
Platform for Control and Delivery of services in Next Generation Networks

DELIVERABLE 4.1

Prototype demo of a “context-aware”
application streaming service delivery





PROPRIETARY INFORMATION

©CEFRIEL 2008 All Rights Reserved

©PoliTO 2008 All Rights Reserved

This document and the data included in this document is proprietary to CEFRIEL and Torino's polytechnic, and is not to be reproduced, used, or disclosed in whole or in part to anyone without the express written permission of the parts above. The content of this document is provided for informational use only and is subject to change without notice.

Questions about this document or the features it describes should be directed to:

CEFRIEL

RECORD OF CHANGES

| Version | Date | Description |
|----------------|-------------|---------------------|
| 0.1 | 24/11/2010 | First draft version |
| 0.2 | 16/03/2011 | Second version |
| 1.0 | 28/04/2011 | Final version |

Index

| | |
|--|-----------|
| Index | 5 |
| 1 Introduction..... | 6 |
| 2 Execution Environment | 7 |
| 2.1 Database | 7 |
| 2.2 PICO Server | 7 |
| 2.3 IMS / SIP Proxy | 7 |
| 2.1 PICO Mobile..... | 8 |
| 3 Prototype Scenario..... | 9 |
| 3.1 Actors | 9 |
| 3.2 Context..... | 10 |
| 3.3 Basic scenario with Storyboard | 11 |
| 3.4 Alternative scenario..... | 39 |
| Index of Figures..... | 42 |

1 Introduction

This document shows the Execution Environment (EE) of the PICO Prototype and the execution of a scenario. The scenario have storyboards that describe its behavior. The Execution Environment contains 4 main components:

- the Database used to storage the information needed to PICO Server;
- the PICO Server which represents the application logic;
- the Proxy which allows the IMS communications between the actors;
- the Client which runs the mobile application capable of communicating with PICO Server through IMS.

A real use case is analyzed using a specific scenario. At the beginning the actors of the scenario are considered, then the context of the emergency is analyzed and finally the steps of the scenario are described. Each step comes with a screenshot which shows graphically the behavior of the application for the server and mobile side.

2 Execution Environment

2.1 Database

The database is one of the components of the PICO architecture. It stores the details of the emergency, applications, contexts etc. As described in the D 3.2, in PICO there are dynamic data that frequently change (Location, battery level of device etc.) and data which need to be stored such as crisis/emergency details, applications etc.

PICO uses PostgreSQL 8.3 as database solution. For the prototype it is installed on a dedicated machine, but it could be installed on the same machine running PICO Server.

2.2 PICO Server

Pico Server is a JAVA EE application which runs on a container such as Tomcat. In the prototype Tomcat 6.5 has been used to run the application. For the IMS part, we used doubango libraries which have been compiled for Windows 32 (XP and Vista). In this way PICO server is able to communicate with the client using all protocols offered by IMS. The PICO Server has a configuration file need to be set before launching the application. In the configuration file there are information related to the database, the SIP account, the proxy settings, the URL of repository and other useful information needed to PICO Server.

During the start phase, PICO Server performs several initializations, among these also an IMS/SIP registration using the account pico001@cefriel.it. Since then the PICO Server is reachable from other clients using the account above.

2.3 IMS / SIP Proxy

In the prototype Mjsip acts as IMS/SIP Proxy. All clients such as PICO Server and all PCSD Users have to register to the proxy in order to communicate. The proxy has been installed to another machine (different from PICO Server). It is listening on standard SIP port (5060). All the components of the architectures such as PICO Server, mobile clients and the proxy work together if they are on the same subnet. To avoid this, a NAT traversal mechanism has to be implemented but it is out of the purpose of the Prototype.

The proxy is configured to accept the registrations with the following syntax: picoXXX@cefriel.it where XXX identifies the number of the client. In the prototype we have 6 actors which use account from pico001@cefriel.it to pico006@cefriel.it.

2.1 PICO Mobile

The PICO Mobile is an Android application able to run on different devices which support Android higher than 1.6. Indeed the device could be a tablet or a smartphone indifferently. In the prototype the following devices have been used:

- Samsung Galaxy TAB
- Samsung Galaxy S
- Nexus One
- HTC G1

The device has to have the wireless and the GPS enabled in order to send the location information to the PICO Server. The application presents a configuration section in which configurations will be saved. The most important parameters to be set are the PICO server (PSC Host), the PICO Server account (ex: pico001@cefriel.it), the SIP PSCD User account (ex: pico004@cefriel.it), the password of the SIP account.

3 Prototype Scenario

3.1 Actors

The actors involved for the Prototype Scenario are:

- **PICO Server (pico001@cefriel.it):** It is the main application of the PICO Architecture with all the application logic. It provides the reasoner which is always running and performs reasoning analyzing the context of the emergency and the context of the PCSD Users connected to. It receives all the PCSD Users contexts and suggests the type of communication between them (Audio/Chat/Video) and the applications relevant to the user. Each mobile client has the PICO server among the buddy list (pico001@cefriel.it)
 - **Operator (pico002@cefriel.it):** It is the operator that corresponds to an agent at the command station. He uses an IMS client based also on doubango libraries named Boghe. He performs a IMS registration to the proxy and is available at pico002@cefriel.it to receive calls. He can also open an emergency using a web interface.
 - **Policeman 1 (pico003@cefriel.it):** It is the first policeman. It will be the only one to be associated to just one emergency without any other users. It will be the first actor of the prototype scenario which will show an example of contextual application sharing.
 - **Policeman 2 (pico004@cefriel.it):** It is the second policeman associated to the second emergency. It will show some communications features and application sharing.
 - **Firefighter (pico005@cefriel.it):** It is the firefighter associated to the second emergency. It will show some communication features.
 - **Paramedic (pico006@cefriel.it):** It is the paramedic associated to the second emergency. It will show some communication features, application sharing and QR Code features.
 - **Injured people:** There will be some injured people with a QR code information which will have personal and medical information such as name, number to call, blood group and so on.
-

3.2 Context

The prototype scenario takes place in Milan. At the beginning there will be no opened emergency in the system (database of the PICO Server). During the course of the scenario two kinds of emergencies, a car accident and a fire, will be opened. The first emergency is located in *Stazione Centrale* and will be opened directly from the device of the users. The user involved in this emergency (Police 1) will have a forced position (near *Stazione Centrale*) to allow the simulation of the Prototype. The second emergency is located in *Piola* and will be opened by an Operator. In the second emergency more actors will be involved and each of them will have a true position since the real place in which the users are, is close to *Piola*.

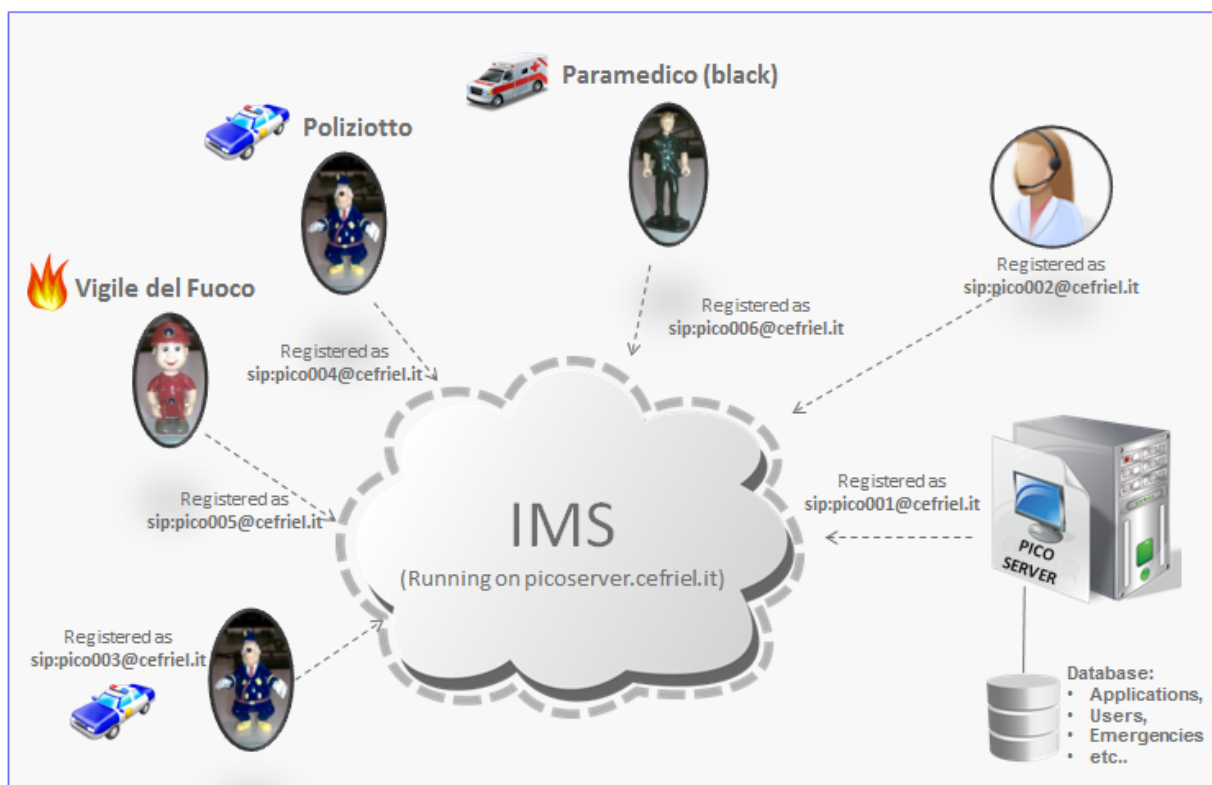
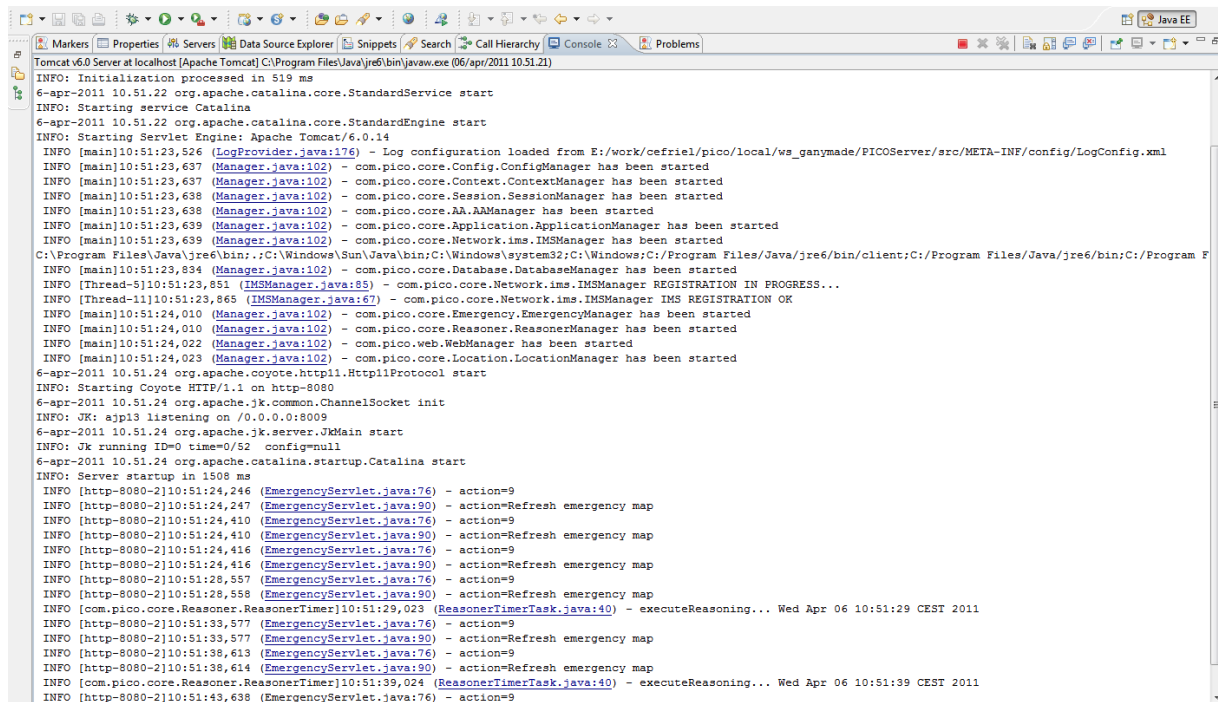


Figure 1 - Actors

Also, the second emergency will have some injured people with a QRCode. The QRCode will embed some personal and medical information such as number to call, name, age, blood group and so on. This QRCode will be scanned and used to send the information between Public Safety Communication Users.

