

Requirement analysis : *TraffiX*

Designing a software architecture

Joris Schelfaut (DistrICT Education)

May 21, 2014

Contents

1	Domain description	2
1.1	Domain model	2
1.2	Glossary	2
2	Functional requirements	4
2.1	UC1 : Log in	4
2.2	UC2 : Log out	5
2.3	UC3 : Create account	6
2.4	UC4 : Manage account	7
2.5	UC5 : Consult parking advice	7
2.6	UC6 : Consult travelling advice	8
2.7	UC7 : Pay parking meter	9
2.8	UC8 : Manage parking meters	9
2.9	UC9 : Perform research	10
2.10	UC10 : Install meter	11
2.11	UC11 : Evaluate system state	11
2.12	UC12 : Send parking data	12
2.13	UC13 : Send traffic flow data	12
3	Non-functional requirements	13
3.1	Availability	13
3.1.1	A1 : Parking data transmission failure	13
3.1.2	A2 : Internal traffic flow database failure	13
3.2	Performance	14
3.2.1	P1 : Travelling advice computation	14
3.2.2	P2 : Requests to the traffic flow database	14
3.3	Security	14
3.3.1	S1 : Client database safety	14
3.4	Usability	15
3.4.1	U1 : Client user interface	15

1 Domain description

1.1 Domain model

Figure 1 shows the domain model diagram.

