Applicazioni telematiche

Location Awareness Computing

Prof. MASSIMO FICCO

E-mail: ***ficco@unina.it***

Location Based Services (LBS)

Location awareness is a key factor for many advanced and adaptive nomadic/ubiquitous/mobile computing applications

Range of services

Information
Entertainment, news, weather, **traffic & navigation**, advertising, "find your friend"

Assistance
Personal or vehicle emergency, **roadside assistance**, alarm management

Asset tracking
Wireless remote management & control

B: Business
C: Consumer

M: Machine
P: Person

B2B
Fleet & freight tracking
…
M2M & P2P

B2C
Pushed info.
Discount, ad, special event
…
M2P

C2B
Pulled info.
Find Gas station, restaurant,
…
P2M

C2C
Pulled info.
Find friend, pets
…
P2P

Application Scenarios

Device initiated mode: the mobile device contacts its agent to determine its own location.

Network initiated mode: a network application starts the positioning procedure (it may ask mobile device's collaboration).

The considerate *wireless network architecture* includes the wide area cellular networks (e.g. UMTS and GPS/GPRS networks) and wireless data networks (i.e. WLAN, Bluetooth networks).

Interworking between 3G systems and WLANs

- ***The** interworking **between 3GPP/3GPP2 (3G) systems (e.g. UMTS) and wireless Local Area Networks (WLANs) pave the ground to new and more favourable services.***
- ***The** 3G–WLAN interworking **aims to extend 3G services and functionalities to the WLAN access environment, allowing to a 3G user to use a WLAN to access 3G based services.***

Business opportunity

- ***Several solutions have been proposed***
  - adopt one single location sensing technology
  - not fit all scenarios (indoor/outdoor)
- ***More Location Technologies***

UMTS and GPS/GPRS					
GPS/A-GPS	WiFi	Bluetooth	Radio Frequency	RFID	Ultrasonic

- ***Several technologies coexist on common user terminals***

New location-aware applications

New location aware systems are needed

System requirements

- **Architecture and protocol:**
  - to acquire and transfer positioning information between mobile devices and the applications;
  - to support multiple positioning systems for different scenarios (indoor and outdoor).
- **Programming abstractions (API)** that masks technology-dependent attributes, and it allows dynamic combination of location information from multiple source in a manner that is transparent to applications.
- **Data model** for location information.

Architecture models

- **Control plane** *architecture: the location delivery capability is an integral part of the overall network infrastructure. For instance, the location process of E911 emergency service in the United States involves all components among the wireless modem and emergency service provider.*

- **User plane** *architecture: the information is part of the wireless user data and is transported over user bearers such as the wireless packet data network or SMS.*

Secure User Plane Location (SUPL)

- *The Open Mobile Alliance (OMA) has recently finalized the Secure User Plane Location (SUPL) standard in order to cope with location-awareness services development [2].*
- *SUPL is an enabler that utilises existing standards to transfer assistance data and positioning information over a User Plane bearer, such as Internet Protocol (IP).*
- *SUPL defines the corresponding interfaces and location information exchanged between the SUPL network and SUPL enabled terminal (SET).*
- *It includes security functions (e.g., authentication, authorization), charging functions, roaming functions and privacy functions.*
- *The first SUPL release enabled multiple positioning technologies including Assisted Global Positioning System (A-GPS), cell ID, and enhanced cell ID.*
- *For SUPL release 2, the new concepts are introduced to allow additional positioning technologies (WLAN).*

Secure User Plane Location (SUPL)

The proposed location architecture is structured of several logical components:

- ***SUPL Enabler Terminal (SET): any device capable of communicating with a SUPL network.***
- ***Mobile Location Services (MLS): an application which requests and consumes the location information.***
  - ***The SUPL functionalities are used by MLS applications via a SUPL Agent.***
  - ***The MLS services can reside on the network as well as on the SET.***

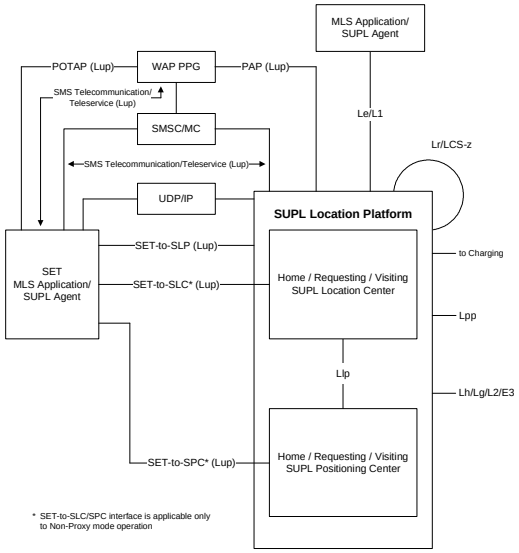
Secure User Plane Location (SUPL)

- ***SUPL Location Platform (SLP):*** is the entity responsible for estimating and delivering the SET position information.