3.3 Basic scenario with Storyboard

In the figure 2 the PICO server has been started. The server performs an IMS registration to the IMS proxy and shows all log messages pertaining. The server manages all incoming requests from the users (context updates, download of the applications and so on)



```
Tomcat v6.0 Server at localhost [Apache Tomcat/6.0.14]
INFO: Initialization processed in 519 ms
6-apr-2011 10:51:22 org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
6-apr-2011 10:51:22 org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.14
INFO [main]10:51:23,526 (LogProvider.java:176) - Log configuration loaded from E:/work/cefriel/pico/local/wa_ganymade/PICOserver/src/META-INF/config/LogConfig.xml
INFO [main]10:51:23,637 (Manager.java:102) - com.pico.core.Config.ConfigManager has been started
INFO [main]10:51:23,637 (Manager.java:102) - com.pico.core.Context.ContextManager has been started
INFO [main]10:51:23,638 (Manager.java:102) - com.pico.core.Session.SessionManager has been started
INFO [main]10:51:23,638 (Manager.java:102) - com.pico.core.AA.AAManager has been started
INFO [main]10:51:23,639 (Manager.java:102) - com.pico.core.ApplicationManager has been started
INFO [main]10:51:23,639 (Manager.java:102) - com.pico.core.Network.ims.IMSManager has been started
C:\Program Files\Java\jre6\bin;.C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Program Files\Java\jre6\bin;client;C:\Program Files\Java\jre6\bin;C:\Program F
INFO [main]10:51:23,834 (Manager.java:102) - com.pico.core.Database.DatabaseManager has been started
INFO [Thread-5]10:51:23,851 (IMSManager.java:85) - com.pico.core.Network.ims.IMSManager REGISTRATION IN PROGRESS...
INFO [Thread-11]10:51:23,865 (IMSManager.java:67) - com.pico.core.Network.ims.IMSManager IMS REGISTRATION OK
INFO [main]10:51:24,010 (Manager.java:102) - com.pico.core.Emergency.EmergencyManager has been started
INFO [main]10:51:24,010 (Manager.java:102) - com.pico.core.Reasoner.ReasonerManager has been started
INFO [main]10:51:24,022 (Manager.java:102) - com.pico.web.WebManager has been started
INFO [main]10:51:24,023 (Manager.java:102) - com.pico.core.Location.LocationManager has been started
6-apr-2011 10:51:24 org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
6-apr-2011 10:51:24 org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
6-apr-2011 10:51:24 org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/52 config=null
6-apr-2011 10:51:24 org.apache.catalina.startup.Catalina start
INFO: Server startup in 1508 ms
INFO [http-8080-2]10:51:24,246 (EmergencyServlet.java:76) - action=9
INFO [http-8080-2]10:51:24,247 (EmergencyServlet.java:90) - action=Refresh emergency map
INFO [http-8080-2]10:51:24,410 (EmergencyServlet.java:76) - action=9
INFO [http-8080-2]10:51:24,410 (EmergencyServlet.java:90) - action=Refresh emergency map
INFO [http-8080-2]10:51:24,416 (EmergencyServlet.java:76) - action=9
INFO [http-8080-2]10:51:24,416 (EmergencyServlet.java:90) - action=Refresh emergency map
INFO [http-8080-2]10:51:28,557 (EmergencyServlet.java:76) - action=9
INFO [http-8080-2]10:51:28,558 (EmergencyServlet.java:90) - action=Refresh emergency map
INFO [com.pico.core.Reasoner.ReasonerTimer]10:51:29,023 (ReasonerTimerTask.java:40) - executeReasoning... Wed Apr 06 10:51:29 CEST 2011
INFO [http-8080-2]10:51:33,577 (EmergencyServlet.java:76) - action=9
INFO [http-8080-2]10:51:33,577 (EmergencyServlet.java:90) - action=Refresh emergency map
INFO [http-8080-2]10:51:38,613 (EmergencyServlet.java:76) - action=9
INFO [http-8080-2]10:51:38,614 (EmergencyServlet.java:90) - action=Refresh emergency map
INFO [com.pico.core.Reasoner.ReasonerTimer]10:51:39,024 (ReasonerTimerTask.java:40) - executeReasoning... Wed Apr 06 10:51:39 CEST 2011
INFO [http-8080-2]10:51:43,638 (EmergencyServlet.java:76) - action=9
```

Figure 2 - Pico Server Starting

Figure 3 shows the initial map with no users. It means that no PSCD users have performed an authentication to the PICO Server (IMS Registration)

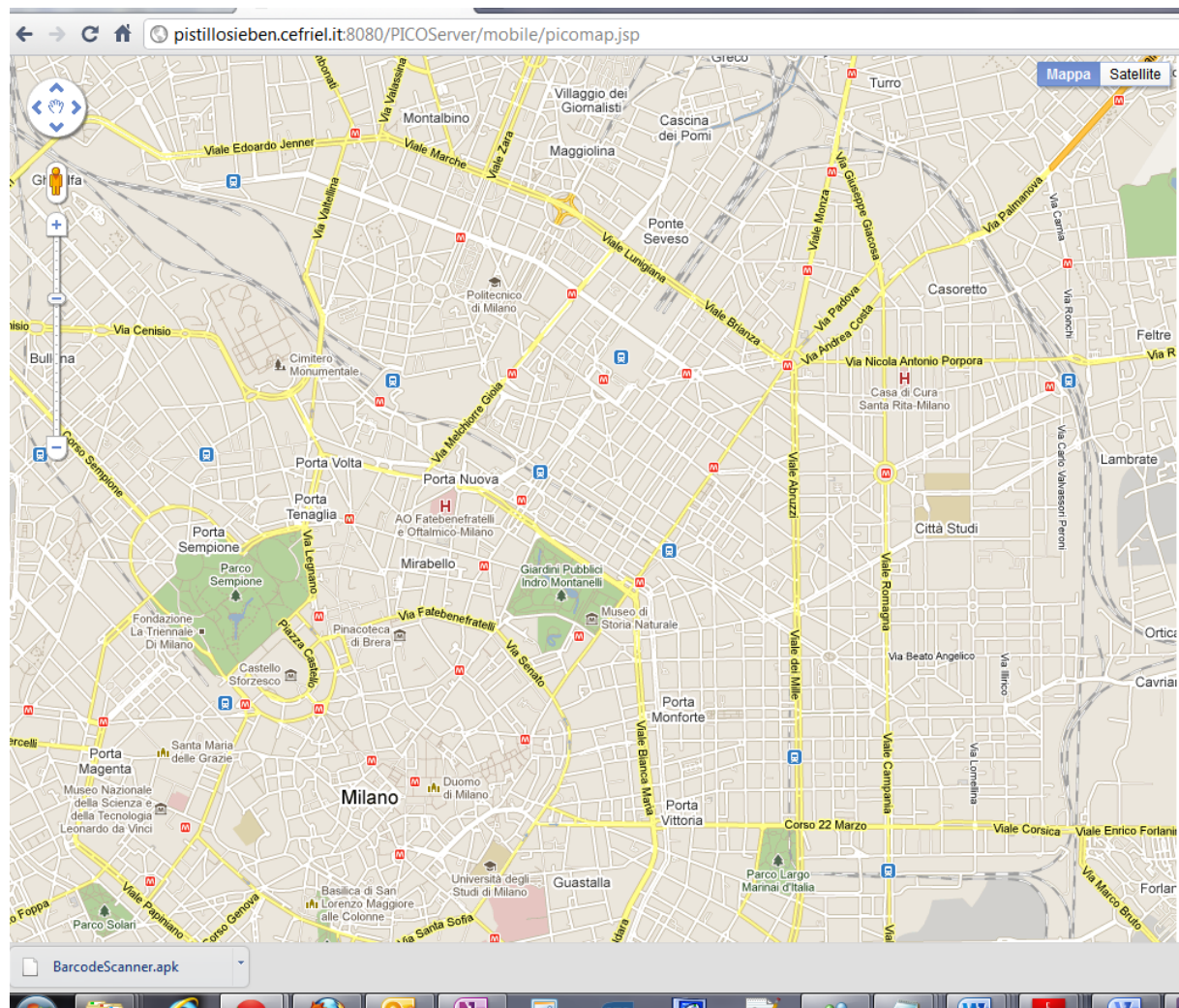


Figure 3 - Initial Pico

The first user (the policeman) activates the PICO Mobile application from its device. The application will perform an IMS registration and will send the first context update to the PICO Server. Among the update there will be a specific scope containing the location of the policeman. The map will show the location of the user connected to.

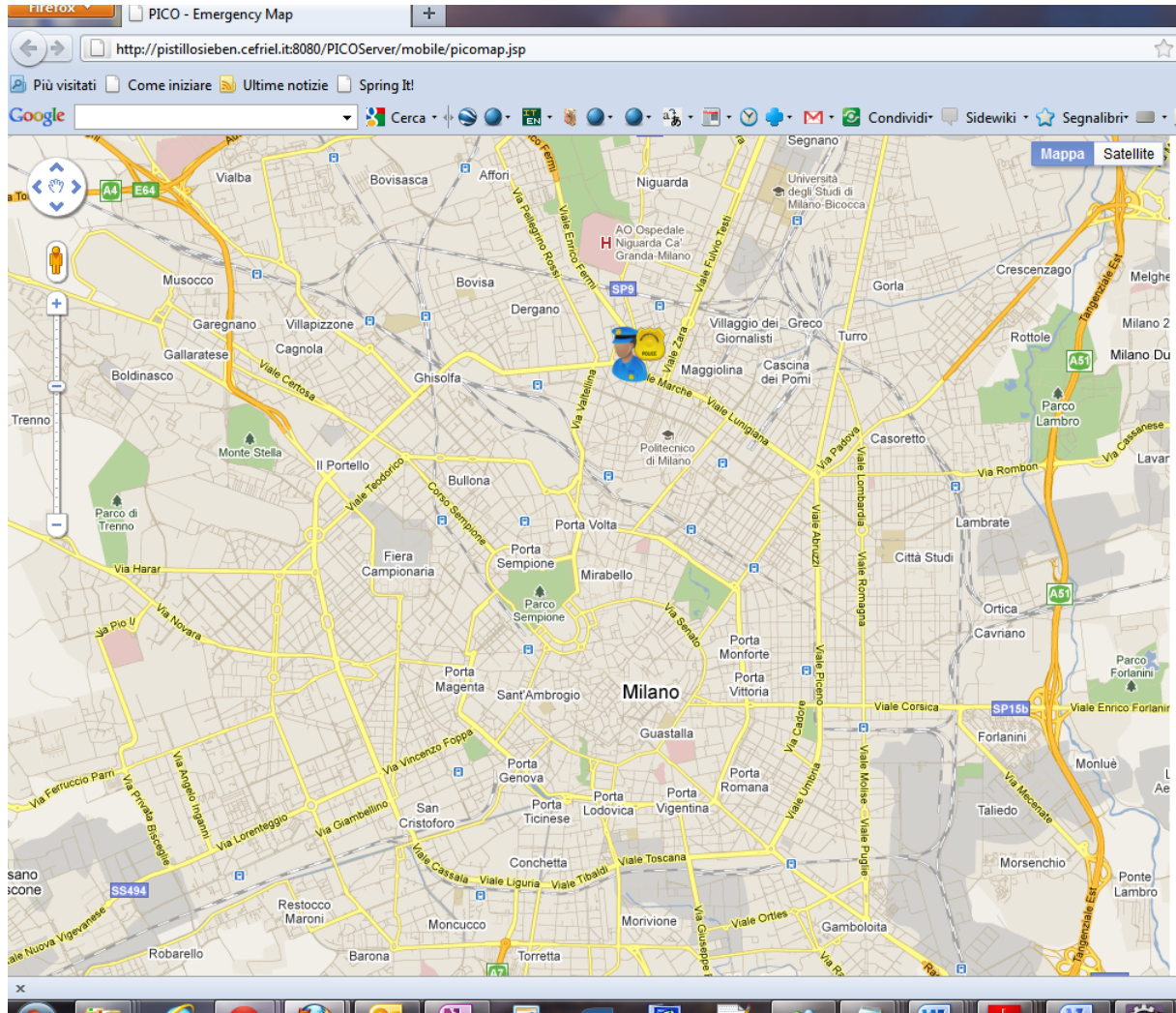


Figure 4 - Map with first police

From the user perspective (mobile application of the policeman) there are 3 main sections. The sections are accessible via a tab navigator which shows:

