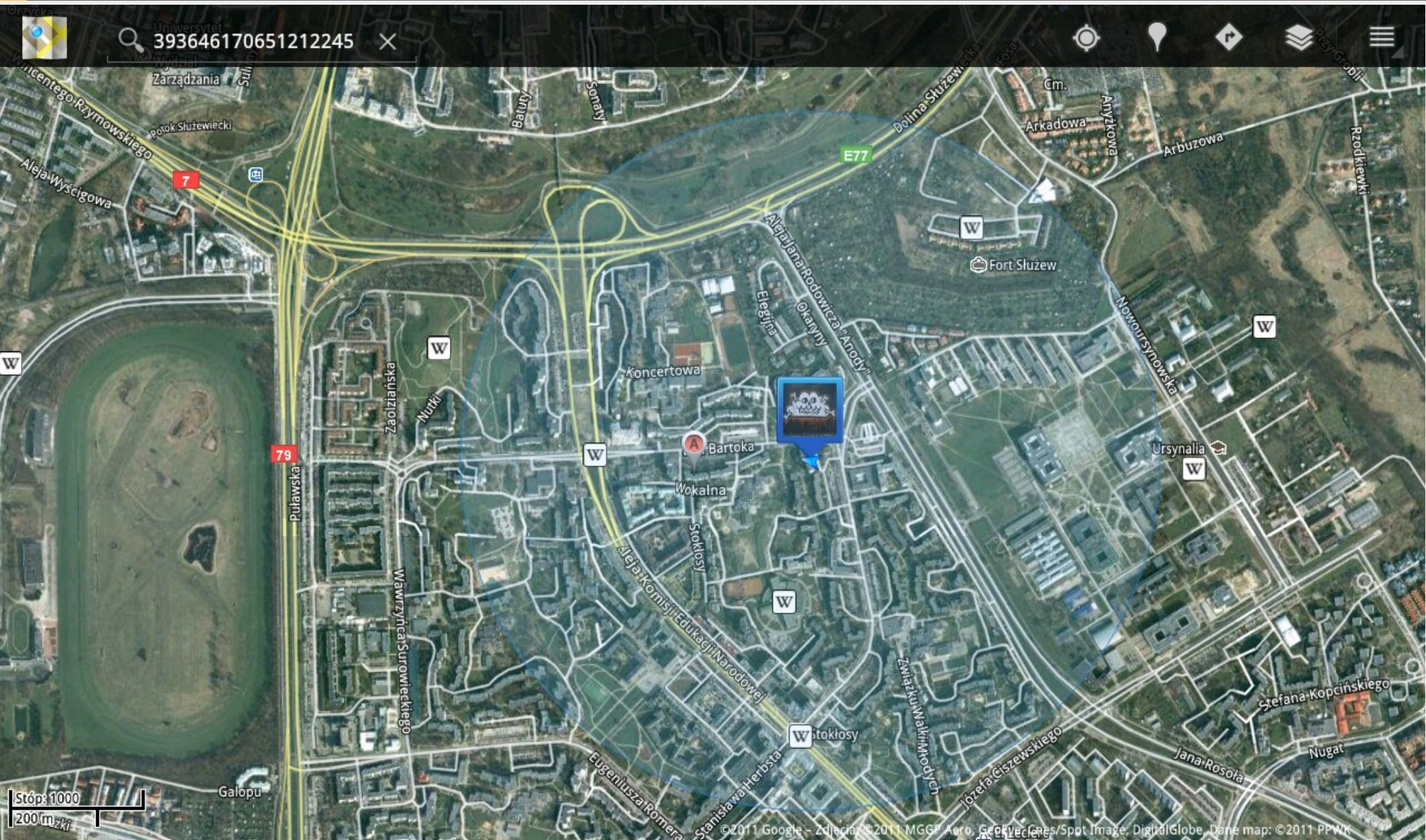


# Geolokalizacja odc. 2



# Coarse location - dokładność



# Dokładność lokalizacji