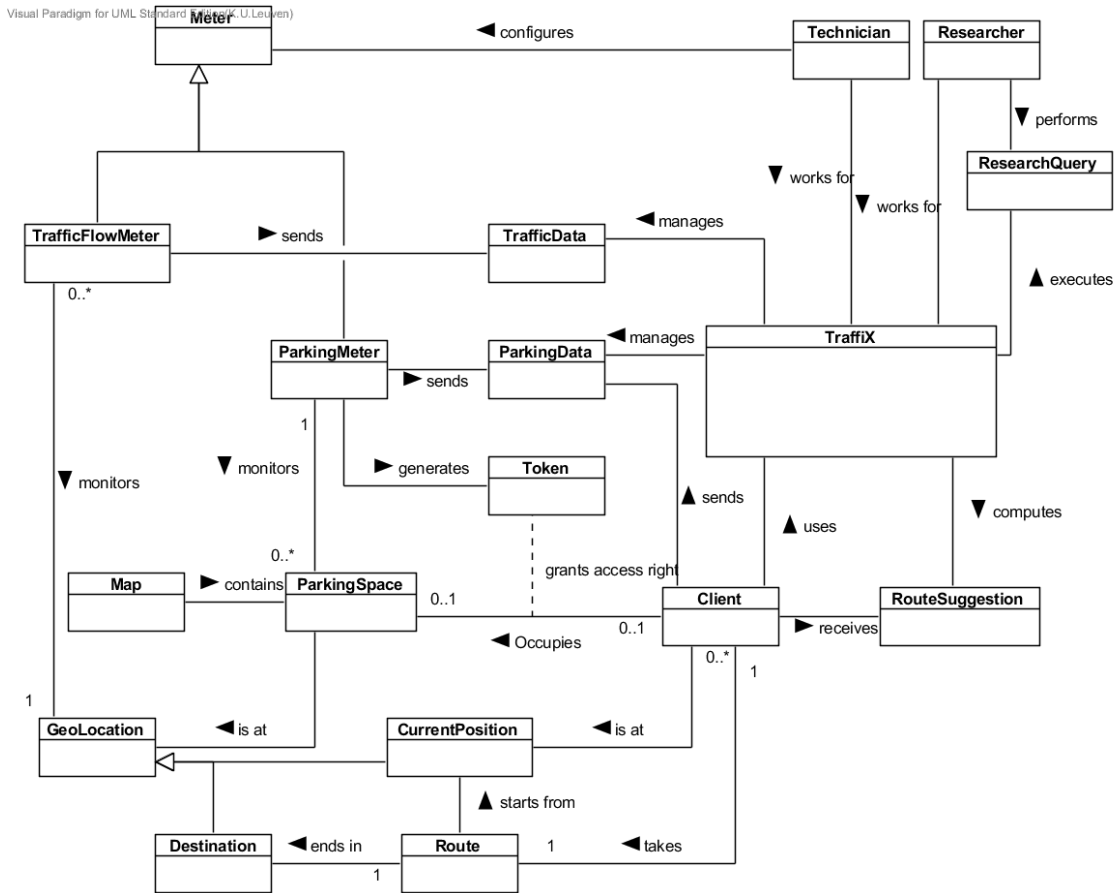


Figure 1: Domain model diagram for the *TraffiX* system.

1.2 Glossary

- **Client** : A client user of the Traffix system.
- **CurrentPosition** : The current position (GeoLocation) of a client.
- **Destination** : The destination of a route a client is taking.
- **GeoLocation** : The coordinates of a person or object on a Map.
- **Map** : An abstract, two-dimensional representation of the physical world using a coordinate system to uniquely identify each location.
- **Meter** : A device for monitoring a certain state and send updates on this state to the Traffix system.
- **ParkingData** : The data send from a parking meter or a client about the state of a parking space.

- **ParkingMeter** : A meter that monitors the state of a set of parking spaces and also responsible for creating a Token for a Client wishing to park at a certain parking space.
- **ParkingSpace** : A space, corresponding to a GeoLocation, where a client can park his/her car.
- **Researcher** : A person who can perform research queries on anonymous TrafficX data.
- **ResearchQuery** : A specific query on anonymous TrafficX data.
- **Route** : The path on a Map that a Client will take to get from his/her CurrentPosition to his/her Destination.
- **RouteSuggestion** : A suggestion on which Route best to follow to get to the Destination most efficiently according to the TrafficX system.
- **TrafficFlowMeter** : A Meter monitoring the traffic flow state at a certain GeoLocation.
- **TrafficData** : Piece of information sent from a TrafficFlowMeter containing traffic flow data.
- **Technician** : A person who can install and configure meters.
- **Token** : A certificate a client receives after paying for the right to use a parking space.
- **TrafficX** : An abstraction of the system's data and computational capabilities.

2 Functional requirements

Figures 2 and 3 show the use case diagrams for the *TraffiX* system.