- Emergency tab (dashboard of the emergency). It has 3 subsections, on the top the user and emergency details. On the left there is the buddy list and in the middle the map of the emergency.
- Application tab contains the contextual application list proposed by PICO Server)
- Resources (not implemented in this prototype)

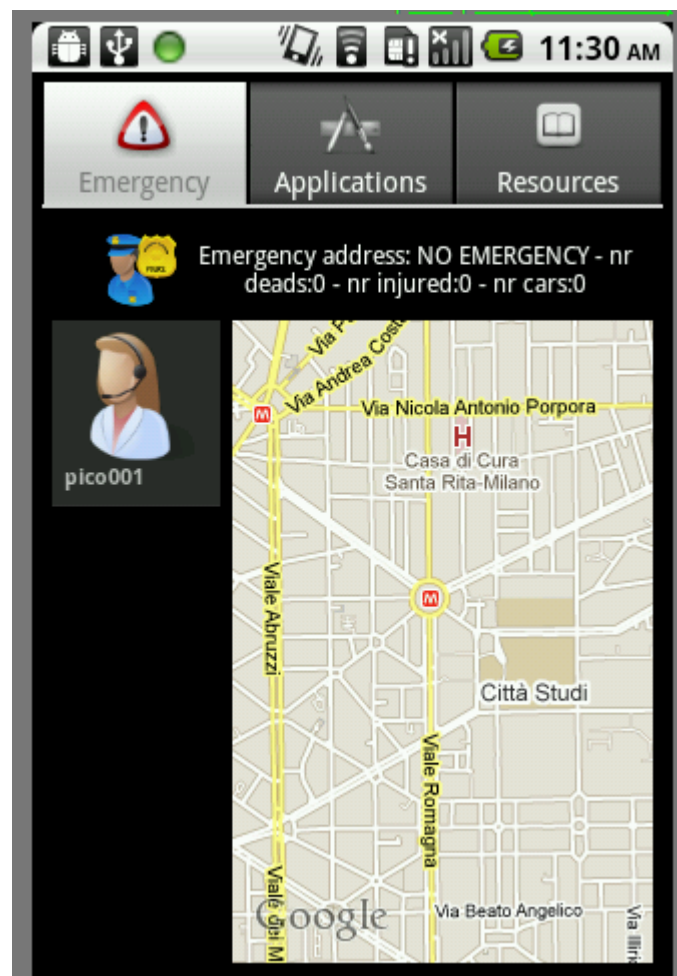


Figure 5 - First Police Screen

The first user can open an emergency directly from its device. Clicking on operator which is pico001 a form containing all relevant information about the emergency will be shown.



The screenshot displays the 'PICO Mobile' application interface. At the top, a status bar shows various icons and the time '11:34 AM'. Below this, a header bar contains the text 'PICO Mobile' and three buttons: 'Back', 'Open Emergency', and 'Info'. The main content area is a form titled 'Open Emergency'. It features three input fields: 'Name of the Emergency:' (with a green dot icon to its left), 'Description:', and 'Date:'. The 'Date:' field is pre-filled with '04/06/2011'. At the bottom of the form, there is a date selector showing 'April 2011' with left and right navigation arrows. The bottom of the screen features a navigation bar with four buttons: 'Back', 'Menu', 'Home', and 'Search'. A green dot is visible on the 'Back' button.

Figure 6 - Opening an Emergency from device

The figure 7 shows the crisis which has been created with the details inserted by the first user (the policeman)

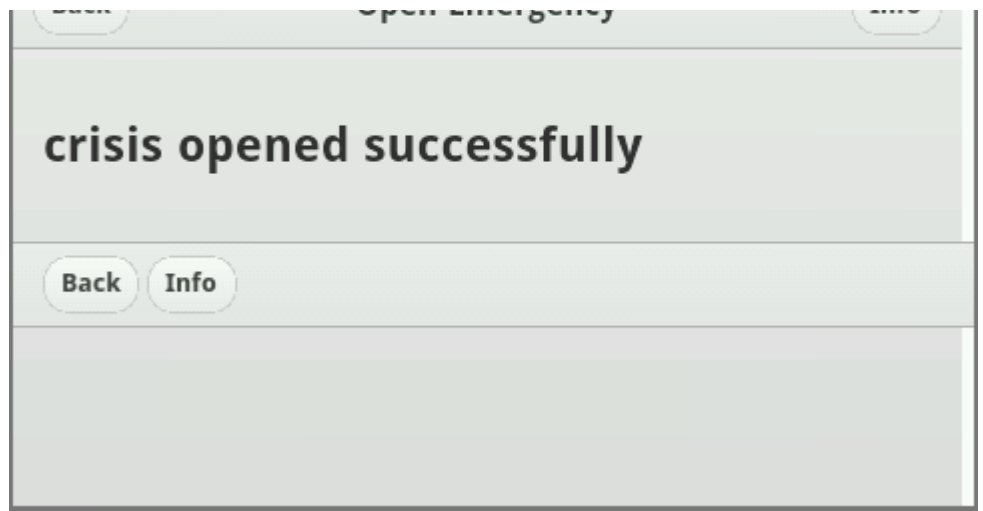


Figure 7 - Feedback Emergency

The figure shows the automatic association between the user and the emergency created. The map shows to the user the best path to reach the location of the emergency.

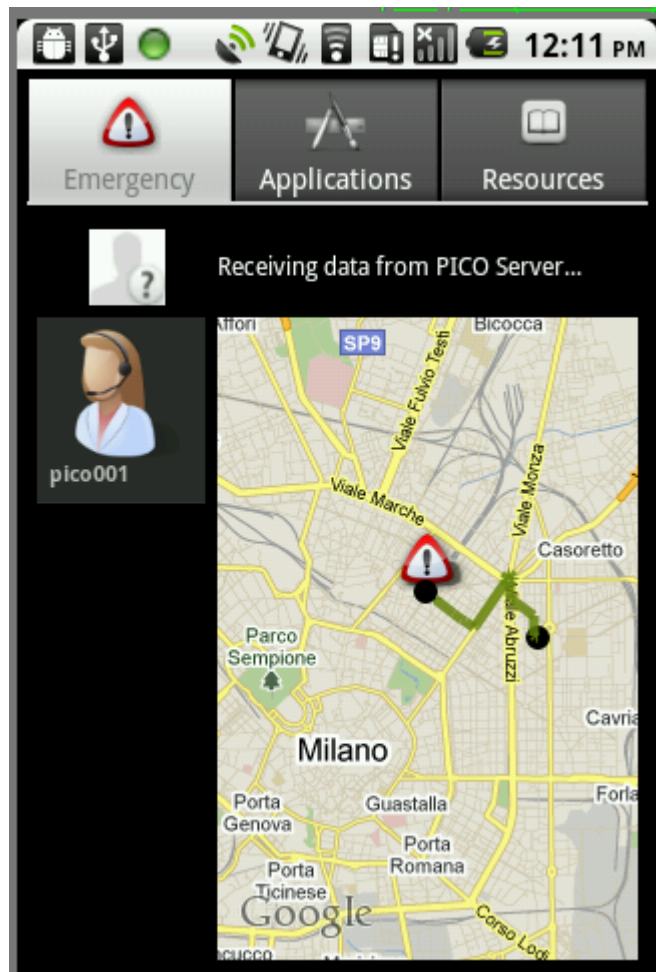


Figure 8 - System auto association between the user and the emergency

From the operator perspective (operator station), there is an emergency map which shows all emergencies created and all users associated to. In this case the first user (the policeman) has been

associated to the first emergency. The association is represented by a line between the emergency location and the user location.

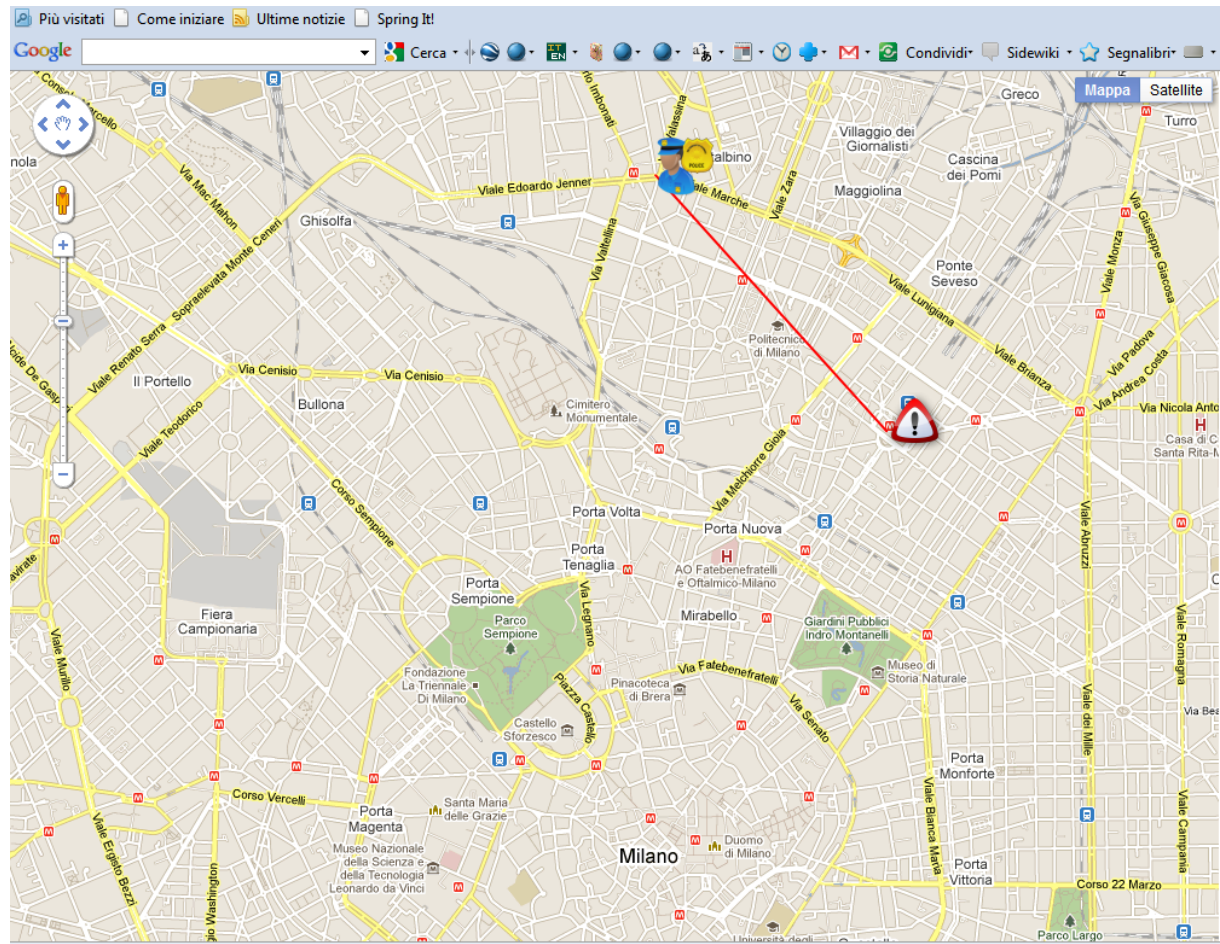


Figure 9 - Showing the association on the map

The Pico Server start the reasoning based on the context of the user (location, type, battery level) and the context of the emergency (how many injured, fire, water, outside, inside) and proposes a list of application pertaining to both contexts.