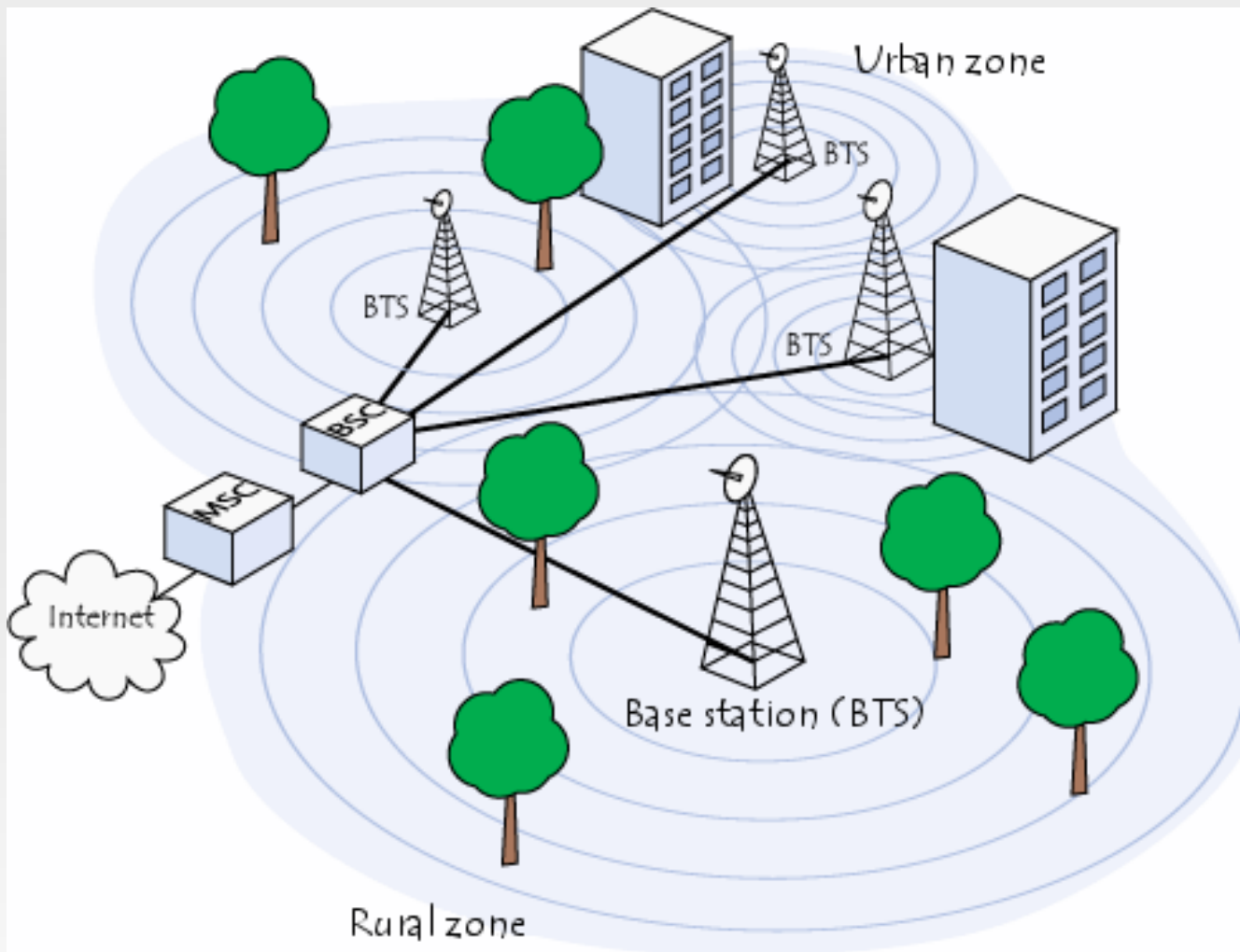
## Usługi:

- `android.permission.ACCESS_COARSE_LOCATION` – używana przez GSM lub WiFi. Niezbyt dokładna