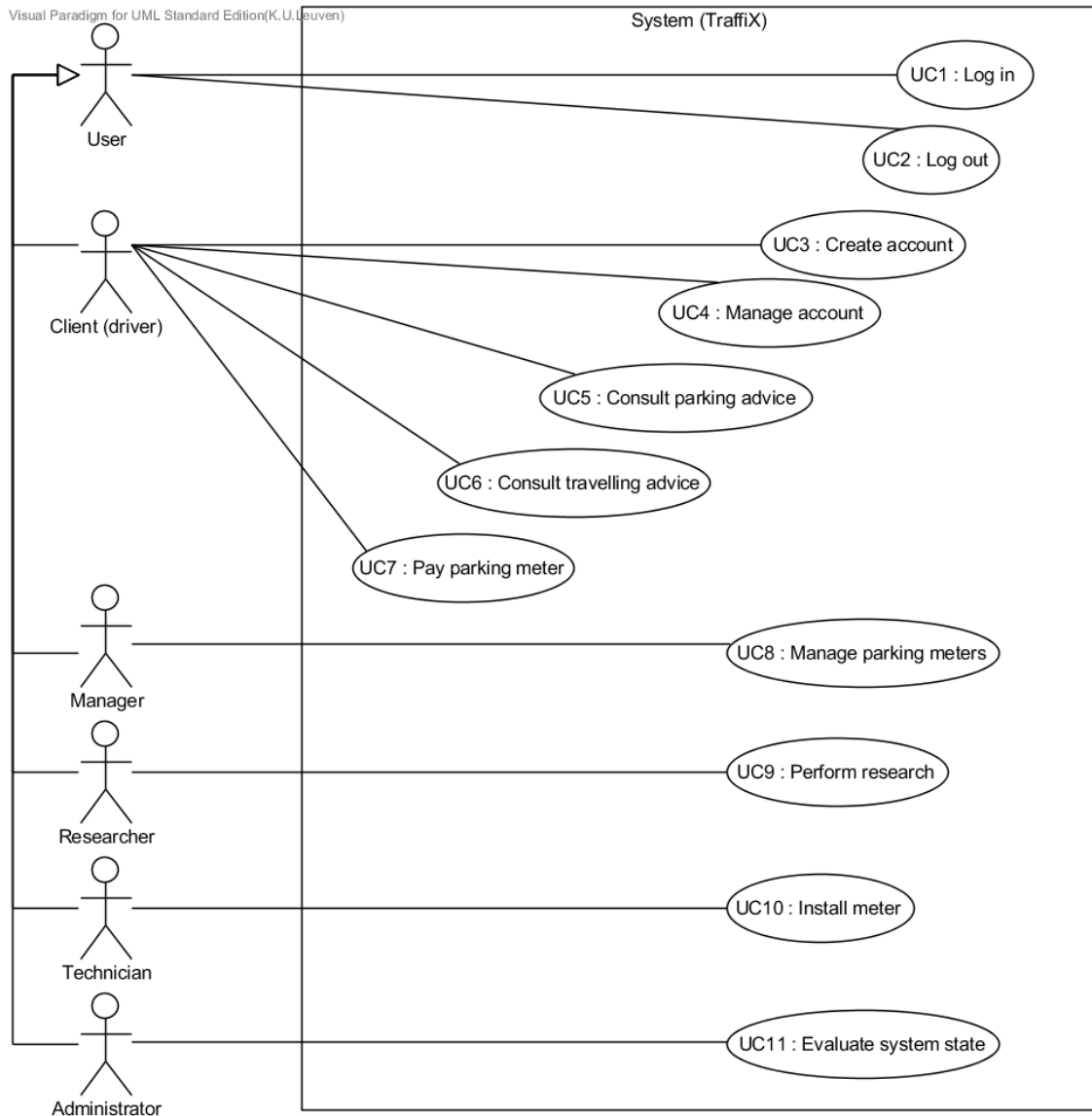


Figure 2: Use case diagram for the stakeholders in the *TraffiX* system.

2.1 UC1 : Log in

- **Name** : Log in.
- **Primary actor** : User.
- **Interested parties** :
 - TraffiX : want to authenticate its users for access control.
 - User : wishes to use TraffiX services.

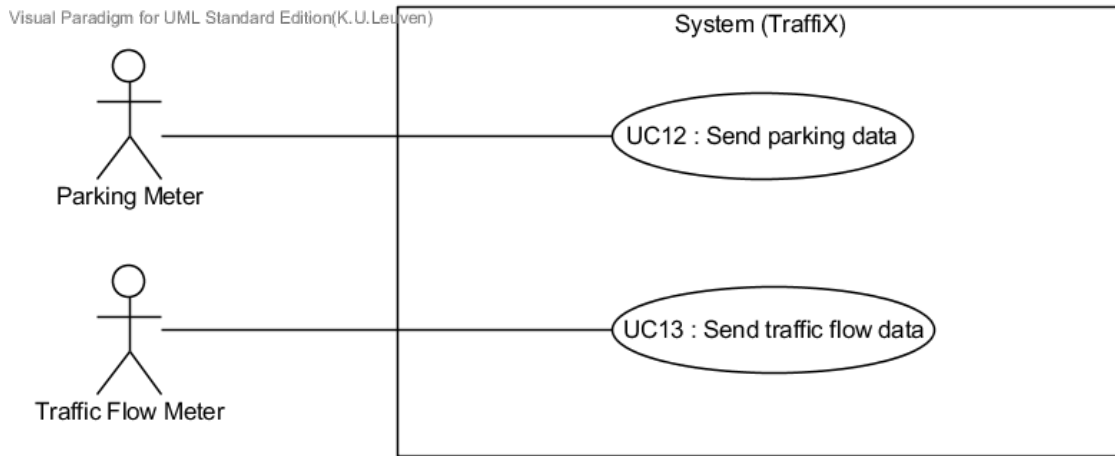


Figure 3: Use case diagram for peripheral devices in the *TraffiX* system.