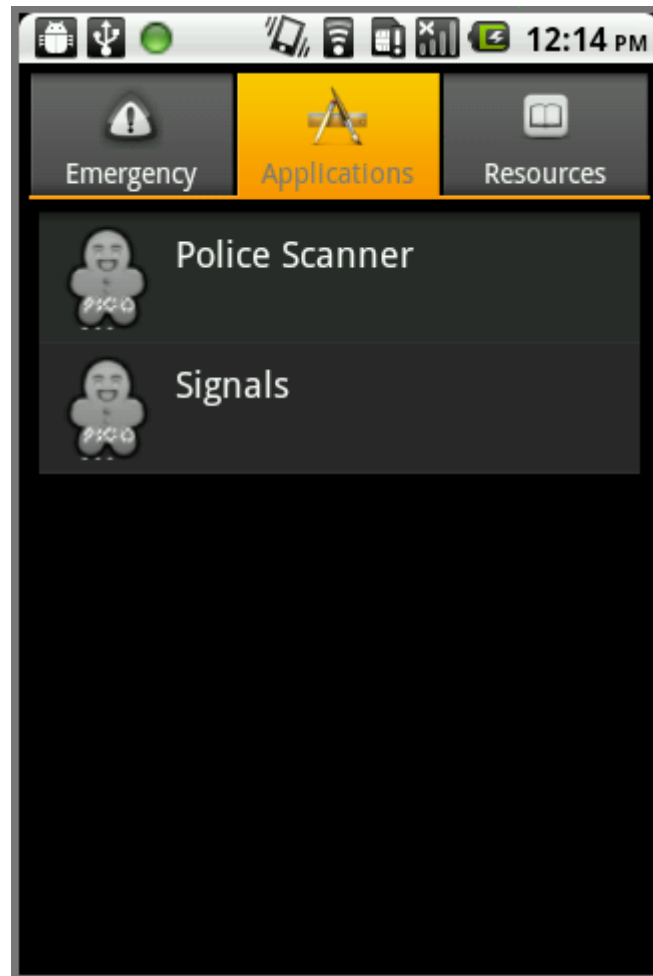


Figure 10 - List of application

The users can select an application and can start the download of the selected application using the MSRP protocol.

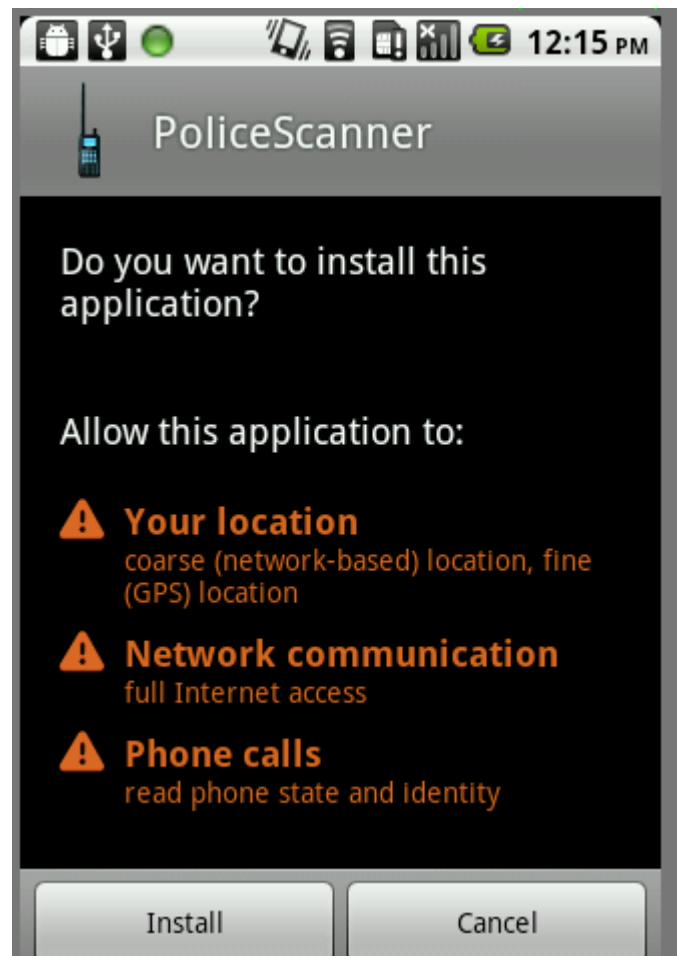


Figure 11 - Installation and confirmation

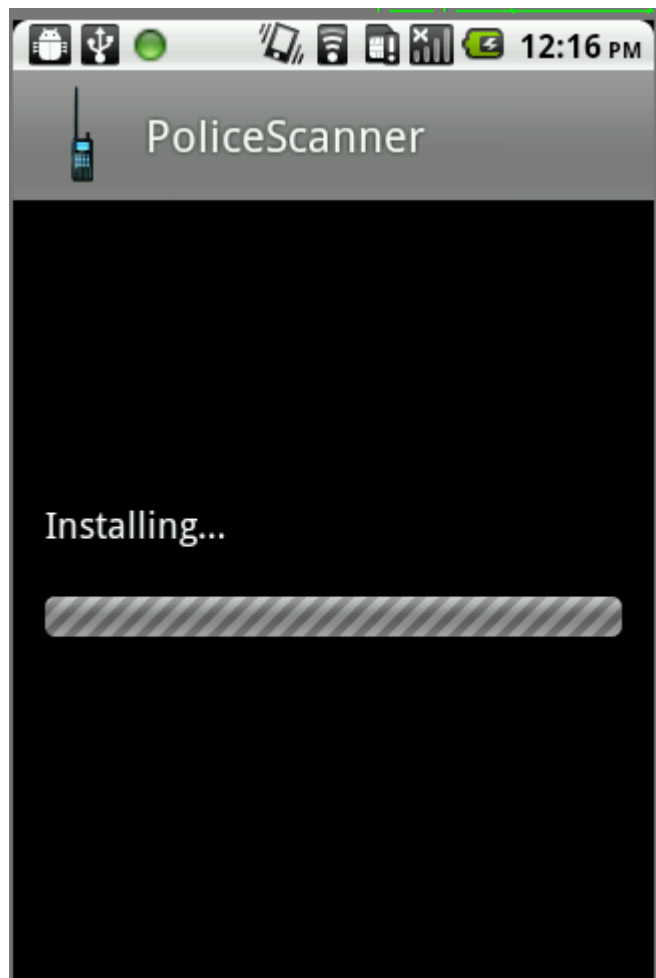


Figure 12 - Installation of an application

In the application list tab, the icon of the application which has been installed will change the color because it will be available on the device. Simply touching the row of the application, PICO Mobile will execute it.

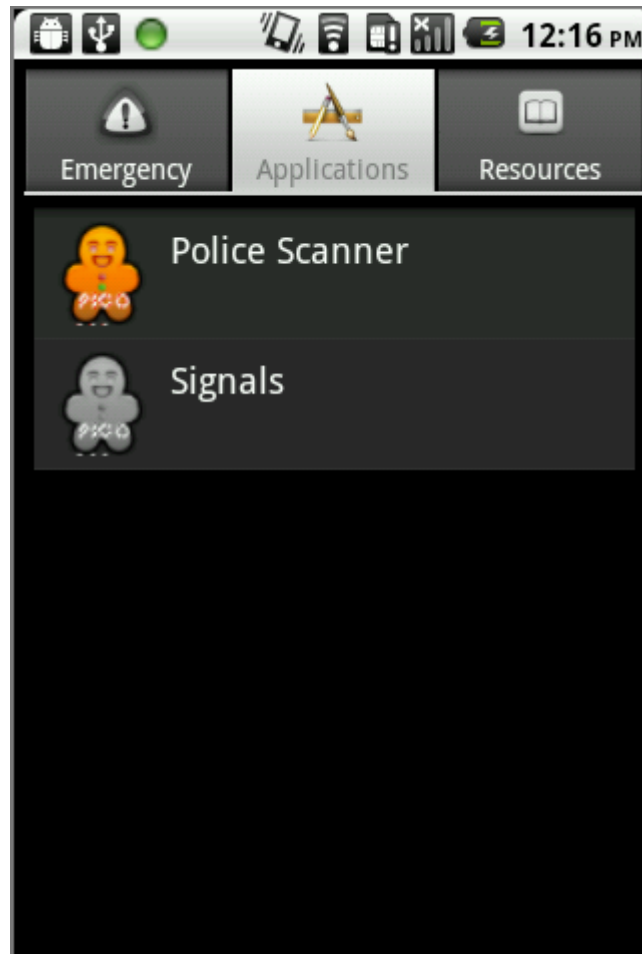


Figure 13 - Application installed available using a feedback

This figure represents the screen of the second policeman. At the beginning there will be no emergencies associated to the second user as shown in the figure.

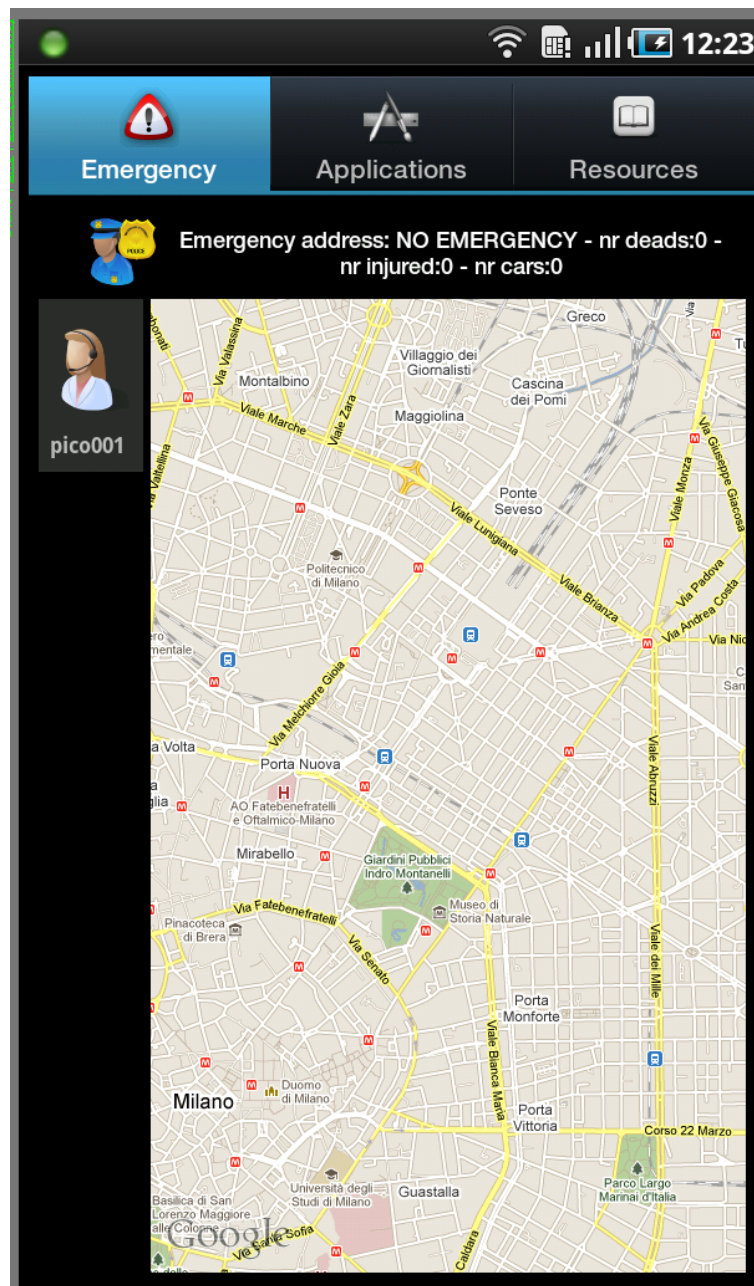


Figure 14 - Second Policeman screen

In the map the operator will see the first policeman associated to the first emergency and the second policeman with no association because the first emergency already has a policeman. Anyway it is possible to set the emergency with more policemen.