It is composed of:

- a ***SUPL Location Center (SLC)***, that coordinates the operations in the network and interacts with the SET;
- a ***SUPL Positioning Center (SPC)***, that is responsible for all messages and procedures required for position determination of a SET.

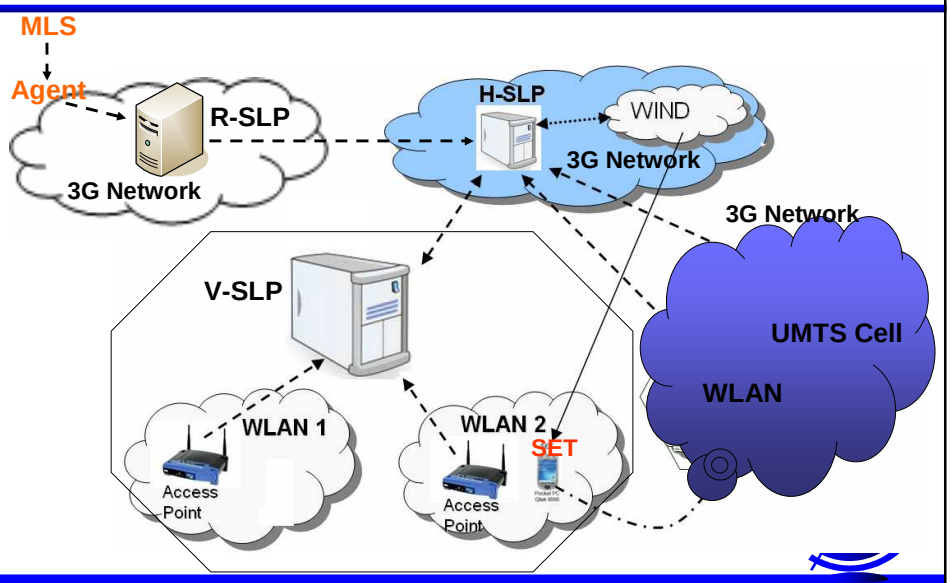
Secure User Plane Location (SUPL)



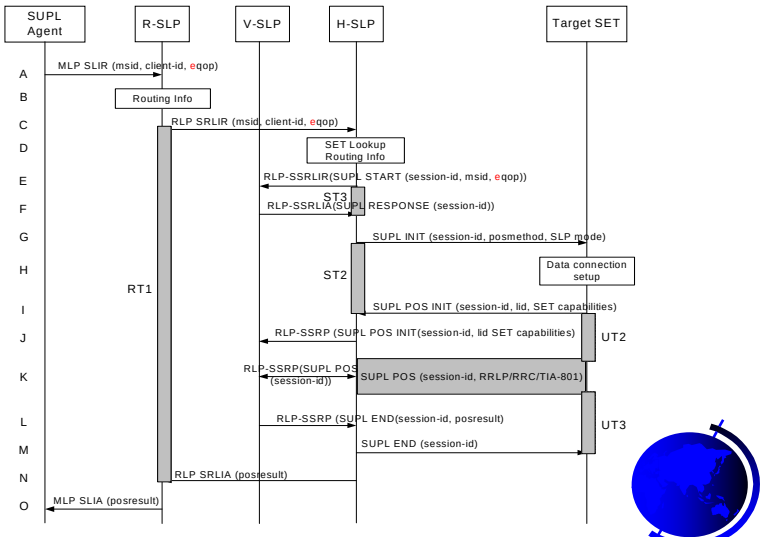
Secure User Plane Location (SUPL)

- ***When the SLP performs the role of the Home SLP (H-SLP), it contains the SET's personal information (the subscription, authentication and privacy related data).***
- ***When a SET leaves the service area of its H-SLP, a SUPL roaming may occur. In such case, in order to provide the SET position estimation the H-SPL can contact the SLP to which the SET is connected (Visited-SLP).***
- ***For network initiate services, if the SUPL Agent is within the H-SLP area, it can directly interacts with the H-SLP, otherwise it interacts with the Requester SLP (R-SLP) with which it is associated.***