- **Preconditions :**

- The User is registered in the system.

- **Post-conditions :**

- The User has authenticated him/herself in the Traffix system.

- **Main scenario :**

1. The User indicates he/she wants to authenticate into the Traffix system.
2. Traffix asks him/her to provide his/her credentials.
3. The User provides his/her credentials.
4. Traffix verifies the provided credentials and authenticates the User.

- **Alternative scenarios :**

- 4b. The provided credentials were incorrect, resume at step 2.

2.2 UC2 : Log out

- **Name :** Log out.

- **Primary actor :** User.

- **Interested parties :**

- Traffix : want to authenticate its users for access control.
- User : For the moment no longer needs the Traffix services and wishes to end the current session.

- **Preconditions :**

- The User is currently authenticated into the Traffix system.

- **Post-conditions :**

- The User is no longer authenticated into the Traffix system.

- **Main scenario :**

1. The User indicated he/she wants to log out of the TraffiX system.
2. TraffiX logs the user off.

2.3 UC3 : Create account

- **Name :** Create account.

- **Primary actor :** Client.

- **Interested parties :**

- Client : wishes to use TraffiX services.
- TraffiX : wants to keep track of its users.

- **Trigger :** An external person wants to be able the use TraffiX services, e.g., to find parking spots.

- **Preconditions :**

- The Client's device is not yet registered in the system.

- **Post-conditions :**

- The Client is now registered in the system. His/her device's GPS can now be used to gather traffic data and payment metadata can be used for gathering parking data. The Client can now use TraffiX services.

- **Main scenario :**

1. A Client indicates he/she wants to register him/herself into the TraffiX system.
2. TraffiX shows a registration form. The Client is asked to provide the following information:
 - Login credentials (user name and password);
 - Contact data including email address and phone number;
 - Hometown and postal code;
 - ...
3. The Client enters the requested information.
4. TraffiX registers the new Client and displays a message that the account was successfully created.
5. The User confirms his/her account via SMS or email.

- **Alternative scenarios :**

- 4b. The registration has failed. The system indicates what went wrong; in case that incorrect information was provided, the system indicates this on the form after returning to step 2 (while keeping the previously entered information filled in).
- 5b. If after 10 days the account was not confirmed, the account is deleted automatically.

2.4 UC4 : Manage account

- **Name** : Manage account.
- **Primary actor** : Client.
- **Interested parties** :
 - Client : wants to update his/her account data and/or change certain settings.
 - Traffix : wants to have the most accurate information to provide the best services.
- **Trigger** : Some data in the user profile is not up-to-date or some of the current settings aren't satisfactory to the Client.
- **Preconditions** :
 - The Client has an account in the Traffix system.
 - The Client is authenticated.
- **Post-conditions** :
 - The account data and settings are updated correctly.
- **Main scenario** :
 1. A Client indicates he/she wants to update account data or alter settings.
 2. Traffix shows a form where account data can be edited.
 3. The Client changes updates the relevant fields.
 4. Traffix registers the changes and notifies the Client the changes were successfully stored.
- **Alternative scenarios** :
 - 4b. Traffix detected that incorrect information was provided. The system indicates this on the form after returning to step 2 (while keeping the previously entered information filled in).

2.5 UC5 : Consult parking advice

- **Name** : Consult parking advice
- **Primary actor** : Client
- **Interested parties** :
 - Client : wants to get efficiently to a parking space.
 - Traffix : wants to help the Client finding a parking spot.
 - Manager : wants Clients to use his/her parking lot(s).
- **Trigger** : the Client needs to park his/her car somewhere in the city.
- **Preconditions** :
 - The Client has a Traffix account.
 - The Client is logged in.
 - The Client has a destination and current route to take.
- **Post-conditions** :

- The Client has received parking advice for his/her chosen destination.
- The chosen suggestion is added to the route.
- **Main scenario :**
 1. The Client indicates he/she wants to receive parking advice for a chosen destination.
 2. TraffiX computes suggestions and shows them on the screen.
 3. The Client selects a chosen suggestion.
 4. TraffiX adds the data to the route plan.