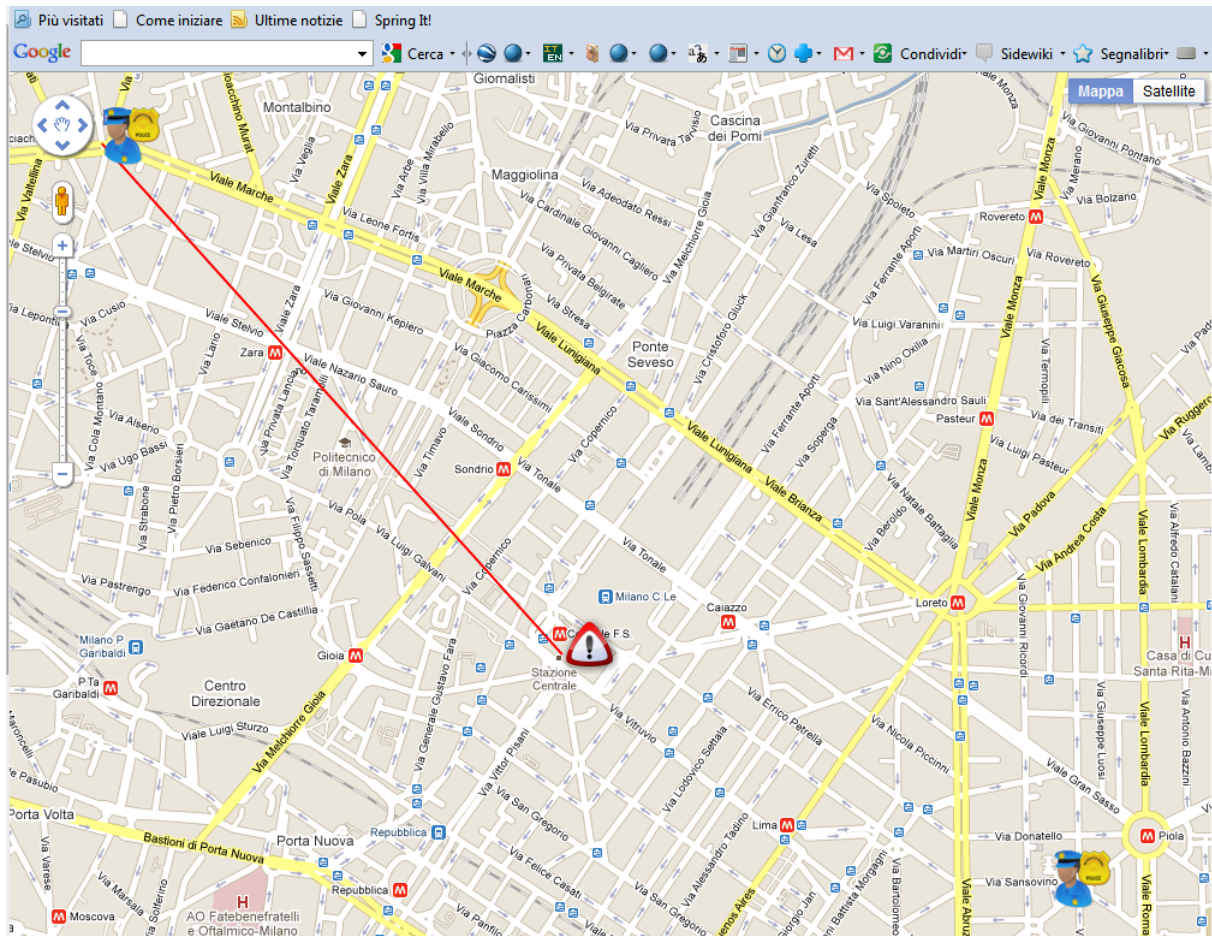


Figure 15 - First and second policeman on the map

By pressing the operator icon, the second policeman can acts some operations such as Open an Emergency, call the selected user (Audio,Video,Chat) and send an application. In this case the second policeman will call (audio) the operator to communicate the emergency details.

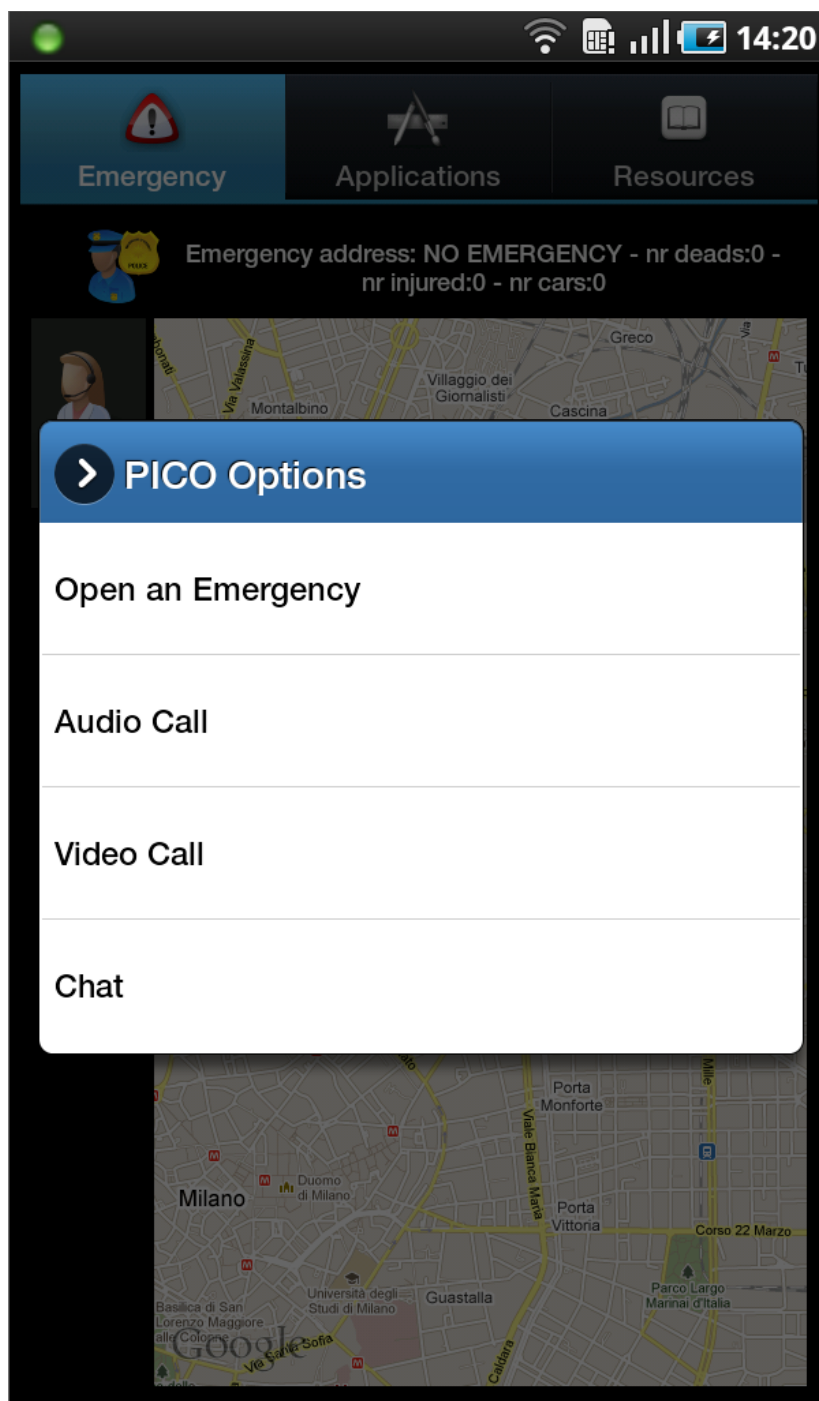


Figure 16 - Pico server options

From a desktop station, the operator will receive the audio call from the second policeman using an IMS Client.

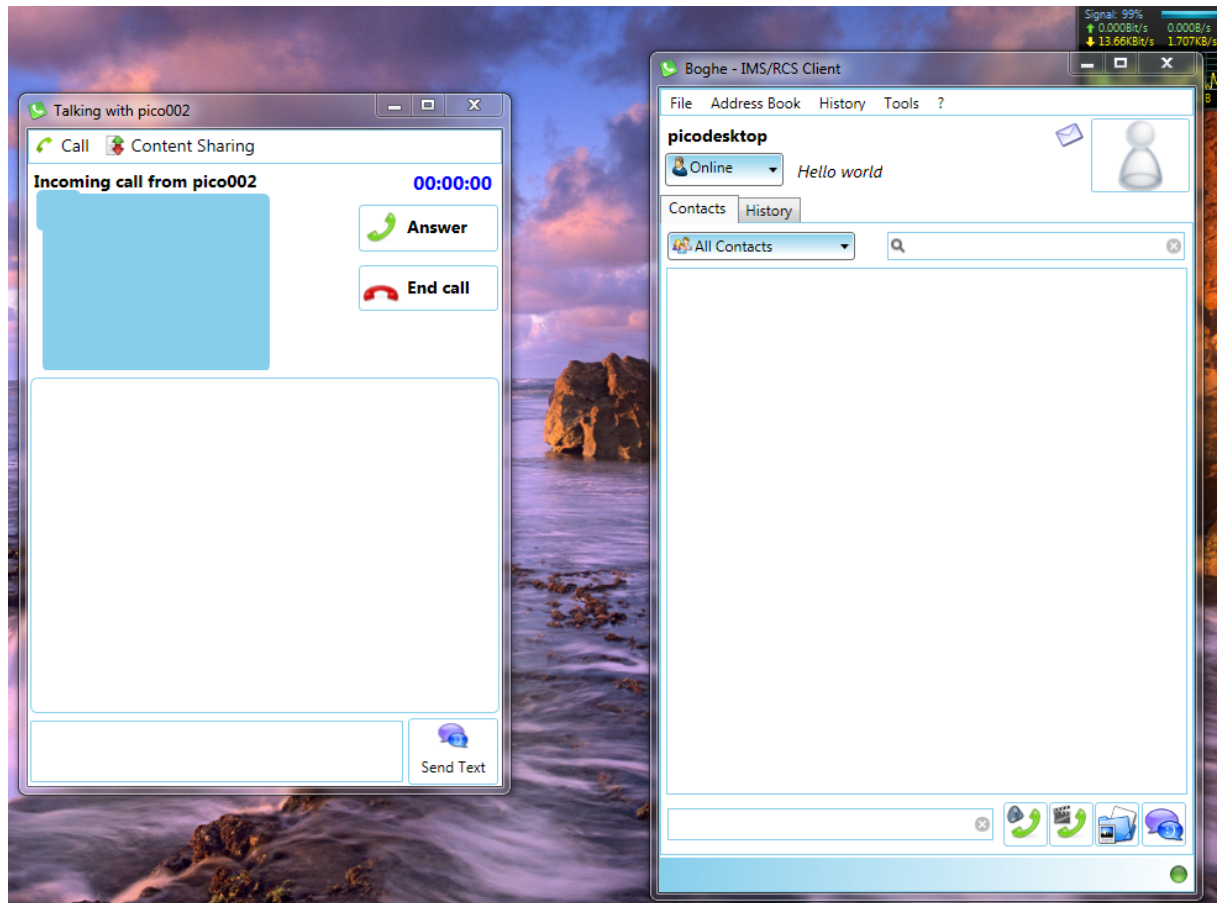


Figure 17 - Incoming call from second policeman to the operator

This is the screen of the second policeman during the call to the operator.



Figure 18 - In Call option

The operator will have a web interface to insert all emergency details communicating by the second policeman. It is the same interface available to the mobile client because it has been written using web technologies.