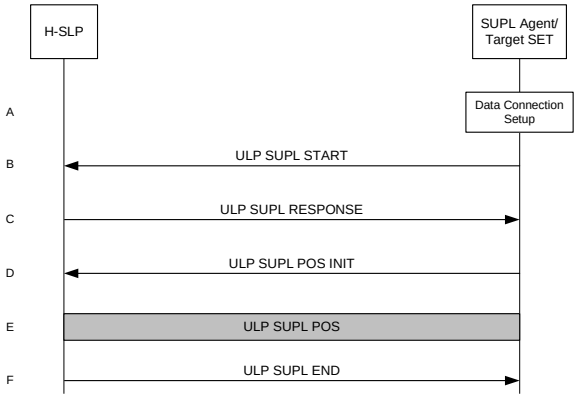
Secure User Plane Location (SUPL)



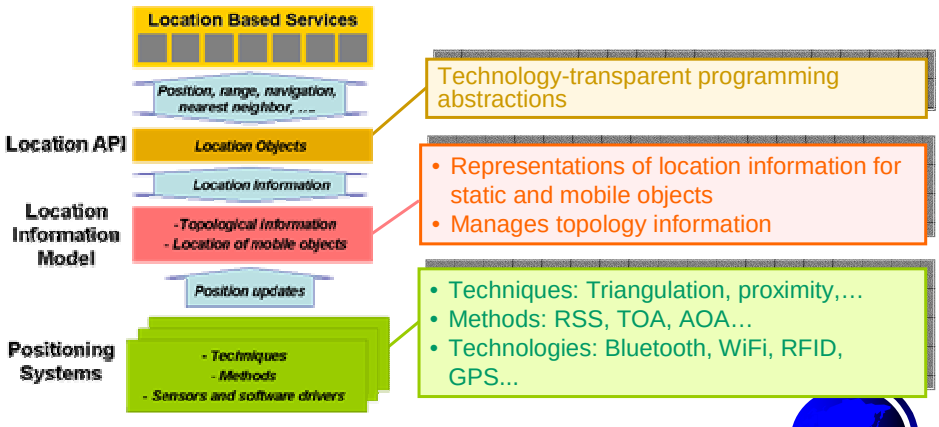
Network initiated scenario



Set initiated scenario



Location sensing architecture



The Location API

JSR-179 specifications ***define a Java 2 Micro Edition (J2ME) optional package to enable location-aware applications for Mobile Information Device Profile (MIDP) based devices.***

☞ ***Two main functionalities within JSR-179:***

- obtaining information about location and orientation of the mobile device
- accessing an on-device landmark database

Provide abstractions to the applications used from the architecture to choose the proper technology on behalf of the application, thus easing the development of applications that must function in both indoor and outdoor scenarios.

The Location API

☞ ***Each functionality exploits specific objects as information containers:***

☞ Location class: ***represents the standard set of basic device-location information;***

☞ Landmark class: ***represents known locations***

- Landmark objects have a name and may be placed either into a single category or into several categories. Each category is intended to group Landmark that are of similar type to the end user (e.g. restaurants, museums).
- Landmarks are stored into a persistent repository, called LandmarkStore.

Landmark objects

```
<?xml version="1.0" ?>
<Map>
        <Landmark>
          <name>Landmark1</name>
          <description>Stanza edificio</description>
          <latitude>45</latitude>
          <longitude>50</longitude>
          <altitude>0</altitude>
          <extension>Flat5</extension>
          <street>10 washington Street</street>
          <postal_code>12345</postal_code>
          <city>Palo Alto</city>
          <county>Santa Clara County</county>
          <state>California</state>
          <country>United States of America</country>
          <country_code>us</country_code>
          <district>distretto</district>
          <building_name>Edificio1</building_name>
          <building_floor>terzo</building_floor>
          <building_room>gialla</building_room>
          <building_zone>gialla</building_zone>
          <crossing1>crossing</crossing1>
        </Landmark>
```

The Location API

☞ ***The*** LocationProvider class ***is a module able to determine the location of the terminal by using existing location methods, including satellite based methods like GPS, and short-range positioning methods like Bluetooth positioning approach.***

- Each device can have several location providers installed, each related to a different positioning technique (e.g., GPS and RSS-based triangulation).