2.6 UC6 : Consult travelling advice

- **Name :** Consult travelling advice
- **Primary actor :** Client
- **Interested parties :**
 - Client : wants to get advice on how to adjust the current path in the presence of potential traffic congestion.
 - TraffiX : wants to help the client find the best route to take.
- **Trigger :**
- **Preconditions :**
 - The Client has a TraffiX account.
 - The Client is logged in.
 - The Client has a destination and current route to take.
- **Post-conditions :**
 - The travelling advice is added to the current route.
- **Main scenario :**
 1. The Client indicates he/she wants to receive travelling advice for a chosen destination and current route.
 2. TraffiX computes suggestions based on the current traffic situation in the city and presents the results.
 3. The Client selects a travelling suggestion.
 4. TraffiX updates the current route accordingly.
- **Alternative scenarios :**
 - 3b. The Client declines the suggestions. The current route is not modified.

2.7 UC7 : Pay parking meter

- **Name** : Pay parking meter
- **Primary actor** : Client
- **Interested parties** :
 - Client : wants to receive a token that he/she has the right to park in the parking spot his/her car is currently at for a given amount of time.
 - Manager : wants to receive payments for the provided parking service.
 - TraffiX : wants to update its parking occupancy data.
- **Trigger** : The client wants to get a certificate that he/she has the right to park in the parking spot his/her car is currently at for a given amount of time.
- **Preconditions** :
 - /
- **Post-conditions** :
 - The Client has received a token that he/she has the right to park in the parking spot his/her car is currently at for a given amount of time.
 - TraffiX has registered the parking info and updated its parking data.
 - The payment has been registered.
- **Main scenario** :
 1. The Client indicates he/she wants to pay for using a parking space.
 2. The Client enters the duration, the car's licence plate and the parking space code into TraffiX.
 3. The Client confirms his/her parking data.
 4. TraffiX computes the amount the Client has to pay.
 5. The Client choses a payment method and pays for using the parking space and receives a token.
 6. TraffiX updates its parking data.
- **Alternative scenarios** :
 - 5b. The Client has insufficient credit to complete the payment. The client can chose from other payment methods or cancel the purchase.

2.8 UC8 : Manage parking meters

- **Name** : Manage parking meters
- **Primary actor** : Manager
- **Secondary actor** :
- **Interested parties** :
 - Manager : wants manage the parking meters he/she owns or is responsible for.
 - TraffiX : wants to keep its parking meter data up-to-date.
- **Trigger** :

- **Preconditions :**
 - The Manager is logged in into TraffiX.
- **Post-conditions :**
 - The parking meter data is updated correctly.
- **Main scenario :**
 1. The Manager indicates he/she wants to update data on his/her parking meters and related parking spots.
 2. TraffiX shows a list of the current parking meter data.
 3. The Manager selects an action to perform (create, update, delete) on the parking meter data.
 4. TraffiX presents the correct form.
 5. The Manager enters the new or updated data.
 6. TraffiX saves the data and updates its parking data accordingly.
 7. The Manager receives a confirmation message that the data was updated correctly.
- **Alternative scenarios :**
 - 6b. The data could not be saved because of inconsistencies. The Manager is shown an error message.

2.9 UC9 : Perform research

- **Name :** Perform research
- **Primary actor :** Researcher
- **Interested parties :**
 - Researcher : wants to obtain valuable insights in the TraffiX data.
 - TraffiX : wants to provide accurate models of the data.
- **Trigger :**
- **Preconditions :**
 - The Researcher is authenticated into the TraffiX system.
- **Post-conditions :**
 - The research query result is shown to the Researcher.
- **Main scenario :**
 1. The Researcher indicates he/she wants to perform a research query.
 2. TraffiX shows a form in which query parameters can be selected/added.
 3. The Researcher enters the query into the TraffiX system.
 4. TraffiX computes the query result.
 5. The Researcher selects a visualization of the query and/or selects additional algorithms to perform on the data.
 6. TraffiX shows the resulting visualization.