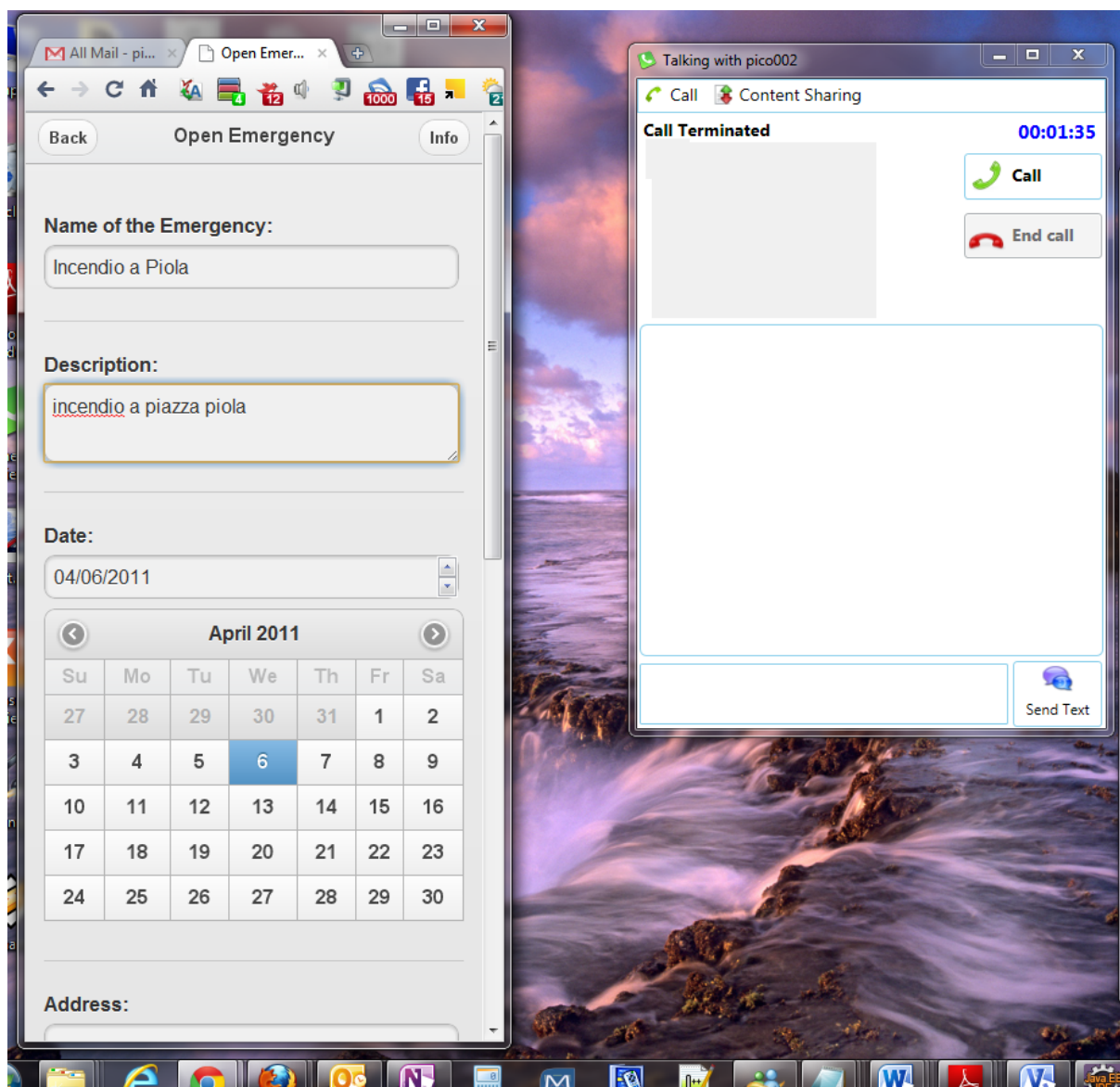


Figure 19 - Opening an emergency from desktop OPERATOR

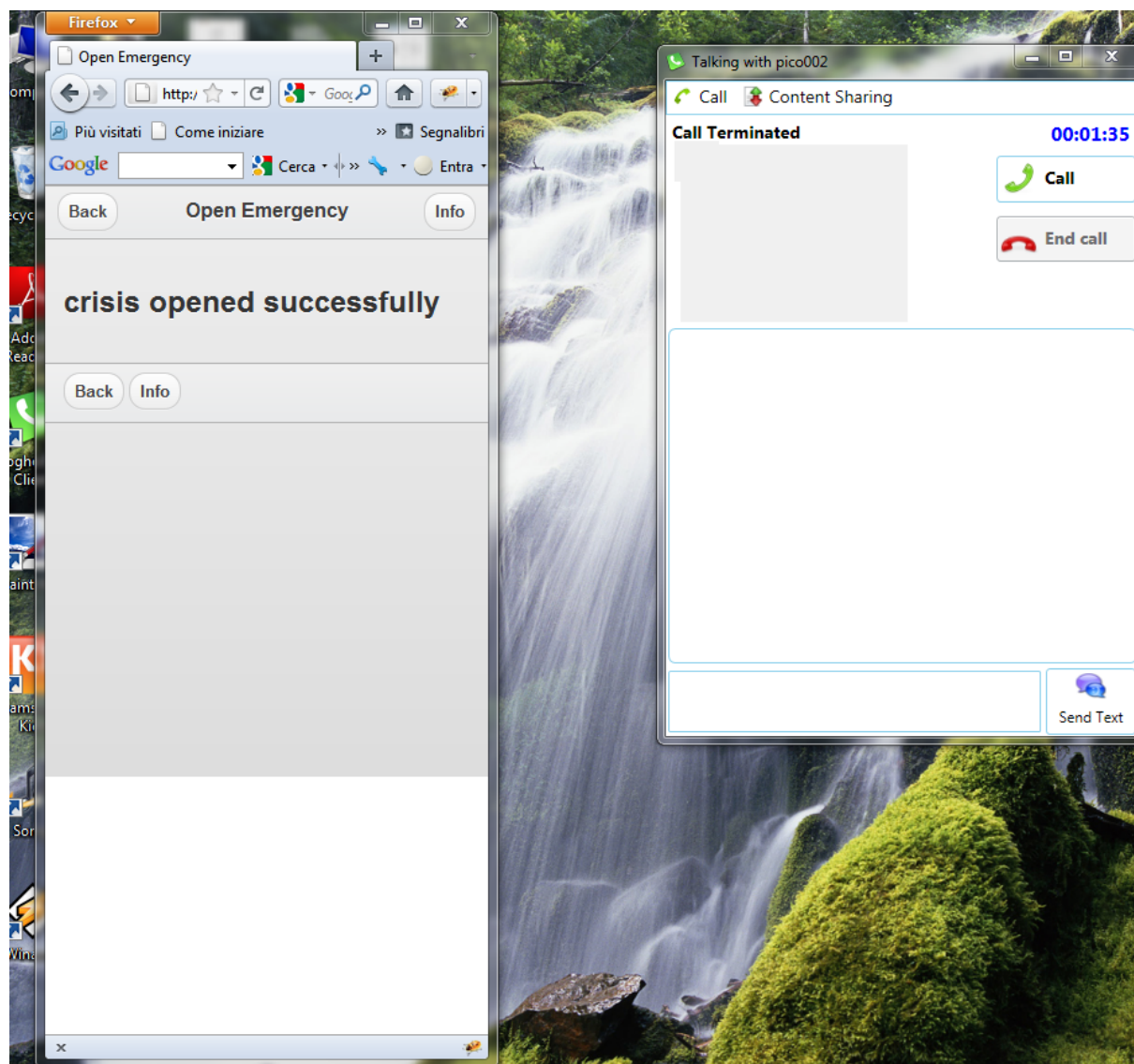


Figure 20 - Crisis opened from operator

Once the second emergency is created, the system will automatically associate the second policeman to the second emergency. On the screen of the second policeman it will be shown the best path to reach the emergency location.

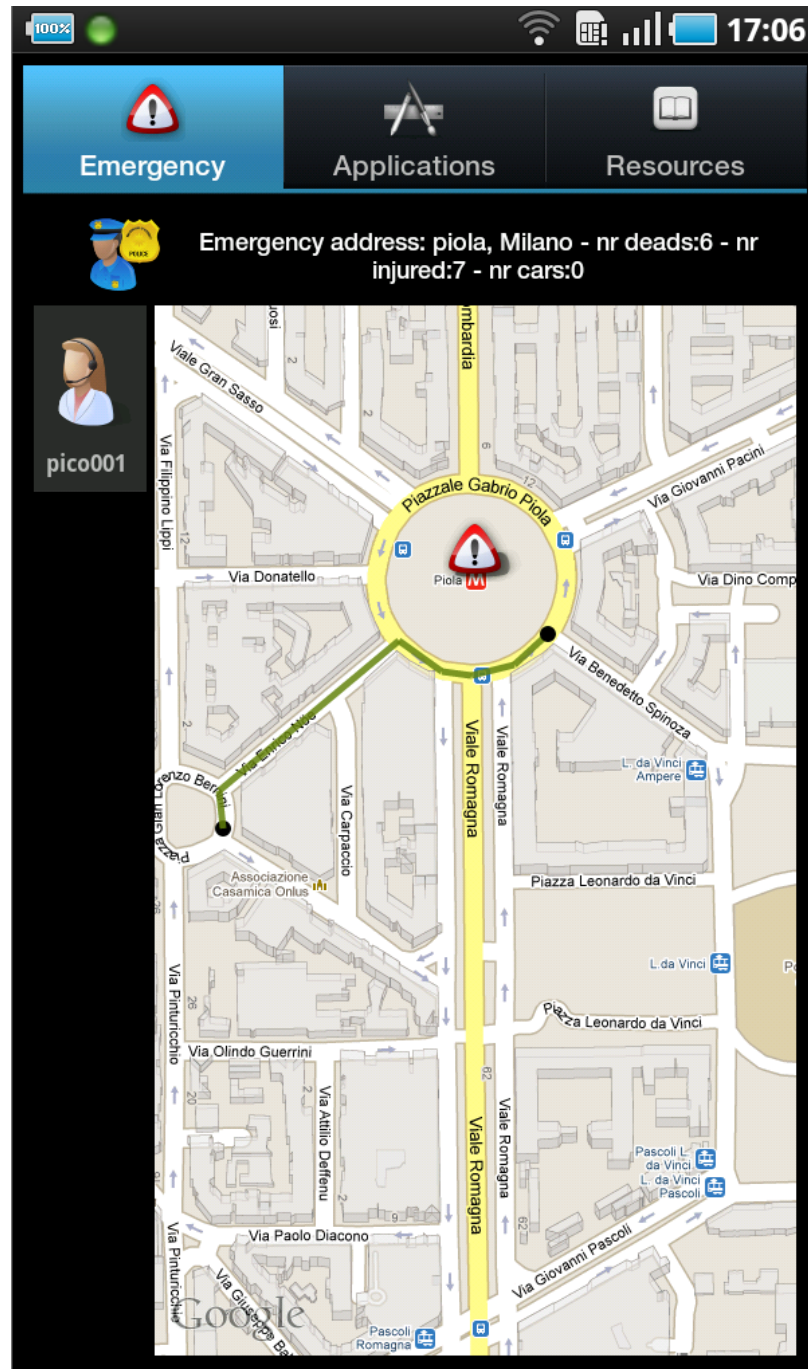


Figure 21 - Association between the second policeman and the second emergency

This time, the operator map will show the first policeman associated to the first emergency and the second policeman associated to the second emergency.