The Location API

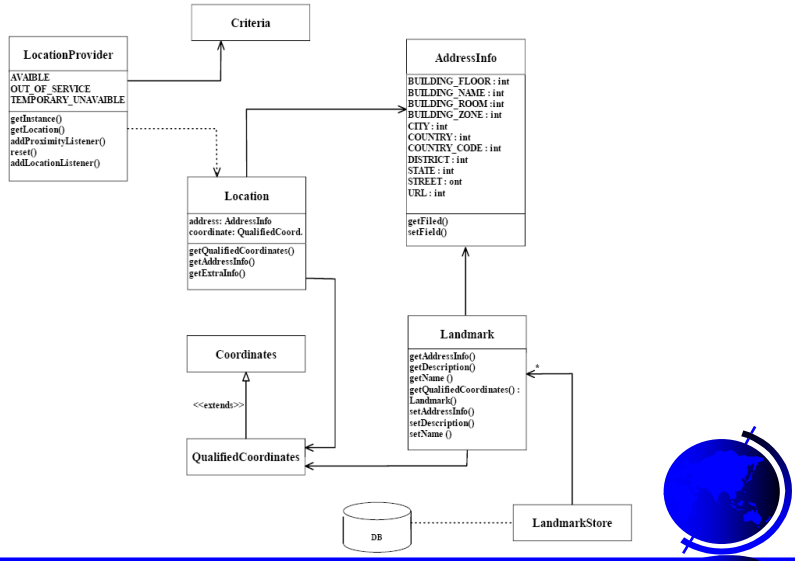
- ***The** Criteria class **allows to specify selection criteria to choose the most suitable LocationProvider.***

Criteria Field	*Default value*
Horizzontal accuracy	*NO_REQUIREMENT*
Vertical accuracy	*NO_REQUIREMENT*
Preferred response time	*NO_REQUIREMENT*
Power consumption	*NO_REQUIREMENT*
Cost allosew	*Tue (allowed to cost)*
Speed required	*False (not required)*
Altitude required	*False (not required)*
Address info required	*False (not required)*

The Location API

- *Upon the selection of a specific LocationProvider, the application can retrieve Location objects by means of either periodic updates or asynchronous queries. For periodic updates, the applications can use the LocationListener, that represents a listener that receives events associated with a particular LocationProvider.*

The Location API

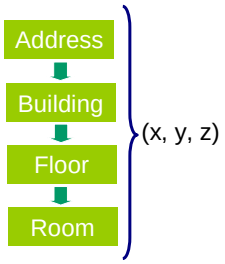


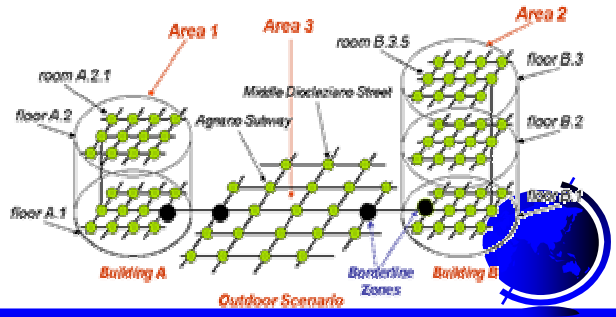
The location Information Model

- Geometric Geometric Location Model Location Model***: describe locations via geometric coordinates.***
- Symbolic Symbolic Location Model Location Model***: manage symbolic coordinates. These are available, for instance, via cell-IDs in cellular networks or wireless LANs, as well as by means of radio frequency tag (RFID), or infrared beacons.***
- Hybrid Hybrid Location Model Location Model

The location Information Model

- Zone = ***a certain area of the physical world that is of interest to end user***
- Hybrid information model
  - Symbolic and geometric coordinates
  - Interconnections and spatial containment relations between the Zones
- Area = ***group of homogeneous (e.g. GPS) adjacent*** Zones
- Borderline Zones = ***boundaries between different areas***
  - Transparent switch between positioning technologies





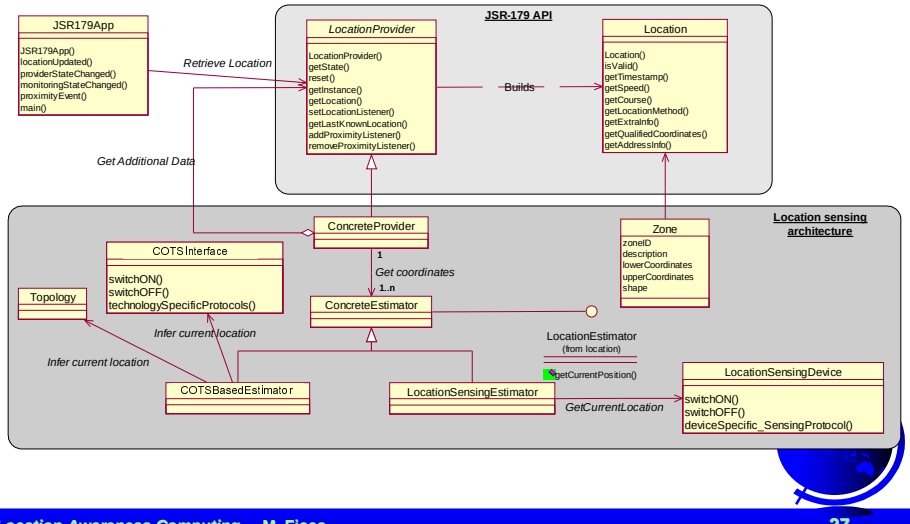
Mapping the location model onto the JSR-179 semantics

