. This request consists of two categories of context, location and activity. Therefore the following QoC parameters are good obfuscation dimensions: Pr, SR and PoC and in lesser degree Fr and TR. By obfuscating in the Pr dimension the SR also decreases and exact retrieval of the location is not possible anymore. By selecting the first row from table 8 we, for example, obfuscate the GPS coordinate to 52.03. Furthermore, obfuscation in the PoC dimension provides the user with plausible deniability. This can be done by stating a lower PoC.
Table 8: Sensor data

<table>
<thead>
<tr>
<th>↓ id/Attribute →</th>
<th>Location</th>
<th>Time</th>
<th>Speed</th>
<th>RPM</th>
<th>Gear</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52.03492426269531, 6.34299325942993</td>
<td>16 : 32 : 23</td>
<td>123</td>
<td>2400</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>52.0351486206055, 6.34258508682251</td>
<td>16 : 32 : 24</td>
<td>125</td>
<td>2440</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>52.0352554321289, 6.34237670898438</td>
<td>17 : 35 : 01</td>
<td>126</td>
<td>1500</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>52.0351486206055, 6.34258508682251</td>
<td>17 : 35 : 02</td>
<td>125</td>
<td>2440</td>
<td>3</td>
</tr>
</tbody>
</table>

7.1.2 Blinking
The government wants to know how many times, in the last week, cars did not use their blinker while changing direction. This context request is from the time and activity category. Therefore the Pr, PoC, TR and, in lesser degree, the Fr are good obfuscation dimensions. The TR is in seconds but only a resolution of a week is needed. Therefore the system must combine the sensor information until a TR of a week is reached. In this case there have been 2 cases of forgetting to blink, in the last week. After that, obfuscation in the Pr and PoC dimension can be applied, both further reduce the QoC and thus the privacy sensitivity.

7.1.3 Leasing billing
To determine the bill a leasing company wants to know how far each of their cars have been driving. This request is from the context categories identity and location, because the distance can be determined from where the car has been. The identity cannot be obfuscated, since than the leasing company cannot send the bill to the correct person. To protect the privacy the leasing company does not need to know the exact location where the cars have been, only the total distance traveled. For the location category we see that the SR together with the Pr and PoC are the best obfuscation dimensions. Obfuscation in the PoC dimension is impossible since the leasing company needs to know for sure how far the car has traveled. Obfuscation in Pr dimension is probably also limited since the leasing company uses, for example, billing for each km traveled. The SR therefore is the best dimension. The application must gather all the location information needed to calculate the distance and send the total distance to the leasing company.

7.2 Evaluation
In all cases obfuscation in the selected dimensions resulted in less privacy sensitive data. The main remark that has to be made is that if we would have adopted the minimal disclosure principle, this would have resulted in almost the same obfuscation. The difference with the minimal disclosure is that we add obfuscation in the PoC dimension to provide users with plausible deniability. Furthermore, we saw that it is not always possible to obfuscate in a certain dimension because the information than becomes useless to the requester.

8. RELATED WORK
Lederer et al. [13] propose to use four levels of abstraction for context. These levels are: precise, approximate, vague and undisclosed. This set is limited by its four levels. In practice it is not always possible to give enough options. To quote an example from Wishart et al. [11]: "For example, assume exact location, the building the user is currently in, the suburb, and the city in which the user is located are provided as the four levels of obfuscation. A user not wanting to provide her exact location, but wanting to give a more specific value for her location than the building, such as a floor number, cannot do so.”

Wishart et al. [11] propose to obfuscate context using hierarchical trees. They provide good reasoning about trees regarding obfuscation. The main issue is that not all information can be represented in a tree.

Anciaux et al. [14] propose to use a timely degradation of precision as time passes by. This reflects the mechanism of people forgetting details. We have not used this technique because of two reasons. First, the technique is aimed at databases with context info about many people and we are not interested in these databases. Furthermore this technique can be simulated by reducing the freshness and the precision.

Sheikh et al. [5] propose a QoC concept. In this paper we have used this concept. We extended their work by investigating the relation between QoC parameters and Dey et al. [6] context categories.

9. CONCLUSION
To protect a user’s privacy, obfuscation of context is needed. Initially we have researched the Quality of Context (QoC) concept. After that we investigated several obfuscation methods. The third research was aimed at categorizing context. We categorized context information into Dey’s categories, namely: Identity, Location, Time and Activity. We concluded that the identity category is needed for data to be privacy sensitive.