2.10 UC10 : Install meter

- **Name** : Install meter
- **Primary actor** : Technician
- **Interested parties** :
 - Technician : wants to install the meter.
 - Manager : wants a meter for the parking spaces he/she owns.
 - TraffiX : wants to register a new meter to retrieve parking info or traffic flow data.
 - Client : wants to be able to receive a token that he/she has the right to park in the parking spot his/her car is currently at for a given amount of time.
- **Trigger** : a new meter has to be installed for a set of parking spaces.
- **Preconditions** :
 - The Technician is authenticated into the TraffiX system.
- **Post-conditions** :
 - A new meter is installed.
- **Main scenario** :
 1. The Technician indicates he/she wants to install a new meter.
 2. The Technician configures the meter and connects it to TraffiX.

2.11 UC11 : Evaluate system state

- **Name** : Evaluate system state
- **Primary actor** : Administrator
- **Interested parties** :
 - Administrator : wants to make sure the system remains available, secure and consistent.
- **Trigger** : Response to a system failure or general checkup.
- **Preconditions** :
 - The Administrator is authenticated into the TraffiX system.
- **Post-conditions** :
 - An evaluation is completed successfully.
- **Main scenario** :
 1. The Administrator indicates he/she wants to evaluate the TarffiX system state.
 2. TraffiX gives an overview of the most important components of the TarffiX system and lists notification messages.
 3. The Administrator handles failures and updates.

2.12 UC12 : Send parking data

- **Name** : Send parking data
- **Primary actor** : Parking Meter
- **Interested parties** :
 - Parking meter: wants to send parking data to TraffiX.
 - TraffiX : wants to collect parking data.
- **Trigger** : new parking data has become available for sending.
- **Preconditions** :
 - The parking meter is correctly installed.
- **Post-conditions** :
 - The parking data is successfully sent and registered in the TraffiX system.
- **Main scenario** :
 1. The parking meter sends the parking data to the TraffiX system.
 2. TraffiX confirms receiving the data.
 3. TraffiX successfully registers the parking data.
- **Alternative scenarios** :
 - 2b. The message containing the data is not confirmed and the parking meter tries sending it again.

2.13 UC13 : Send traffic flow data

- **Name** : Send traffic flow data
- **Primary actor** : Traffic Flow Meter
- **Interested parties** :
 - Traffic flow meter: wants to send parking data to TraffiX.
 - TraffiX : wants to collect traffic flow data.
- **Trigger** : new traffic flow data has become available for sending.
- **Preconditions** :
 - The traffic flow meter is correctly installed.
- **Post-conditions** :
 - The traffic flow data is successfully sent and registered in the TraffiX system.
- **Main scenario** :
 1. The traffic flow meter sends the parking data to the TraffiX system.
 2. TraffiX confirms receiving the traffic flow data.
 3. TraffiX successfully registers the traffic flow data.
- **Alternative scenarios** :
 - 2b. The message containing the data is not confirmed and the parking meter tries sending it again.

3 Non-functional requirements

3.1 Availability

3.1.1 A1 : Parking data transmission failure

- **Source** : External or internal.
- **Stimulus** :
 - The external communication channel between a parking meter and the TraffiX system is failing;
 - the parking meter has failed;
 - an internal communication component in the TraffiX system has failed.
- **Artifact** : parking meter, external communication channel(s) or internal communication (sub)system(s).
- **Environment** : Normal execution.
- **Response** :
 - External (1) : the telecom operator providing the network service has signed an Service-Level Agreement (SLA) to guarantee a minimum network availability of 99.9% on a yearly basis.
 - External (2) : the parking meter user is notified if the parking data cannot be transmitted and is asked to notify an administrator of the TraffiX system.
 - Internal : TraffiX should be able to detect failing internal components and notify the TraffiX administrator when necessary.
 - TraffiX should acknowledge parking data messages and parking meters will resend the parking data when a time-out should occur based on an exponential back-off algorithm.
- **Response measure** :
 - Internal component failures should be detected within three minutes.
 - A failing communication channel should be detected within 5 minutes.

