



NOKIA



JavaOne

Bring Map and Navigation Capabilities to Your Location-Based Applications With JSR 293, Location API 2.0

Jaana Majakangas
Standardization Manager
Nokia Corporation

Michael Zhang
Senior Manager
SiRF Technology

<http://jcp.org/en/jsr/detail?id=293>

TS-5345

Goal

Learn to use map and navigation services in your applications and to exchange landmark information with others

Agenda

Background and scenarios for Java™
Specification Request (JSR) 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

Agenda

Background and scenarios for Java™ Specification Request (JSR) 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

Background

- Location-Based Services (LBS) is expected to be the fastest growing sector in the wireless data services
 - More than 85% of wireless subscribers are interested in one or more LBS apps*
 - 80% of respondents would leverage personal navigation and LBS content via their cell phones**
 - 25% of WCDMA handsets will have GPS by the end of 2008***
 - GPS-enabled LBS subscribers will rise from 12 million in 2006 to 315 million in 2011***

Application Development Challenges

- Consumer privacy concerns
- Slow implementation of location technologies
- Hard to develop applications
 - Access to location: different methods, different vendors, different platforms
 - Access to mobile mapping: different map providers, different APIs, device-based solutions vs. network-based solutions
 - Access to advanced LBS capabilities like navigation: different map data providers, different APIs, device-based solutions vs. network-based solutions