w celu aktywacji w pliku konfiguracyjnym `AndroidManifest.xml` należy dopisać element `<uses-permission>` dla każdego typu usługi

```
<uses-permission  
android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

# Sieć GSM – schemat uproszczony



BTS - Stacja przekaźnikowa, stacja bazowa, (ang. Base Transceiver Station)

BSC - Kontroler Stacji Bazowej (ang. Base Station Controller)

MSC – cyfrowa centrala telefoniczna (ang. Mobile Switching Centre)

# MCC – Mobile Country code

- Mobile country code - kod krajowy sieci mobilnej. Występuje w użyciu z MNC (kodem sieci mobilnej). Identyfikuje operatora (unikalnie w skali globalnej) w sieciach mobilnych
- MNC (ang. Mobile Network Code) – unikatowy w obrębie danego kraju numer, identyfikujący sieć telefonii bezprzewodowej.

<http://pl.wikipedia.org/wiki/MNC>

[http://pl.wikipedia.org/wiki/Mobile\\_Country\\_Code](http://pl.wikipedia.org/wiki/Mobile_Country_Code)

# Kody MCC MNC - Polska

MCC	MNC	Brand	Operator
260	1	Plus	Polkomtel S.A.
260	2	T-Mobile	Polska Telefonia Cyfrowa Sp. z o.o.
260	3	Orange	Polska Telefonia Komórkowa Centertel Sp. z o.o.
260	4		CenterNet S.A.
260	5		Polska Telefonia Komórkowa Centertel Sp. z o.o.
260	6	Play	P4 Sp. z o.o.
260	7	Netia	Netia S.A.
260	8		E-Telko Sp. z o.o.
260	9	Lycamobile	Lycamobile Sp. z o.o.
260	10	Sferia	Sferia S.A.
260	11	Nordisk Polska	Nordisk Polska Sp. z o.o.
260	12	Cyfrowy Polsat	Cyfrowy Polsat S.A.
260	13	Sferia	Sferia S.A.
260	14	Sferia	Sferia S.A.
260	15	CenterNet	CenterNet S.A.

# LAC – Location Area Code

LAC (ang. Location Area Code) – kod przypisywany do Location Area (LA), obszarów na które jest podzielona sieć radiowa używana w sieciach telefonii komórkowej. LAC zapisywany jest za pomocą dwóch bajtów i jest unikatowy na poziomie sieci komórkowej (dwie LA w danej sieci na pewno będą mieć różne LAC).

Każde Location Area zawiera zazwyczaj kilkadziesiąt lub kilkaset komórek (ang. cell) rozróżnianych za pomocą numeru CID.

# CID – Cell ID

Cell Identifier (C-Id, C-ID[1], Cell Id, CID) – parametr przypisywany przez operatora telefonii komórkowej poszczególnym komórkom (ang. cells) zdefiniowanym na obszarze należącej do niego sieci.

Parametr ten musi być unikatowy wśród komórek (ang. cells) zdefiniowanych na obszarze zarządzanym przez BSC (GSM) lub RNC (UMTS)[3]. Używane jest także pojęcie Global Cell Id (GCI). GCI zapisywane jest w formacie MCC + MNC + LAC + Cell Identifier. Tak zdefiniowana nazwa komórki jednoznacznie opisuje ją wśród wszystkich komórek we wszystkich sieciach mobilnych.



# Telephony Manager – przykład

```
TelephonyManager tm= (TelephonyManager)  
getSystemService(Context.TELEPHONY_SERVICE);
```

```
GsmCellLocation cl = (GsmCellLocation) tm.getCellLocation();
```

```
if(cl!=null){
```

```
    Log.i("CellLocation"+i, "objectx "+cl); // [11,797737,-1]
```

```
    Log.i("CellLocation"+i, "getCid "+cl.getCid()); // 797737
```

```
    Log.i("CellLocation"+i, "getLac "+cl.getLac()); // 11
```

```
}
```

## Usage

```
http://ayauto.in/ayapi/process.php?token=your_token&mcc=404&mnc=49&cid=1161&lac=31
```

## Test it out

First, request a trial token [here](#). Then go right ahead and fill this up!

*Fields with \* are required.*

Token \*

Mobile Country Code \*

Mobile Network Code \*

Location Area Code\*

Cell Tower ID\*

<http://ayauto.in>

```
{"status":"ok","balance":18,"lat":52.23062,"lon":21.00191,"cid":"797747","lac":"11","mnc":"06","mcc":"260","src":"0","time":0.015,"sMode":"0"}
```



# Sposób 2: Geocoder

- `List<Address> getFromLocation(double latitude, double longitude, int maxResults)`

- `List<Address> getFromLocationName(String locationName, int maxResults, double lowerLeftLatitude, double lowerLeftLongitude, double upperRightLatitude, double upperRightLongitude)`

- `List<Address> getFromLocationName(String locationName, int maxResults)`

# Sposób 2: Geocoder

```
String addr = null;
Geocoder gcd = new Geocoder(getApplicationContext(),
    Locale.getDefault());
List<Address> addresses;
try {
    addresses = gcd.getFromLocation(l.getLatitude(),
        l.getLongitude(), 1);
    if (addresses.size() > 0)
        addr = addresses.toString();
} catch (IOException e) {
    e.printStackTrace();
}
```

Warszawa

```
[Address[addressLines=[0:"Złota 44/46",1:"Warszawa",2:"Polska"],feature=44/46,admin=mazowieckie,
sub-admin=Warszawa,locality=Warszawa,thoroughfare=Złota,postalCode=null,countryCode=PL,countryNa
me=Polska,hasLatitude=true,latitude=52.2310818,hasLongitude=true,longitude=21.0030065,phone=null
,url=null,extras=null]]
```