3.1.2 A2 : Internal traffic flow database failure

- **Source** : Internal
- **Stimulus** : The internal database responsible for traffic flow data fails.
- **Artifact** : Internal subsystem.
- **Environment** : Normal execution.
- **Response** :
 - The database has a guaranteed minimal up-time of 99% on a yearly basis.
 - A TraffiX Administrator is notified of the problem and can restart the failed database node.
 - Incoming traffic flow data is temporarily stored on a different node and can be inserted into the database later on.
 - Fail gracefully : users of the database are shown a clear error message that the traffic flow data is temporarily unavailable.
- **Response measure** :
 - Detection of database failure should occur within 10 seconds.
 - A TraffiX Administrator should be notified within one minute.

3.2 Performance

3.2.1 P1 : Travelling advice computation

- **Source** : External.
- **Stimulus** : Client wishes to receive timely suggestions on how to alter his/her route to arrive most efficiently to his/her destination.
- **Artifact** : Computational subsystems and traffic flow database subsystems.
- **Environment** : Normal execution.
- **Response** :
 - For each traffic node, a stream mining algorithm is applied on the rapidly evolving incoming traffic flow data to compute temporal models which can be used for the rest of the computation.
 - Optimized implementations are run for each Client request using a variant of the shortest path algorithm.
- **Response measure** :
 - A suggestion should be computed for the complete route within 2 minutes to be acceptable and ensure adequate response time towards the user.

3.2.2 P2 : Requests to the traffic flow database

- **Source** : Internal TrafficX subsystem(s).
- **Stimulus** : multiple internal subsystems send requests to the database in parallel.
- **Artifact** : Internal subsystem(s).
- **Environment** : Normal execution.
- **Response** :
 - In normal modus, the database requests are processed first-in, first-out.
 - In overload modus, the database requests are processed according to their priority (based on source and in the case of traffic flow monitors, on the importance of the traffic node).
- **Response measure** :
 - Requests originating from important traffic node monitors, are handled within 500msec.
 - All other requests are to be handled within 2000msec.

3.3 Security

3.3.1 S1 : Client database safety

In order to guarantee the privacy of TrafficX users, it is imperative that location data cannot be linked to personal client data, such as real names, licence plates, contact information et cetera.

- **Source** : External.
- **Stimulus** : Attack on the TrafficX Client database.
- **Artifact** : Internal subsystem.

- **Environment** : Normal execution.
- **Response** :
 - Database read access requires authorization by a separate instance.
 - Client accounts can only be accessed through single entry point and requires authentication.
 - Traffix uses pseudonyms to process client data.
 - The Client has to register devices from which he/she will access the Traffix system.
- **Response measure** :
 - Attacks are detected within an hour.
 - Clients are notified as soon as an attack is detected and prompted to alter their log in credentials.

3.4 Usability

3.4.1 U1 : Client user interface

- **Source** : External: the end user (client) of the Traffix system.
- **Stimulus** :
 - Learning system features : the Client should be able to learn to use the application fast.
 - Using a system efficiently : the Client should be able to make use of the application's features and services efficiently.
 - Increasing confidence and satisfaction : the Client should get good feedback while working with the application and get good results using it.
- **Artifact** : User interface on an external device.
- **Environment** : Normal execution (runtime).
- **Response** :
 - A specialized team of user interface designers is asked to design the application.
 - Usability tests are conducted to optimize and test certain aspects of the application.
- **Response measure** :
 - Overall test scores (e.g. SUS questionnaires) should result in an A-level application interface design.
 - The task time for configuring a request for travelling advice should not exceed three minutes.