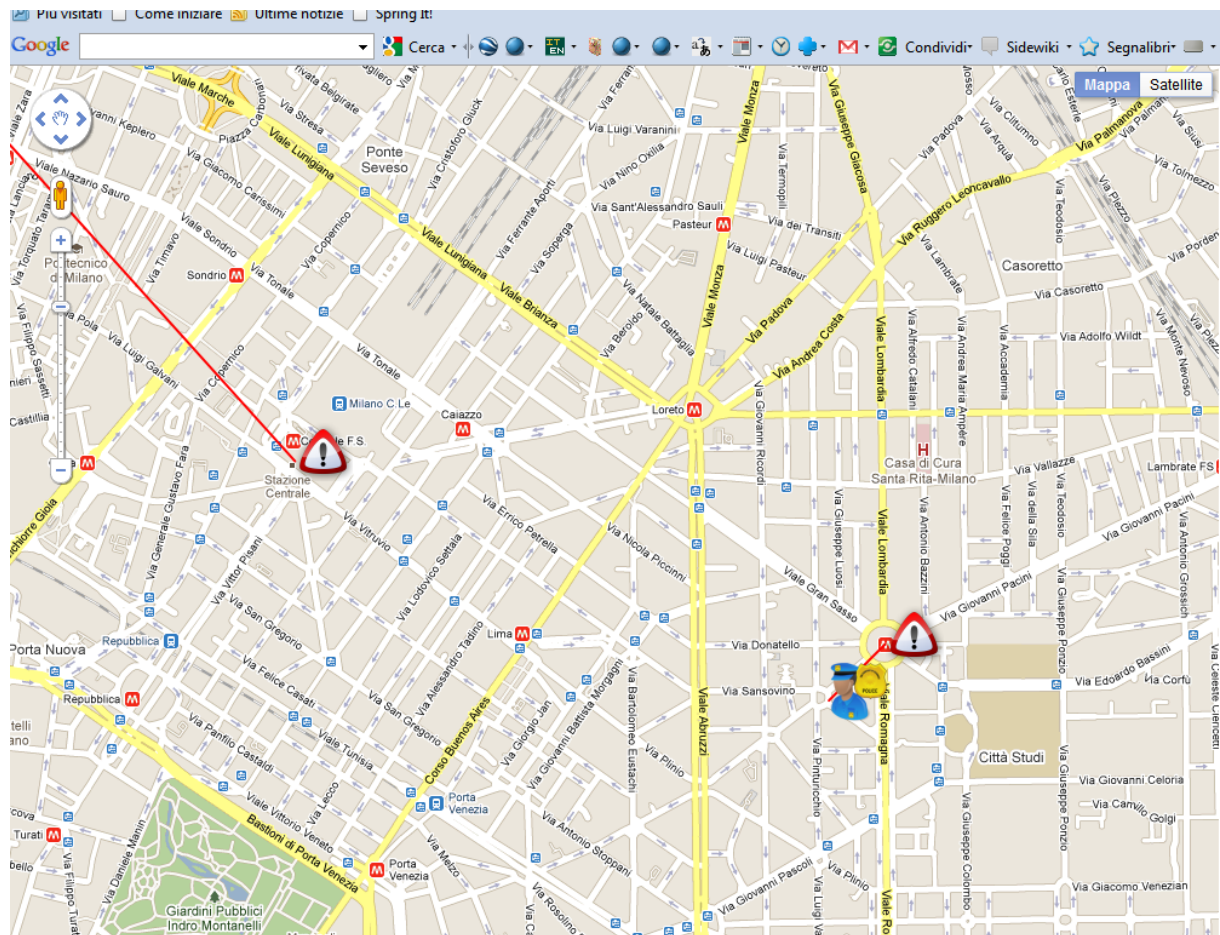


Figure 22 - Association on the map

This figure shows the screen of the second user, the firefighter. When he enters into the system, PICO server will automatically associate the firefighter to the second emergency.



Figure 23 - Fireman screen

The fireman can perform several actions such as send an application to the policeman.

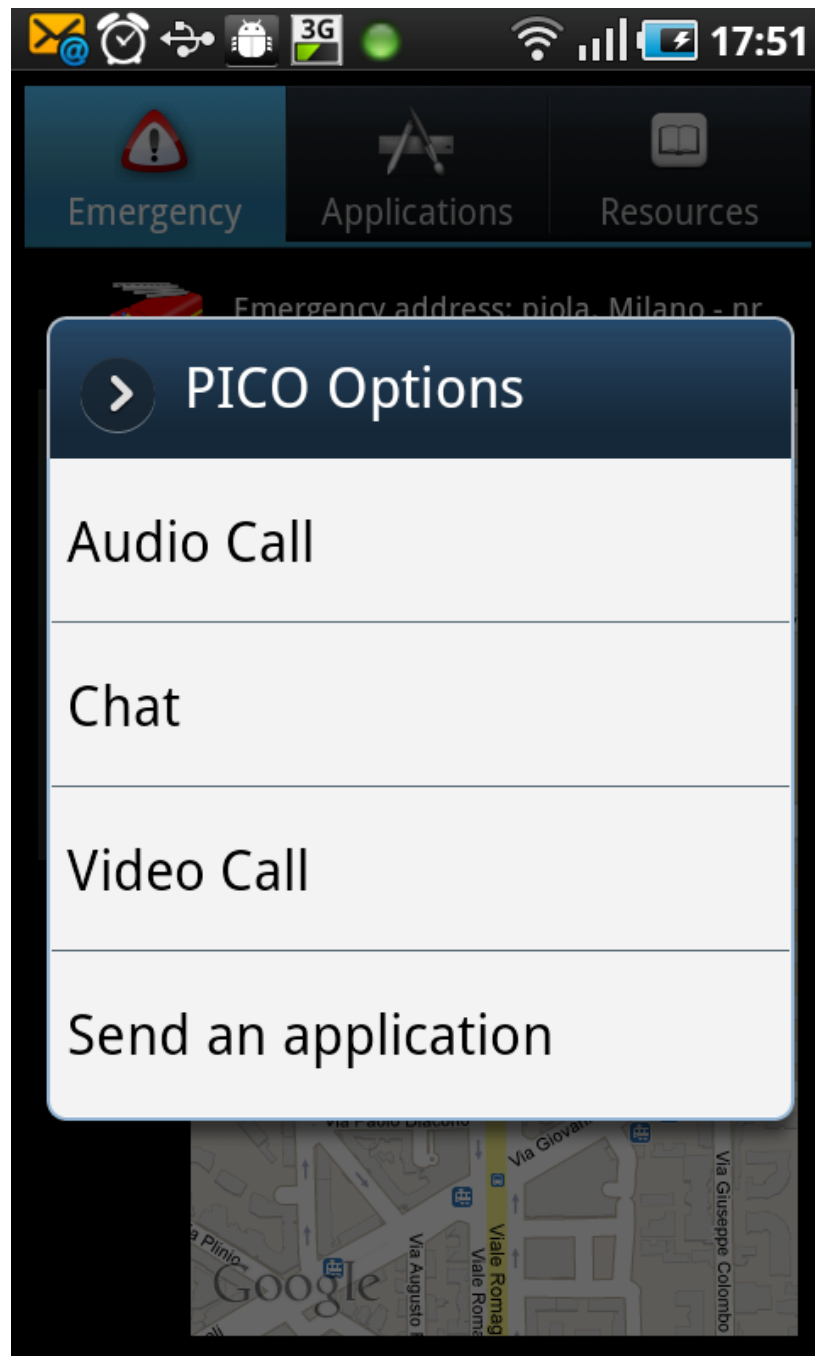


Figure 24 - Option between users

In this screen the user can decide to send an application only if it is available from its device.

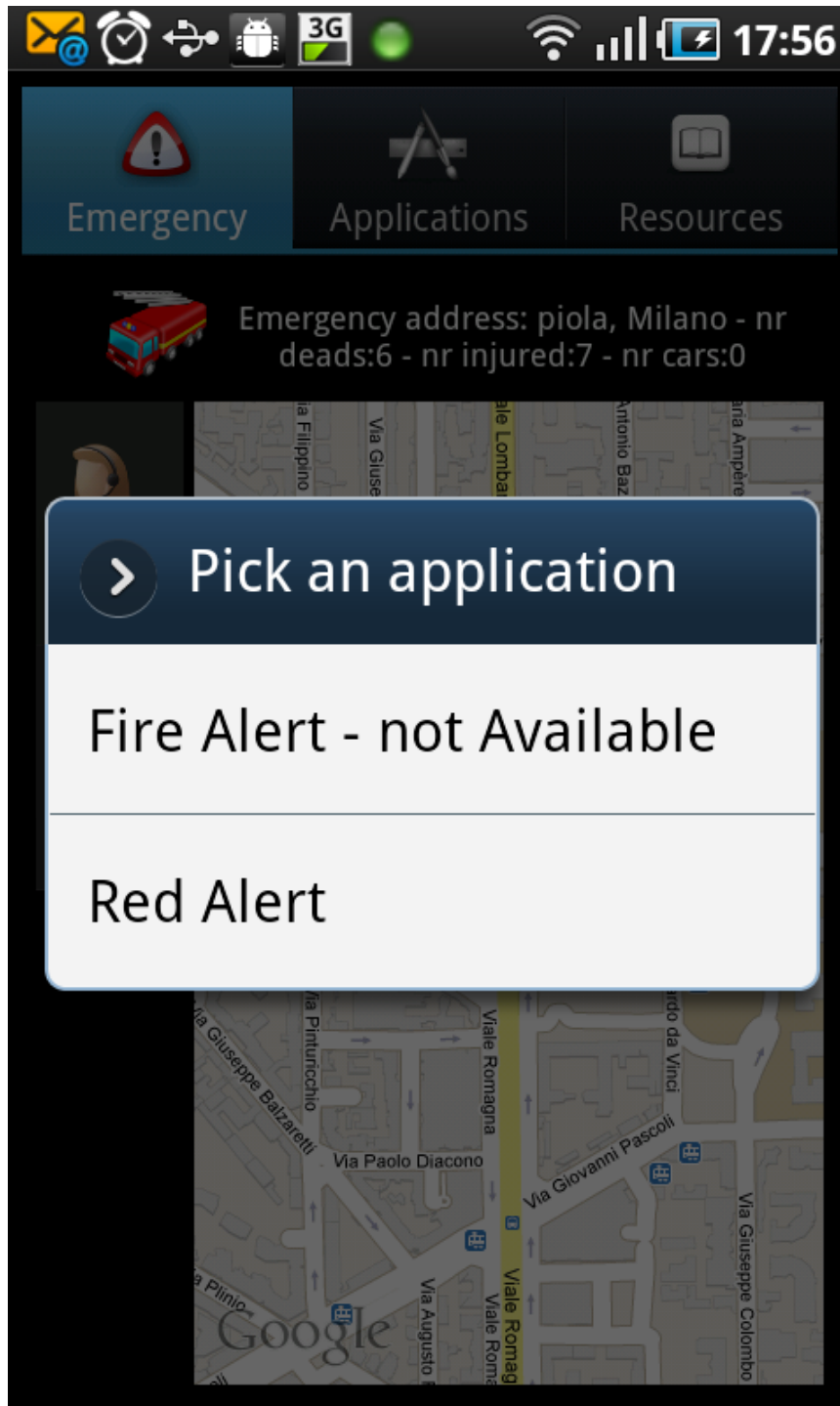


Figure 25 - Sending an application between users

The second policeman can decide to use barcode scanner to obtain the medical and useful information about the injured of the emergency.