Each Landmark have a name that identifies the Zone to the end user (e.g. the Contemporary Art gallery within an exhibition), a set of coordinates, and an optional Address Info, which include definition of topological connection (include neighbours can be definite for each pair of directly connected locations, e.g. rooms which are connected by door) as well as the spatial containment relation (room is spatially contained in floor of a building).

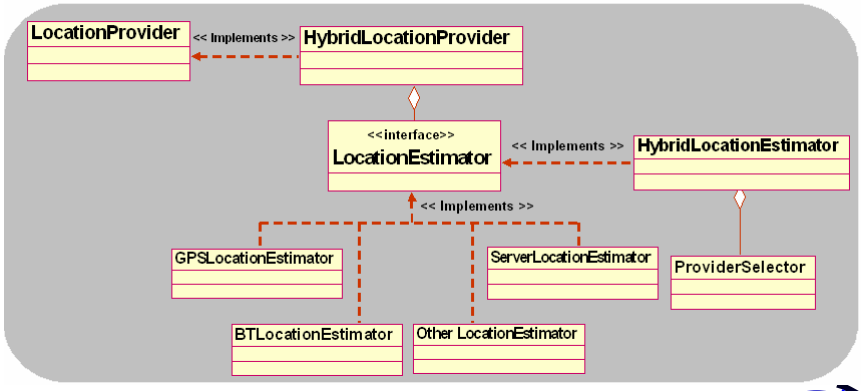
Name: *Room A.2.1*
Longitude: *null* **Latitude:** *null* **Altitude:** *null*
Neighbouring: *A.2.2 - A.2.5*
Contained: *A.2*
Contents: *null*
Description: *Room 1, of the second floor of CINI Laboratory*
Address: *Via Diocleziano 328, Naples (IT)*

AddressInfo

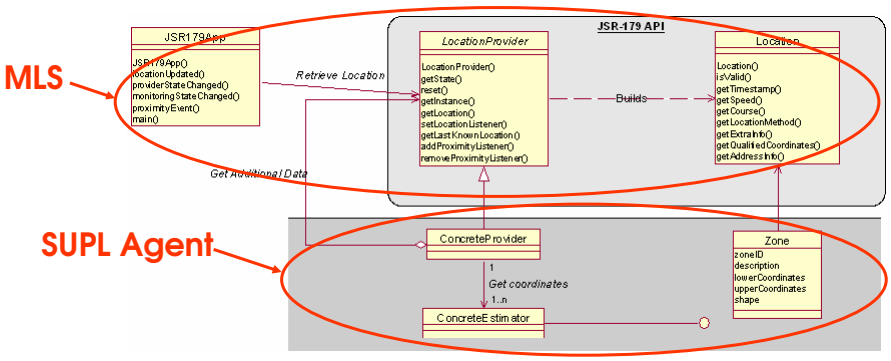
An JSR-179 implementation example



An JSR-179 implementation example



SUPL and the JSR-179



References

- [1] Patterson, C.A., Muntz, R.R., Pancake, C.M.: Challengesin Location-AwareComputing, in IEEE PervasiveComputing, 2(2), pp. 80-89. IEEE CS Press (2003).
- [2] The OMA Secure User Plane Location (SUPL) SUPL v.1, at: http://www.openmobilealliance.org/release_program/supl_v1_0.html
- [3] Java Community Process: Location API forJ2ME Specification1.0 Final Release(2003).
- [4] Becker, C., and Durr, F.: On location modelsforubiquitouscomputing, in Personal and UbiquitousComputing (2005), Springer-verlagLNCS, vol. 9(1), pp. 20-31, (2005).
- [5] Hightower, J., Borriello, G.: Location Sensing Techniques, in Technical report UW-CSE-01-07-01. University of Washington (2001).
- [6] Beresford, A., and Stajano, F.: Location Privacy in PervasiveComputing, in IEEE PervasiveComputing, 2(2), pp. 46-55. IEEE CS Press (2003).
- The OMA Secure User Plane Location (SUPL) SUPL v.1, at: http://www.openmobilealliance.org/release_program/supl_v1_0.html