Table 9: Summary of QoC and Context Categories

<table>
<thead>
<tr>
<th>↓ CC/QoC →</th>
<th>Pr</th>
<th>Fr</th>
<th>TR</th>
<th>SR</th>
<th>PoC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>++</td>
</tr>
<tr>
<td>Location</td>
<td>++</td>
<td>-</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Time</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>++</td>
</tr>
<tr>
<td>Activity</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
</tbody>
</table>

We used the performed research to combine the categories of context with the QoC concept. A summary of this research is given in table 9. From this table we conclude that probability of correct-
ness and precision are always good obfuscation candidates. If the category of a request is determined, one can determine which QoC candidates are good obfuscation dimensions.

When the obfuscation candidates are determined, the context can be obfuscated in one or more of these dimensions. Freshness can be obfuscated by using old information. Context can be obfuscated in the temporal resolution or spatial resolution by combining information from different sensors. The probability of correctness can be obfuscated by selecting false data with a certain chance or by stating that the correctness is lower. Obfuscating in the precision dimension is done by using one of the abstraction methods. These abstraction methods can also be used to obfuscate the other dimensions if they are represented in a data type for which an obfuscation method is available.

10. FUTURE WORK
Future work can be a user study on how users would like to enter their privacy preference. A possible hypothesis is that users prefer to enter location information for a tree based model.

Further research can be done into how to obfuscate other data types e.g. strings, unordered sets.

It is a good idea to perform research on the connection between, how the relation is between people and how much information they want to share. If a user, for example, is requesting information about another user, one might recognize a pattern in how much information someone is willing to reveal. A hypothesis is that people are willing to share information one level of a hierarchical tree below their relationship level. For example, a random colleague from work is wondering what you are doing. If that person is at work, he can only see what project you are currently working on. If some friend, who is not related to your work, is asking what you are doing, he is only allowed see that you are currently working. Furthermore, if some colleague from the same project is wondering what you are doing, that colleague is allowed to know exactly what you are doing at work. This will most likely be in contrast with when you are sporting, than he is only allowed to see that you are relaxing.

ACKNOWLEDGEMENTS
I would like to thank everyone who played a significant role in making this research possible, providing useful information and/or reviewing my work. Especially I would like to thank Maarten Wegdam, Tom Broens and Marten van Sinderen my supervisors in this research.

REFERENCES
APPENDIX A: CONTEXT CATEGORIES

Petrelli et al. [15] developed audio tours for museums. While developing these applications they recognized the following two categories of context: material context and social context. Material context includes the place of use, the device and the available infrastructure. The social context includes all the social aspects, like if the user is alone and if not, who are the people in his neighborhood, what is the attitude towards those people etc. According to Petrelli et al., both aspects are equally important in determining how the service should adapt to the user’s behavior.

Gwizdka et al. [16] developed an almost similar concept to categorize context. Gwizdka makes a distinction based on whether the context is internal or external to the user. Examples of internal context are: Current work project, project team, emotional state, personal events. These personal events are internalized external events. When context is about the environment of the user it is revered to as external context, for example location, distance to other objects, the outside weather.

Schmidt et al. [17] divides context into two primary categories, human factors and physical environment. Each of these categories consists of three subcategories.

Human Factors:

- User
- Social environment
- Task

Physical environment:

- Physical condition,
- Infrastructure
- Location

When taking a close look at these categories and comparing them with Dey’s categories we see that User and Social Environment refer to identity. Task refers to activity and Physical conditions, Infrastructure and Location are combined by Dey’s to the context category Location. The time aspect is omitted in this categorization, but this is an important aspect.

Razzaque et al. [18] propose two broad viewpoints to categorize context, conceptual viewpoint and the measurement viewpoint. The conceptual point of view uses questions like who, what, where to categorize context. A complete list of categories with examples can be found in table 10. When comparing table 1 with table 10 we see that Dey and Razzaque are categorizing context in similar ways.

<table>
<thead>
<tr>
<th>Table 10: Conceptual Categorization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
</tr>
<tr>
<td>User context</td>
</tr>
<tr>
<td>Physical context</td>
</tr>
<tr>
<td>Network context</td>
</tr>
<tr>
<td>Activity context</td>
</tr>
<tr>
<td>Device context</td>
</tr>
<tr>
<td>Service context</td>
</tr>
</tbody>
</table>

The measurement categorization viewpoint takes a look at the data types which can be used. These categories are: Continuous context, Enumerative context, State context, Descriptive context. Since the quality of context concept we uses very different criteria to measure the QoC this categorization is not suitable for our research.