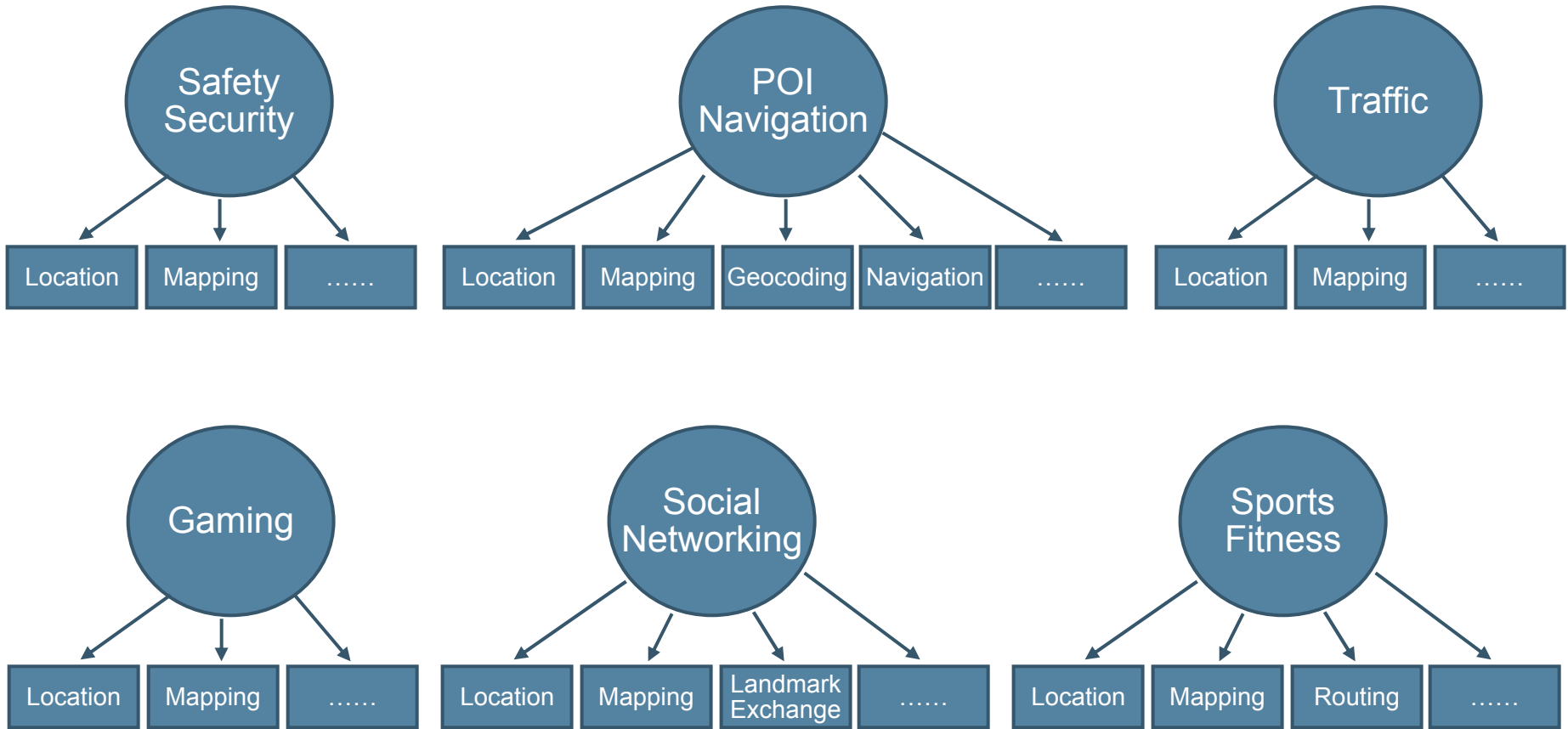
Solutions

- Common and standardized APIs on top of heterogeneous environments
- JSR 179: standardizing location retrieval on the Java Platform, Micro Edition (Java ME platform)
 - Synchronous and asynchronous location retrieval
 - Landmarks
- JSR 293: standardizing access to location-based services from Java ME platform
 - JSR 179 enhancements: criteria, landmark store, proximity listener
 - Mapping
 - Navigation
 - Geocoding and reverse geocoding
 - Landmark exchange

Services in JSR 293

- Geocoding services
 - Geocoding
 - Reverse geocoding
- Map services
 - Display maps with points or landmarks
 - Select items from map
- Navigation services
 - Navigation through a specified route
 - Creating routes from applications

Supported LBS Application Examples



Sample Use Cases for JSR 293

- Route planner
 - Make route of existing landmarks and show it on map
- Restaurant locator
 - Show different types of restaurants of the given area on the map
- Service search
 - Search service based on coordinates or address and navigate to the service
- Hiking trail service
 - Get interesting hiking trail from the web and show it on map

Agenda

Background and scenarios for JSR 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