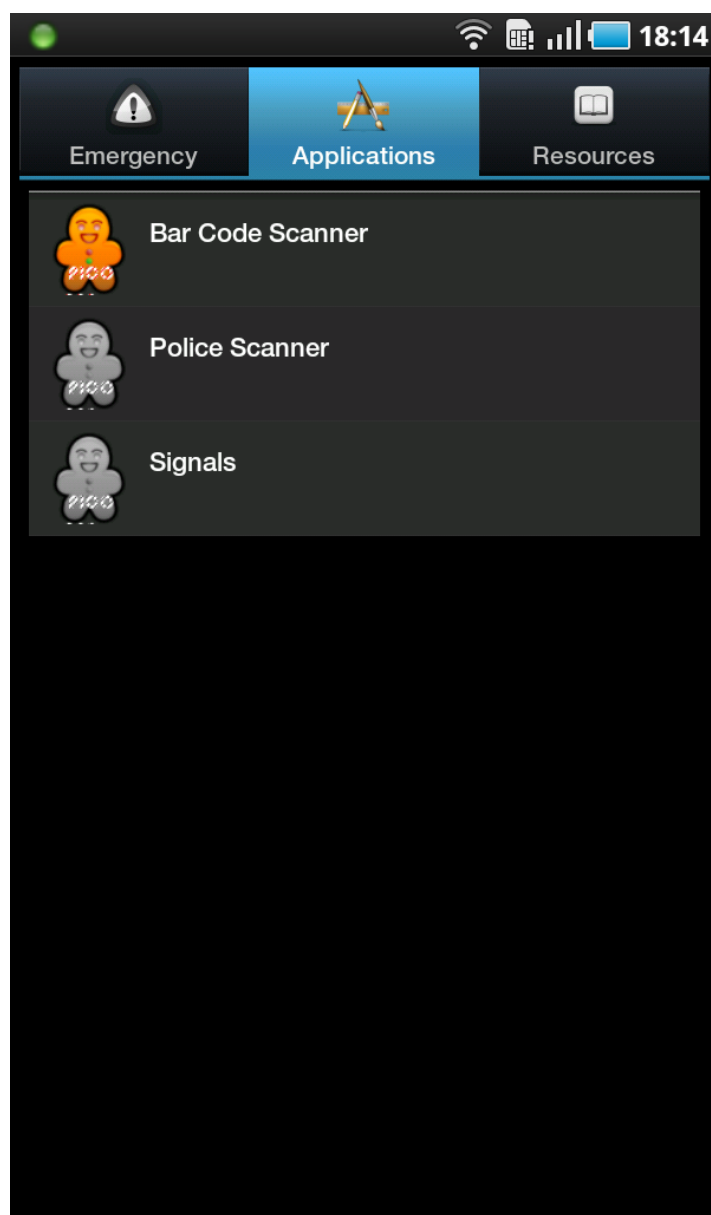


Figure 26 - QR Code scanner received

Once the information is read (for the prototype we used a QR CODE, but in the future the NFC could be an alternative), the client will send this information to the PICO Server. The PICO server will send this information over all users associated to the same emergency. This information could help the paramedic (blood group) or other user (call parents of the injured)

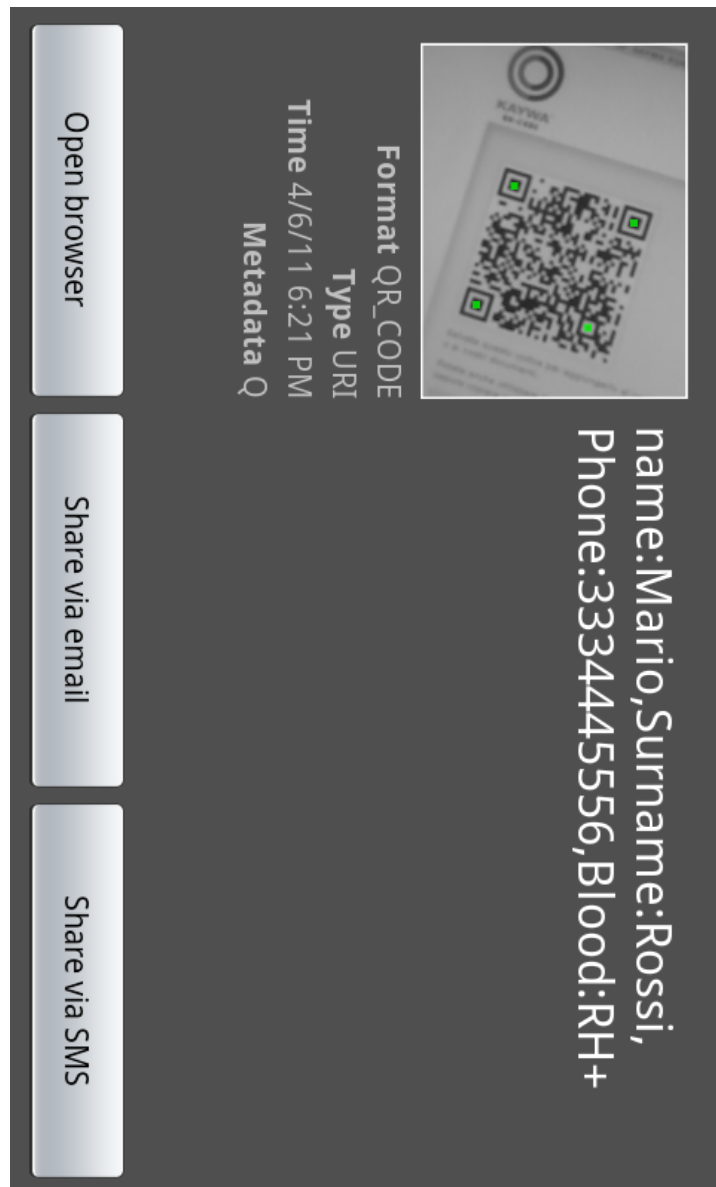


Figure 27 - Reading the bar code scanner

The users associated to the same emergency will receive all information obtained from the QR Code and decide what to do (call the family or blood group for paramedic)

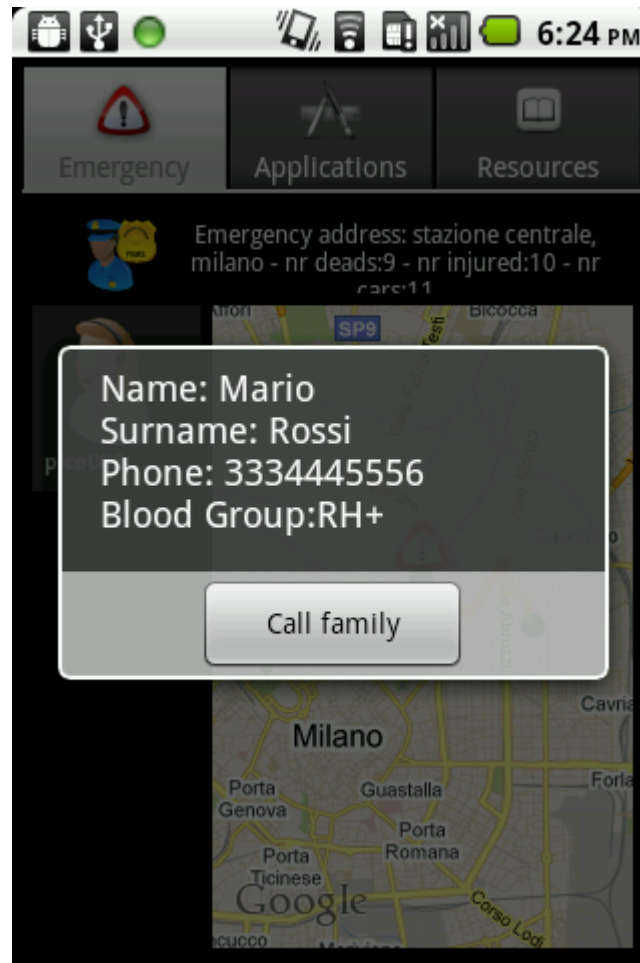


Figure 28 - Information received with a call number option

3.4 Alternative scenario

-
1. A car accident (A1) with three unconscious injured and 3 cars involved has just happened in the Zone 1 (Z1).
 2. A policeman (PSU1) reaches the Accident (A1).
 3. The main desktop of the policeman's device (PSCD1) offers the list of buddies and the position of the policeman in that moment. Among the list, the policeman selects the operator of the police station (OP1) and begins an audio call using a IMS Client.
 4. During the call, the operator (OP1) inserts via a web interface all initial data about the accident into PICO Server (S1) and share the application "First Emergency Call" (AP1) with him.
 5. In the notification area of the policeman's device the download status of the application (AP1) is shown, when it is downloaded, the policeman (PSU1) from the second desktop of the device (PSCD1) launches the first emergency call application (AP1) and starts scanning the QRCode of the first victim (V1)
 6. When Qr code is scanned, the application (AP1) shows a pop-up message with the Name and Lastname of the victim (V1) and calls automatically the first number of an important person (eg. Parents) using the IMS Client. During the call the policeman warns the situation and read the name to the person.
 7. At the same time, the application using the information embedded in the QRCode, sends the name and other important information related to the victim (V1) to the nearest PICO Server (PICO1) available (using LoST , future development*)

Moreover all Accident information such as name and current location are also sent to his important facebook contacts.**

8. PICO Server processes the information related to the accident (A1) and the information related to all users' contexts connected to the PICO Server. The reasoner of the PICO Server (PICO1) begins to analyze contexts and positions closer to the accident(A1). Reasoner will select the best PSCUser available according also to other parameters such as: Team, kind of user....
 9. An ambulance (PSU2) with two paramedics is near the Zone 1(Z1), PICO Server(PICO1) via the reasoner alerts with a message the paramedic's device (PSCD2) that contains all medical informations gained at 7) , using the embedded IMS Client, automatically connects with an audio call or video call (base on network condition, battery level), the paramedic(PSU2) with the policeman (PSU1). On the main desktop is shown a map with 2 IMS buddies, PSU1 and PSU2 and the best path to reach the Accident (A1) base on traffic conditions.
 10. On display of the PSCD2 a message with the availability of a diagnostic application (AP2) with some custom data related to the victim (V1) is shown. The paramedic (PSU2) is able to download and install it during the route to the accident (A1).
 11. The policeman starts from the contextual application list(second desktop) an application to find the nearest tow truck to the zone 1 (Z2) and sends a sms with all accident details (location, number of cars)
 12. A squad of firefighters is in the Zone 2 (Z2) attached to PICO Server 2 (PICO2). The squad is available and using LoST, the request is forwarded from PICO1 to PICO2.
 13. PICO Server starts the application that shows the best path to the FF and begins also a communication session between the Firefighter (PSCU3) and the policeman (PSCU1) based on network condition:
 - a. chat session (low traffic condition) (the first auto chat message contains all accident details)
 - b. Audio session (medium traffic condition)
 - c. Video session (high traffic condition)
-

-
14. The policeman now has 3+1 active buddies in his buddy list and based on session started on 13), decides to:
 - a. share a picture using a IMS session (IMS Client) to the FF cause a small fire was started after the accident. So the FF can see how big is the fire and position of the cars.
 - b.
 15. The reasoner waiting the ambulance, updates the second desktop of the Policeman (PUS1) device based on his contex (battery level):
 - a. With a First Aid application (in case of medium level battery). (second desktop)
 - b. With a PDF file with the aid instructions to download (third desktop).
 - i. If the device has low disk space, then a PDF lite version will be downloaded.
 - ii. If the device has high disk space, then a PDF full version will be downloaded.

Moreover the reasoner can activate/deactivate some sensors of the PSCD Device (GPS, Camera, wifi, screen light...)

16. A notification message on the PSCD1 device aletes the policeman that the contextual applicaiton list (second desktop) has been updated, so the policeman can switch the desktop, selects the First Aid Application (or PDF file from third desktop) and download/launch it.
 17. When the ambulance is arrived to the Accident (A1), the paramedic selects from the contextual application(second desktop) list the application to monitor the heart rate of the Victim
 18. TODO: PICO Answer machine. Automatic text answers via IMS chat
-

Index of Figures

| | |
|---|----|
| Figure 3 - Initial Pico | 12 |
| Figure 4 - Map with first police | 13 |
| Figure 5 - First Police Screen | 14 |
| Figure 6 - Opening an Emergency from device | 15 |
| Figure 7 - Feedback Emergency..... | 16 |
| Figure 8 - System auto association between the user and the emergency | 17 |
| Figure 9 - Showing the association on the map | 18 |
| Figure 10 - List of application | 19 |
| Figure 11 - Installation and confirmation | 20 |
| Figure 12 - Installation of an application..... | 21 |
| Figure 13 - Application installed available using a feedback..... | 22 |
| Figure 14 - Second Policeman screen..... | 23 |
| Figure 15 - First and second policeman on the map | 24 |
| Figure 16 - Pico server options..... | 25 |
| Figure 17 - Incoming call from second policeman to the operator | 26 |
| Figure 18 - In Call option | 27 |
| Figure 19 - Opening an emergency from desktop OPERATOR | 28 |
| Figure 20 - Crisis opened from operator | 29 |
| Figure 21 - Association between the second policeman and the second emergency | 30 |
| Figure 22 - Association on the map..... | 31 |
| Figure 23 - Fire fighter screen | 32 |
| Figure 25 - Option between users..... | 33 |

| | |
|--|----|
| Figure 26 - Sending an application between users..... | 34 |
| Figure 27 - QR Code scanner received | 36 |
| Figure 28 - Reading the bar code scanner..... | 38 |
| Figure 29 - Information received with a call number option | 39 |