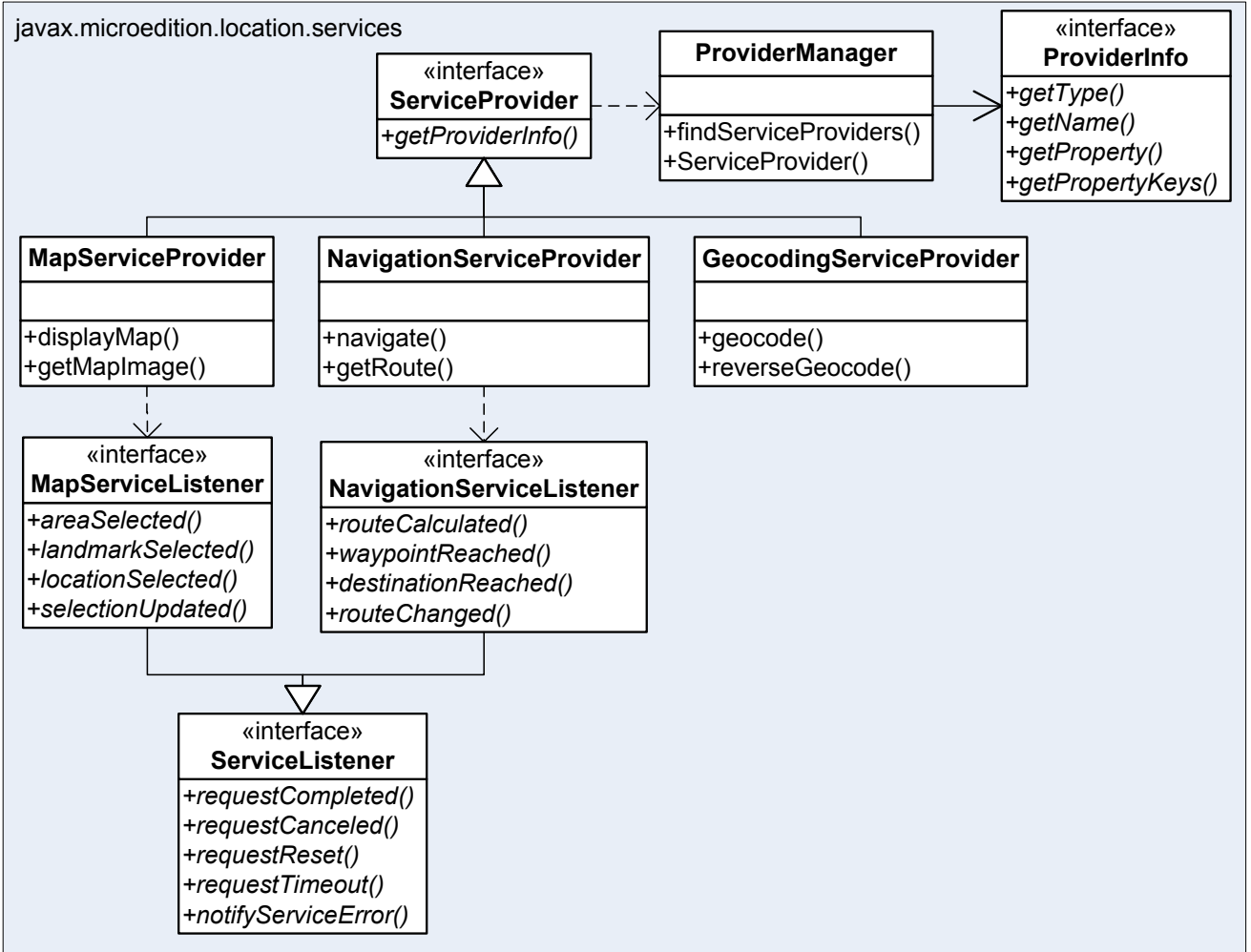
API Design Details

- Defines separate `javax.microedition.location.services` package for location related services
- Provides mechanism to discover and select different service providers
- Allows service requests in two modes
 - Service provider takes control and handles all communication with the user
 - The application is in control of the service requests and handles the communication with the user

API Design Details

javax.microedition.location

javax.microedition.location.services



Agenda

Background and scenarios for JSR 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

Geocoding Service Provider

- Provides mechanism to request following services
 - Geocoding (street address to coordinates)
 - Reverse geocoding (coordinates to street address)
- Methods are synchronous
- Applications control the dialogs shown to the user by the geocoding service provider

Geocoding Service Example

```
// obtaining a service first
GeocodingServiceProvider service =
    (GeocodingServiceProvider) ProviderManager.connectToServiceProvider(null,
                                                                    ProviderManager.GEOCODING);

Enumeration results;
Landmark myLandmark;
try {
    Coordinates myCoord = new Coordinates(37.37497, -121.9132, 0.0);
    results = service.reverseGeocode(myCoord, false);
    if(results.hasMoreElements()) {
        // pick the first result
        myLandmark = (Landmark) results.nextElement();
    }
} catch (InterruptedException e) {
    // Handle exception
} catch (ServiceException e) {
    // Handle exception
}

// Use the landmark
AddressInfo myAddress = myLandmark.getAddressInfo();
.....
```


Agenda

Background and scenarios for JSR 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

Map Service Provider

- Applications can request map service provider to:
 - Show a map with selected items on it
 - Show a map and allow the user to select items from the map
- Items on map
 - An area
 - A set of landmarks
 - A location
 - A route
 - Any combination of these
- Application can request map image with selected items from the map service provider



Map Service Example 1

```
Class MyMIDlet implements MapServiceListener {
...
    // obtain a map service first
    MapServiceProvider mapProvider =
(MapServiceProvider) ProviderManager.connectToServiceProvider(null
, ProviderManager.MAP);
    // display a location on the map
    try {
        Coordinates myCoord = new Coordinates(37.37497,
                                                -121.9132, 0.0);
        // request map service provider to show a map
        mapProvider.displayMap(null, null, myCoord, null,
            MapContainer.MAP_TYPE_REGULAR, 0, -1, false,
            false, this);
    } catch (Exception e) {
        // Handle exception
    }
...
}
```

Map Service Example 2

```
Class MyMIDlet implements MapServiceListener {
...
    // obtain a map service first
    MapServiceProvider mapProvider =
(MapServiceProvider) ProviderManager.connectToServiceProvider(null
, ProviderManager.MAP);
    // request map image containing the location
    try {
        Coordinates myCoord = new Coordinates(37.37497,
                                             -121.9132, 0.0);
        // request a map from the map service provider
        MapContainer map = mapProvider.getMapContainer(null, null,
            myCoord, null, 100, 120, MapContainer.MAP_TYPE_REGULAR, 0);
        // Render the image to the UI
        map.renderMap(graphics, true);
    } catch (Exception e) {
        // Handle exception
    }
...
}
```

Agenda

Background and scenarios for JSR 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

Navigation Service Provider

- Provides a mechanism to navigate through a specified route
- Application may set preferences for navigation
- Two alternatives
 - Navigation service provider controls the UI
 - Route object is returned to the application and it handles the UI itself



Navigation Service Example 1

```
public void navigate(Coordinates start, Coordinates end) {  
    // obtain a navigation service first  
    NavigationServiceProvider service =  
        (NavigationServiceProvider) ProviderManager.connectToServicePr  
        ovider(null, ProviderManager.NAVIGATION);  
  
    // navigate  
    try {  
        // Request navigation service provider to do the  
        // navigation and control the UI  
        service.navigate(route, null, null, this);  
    } catch (Exception e) {  
        System.out.println("Navigation service: " + e);  
    }  
}
```

Navigation Service Example 2

```
public void navigate(Coordinates start, Coordinates end) {
    // obtain a navigation service first
    NavigationServiceProvider service =
        (NavigationServiceProvider) ProviderManager.connectToServicePr
        ovider(null, ProviderManager.NAVIGATION);

    // navigate
    try {
        // Request navigation service provider to do the
        // navigation and control the UI
        service.getRoute(route, preferences);
    } catch (Exception e) {
        System.out.println("Navigation service: " + e);
    }
}
```


Agenda

Background and scenarios for JSR 293

API design details

Geocoding services

Map services

Navigation services

Landmark exchange

Landmark Exchange

- Enables applications to exchange landmark information with other devices
- JSR 293 defines an exchange format that must be supported
 - Other formats may also be supported
- Defines also a set of global landmark categories
 - Must be present in all JSR 293 compliant devices
 - Category names are localized

Summary

- Adds improvements to JSR 179
 - Criteria improvements, easier category handling, proximity exit listener
- Enables use of geocoding, map and navigation services from Java ME platform applications
- Enables importing and exporting landmarks using landmark exchange format
 - Defines a set of global landmark categories

For More Information

- JSR 179 Location API
<http://jcp.org/en/jsr/detail?id=179>
- JSR 293 Location API 2.0
<http://jcp.org/en/jsr/detail?id=293>



Q&A





NOKIA



JavaOne

Bring Map and Navigation Capabilities to Your Location-Based Applications With JSR 293, Location API 2.0

Jaana Majakangas
Standardization Manager
Nokia Corporation

Michael Zhang
Senior Manager
SiRF Technology

<http://jcp.org/en/jsr/detail?id=293>

TS-5345