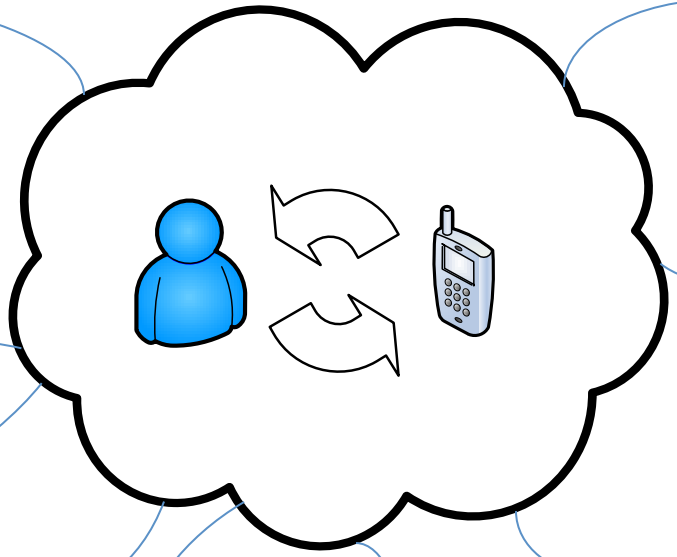

LOCATION
AND MAPS

Outline

- Location & Location-Based Apps
- Location API (android.location)
- Fused Location Service
- Maps
- Geocoding and Geofencing
- Tools & Tricks

LOCATION

Context-Awareness



User
Mobility
Physical activity
Task, plans intentions
Physiological
heart rate, respiration

Physical Environment
Audio
Light
Pressure
Temperature

Social
People
Setting

Device
Orientation, Acceleration
Screen
I/O modalities
Battery

Networks
Availability
Bandwidth
QoS

Location
Absolute
Relative

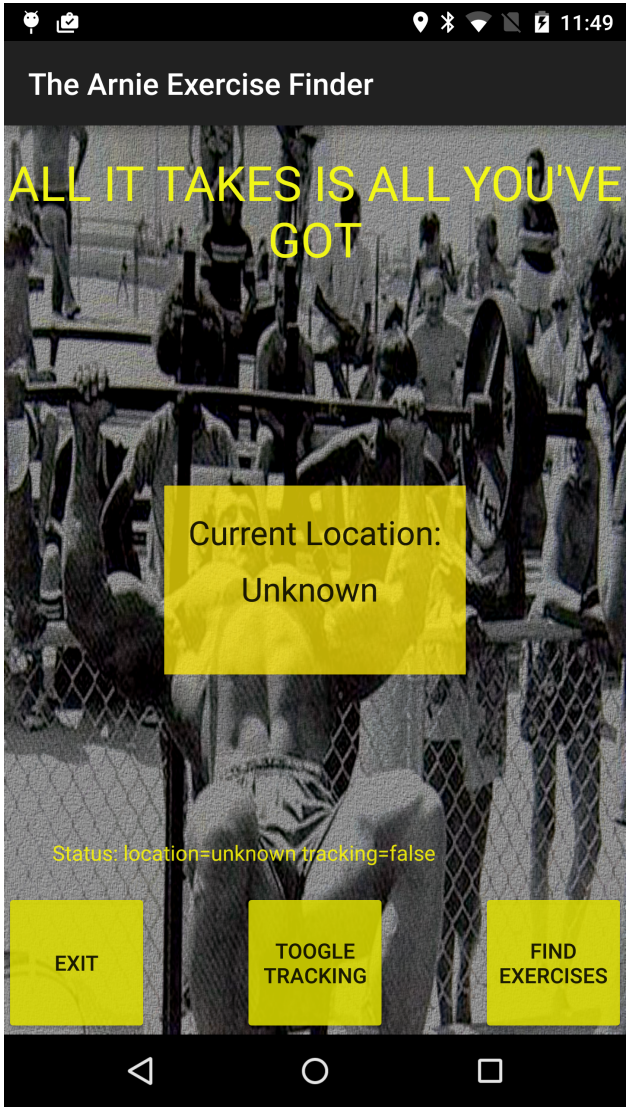
Time

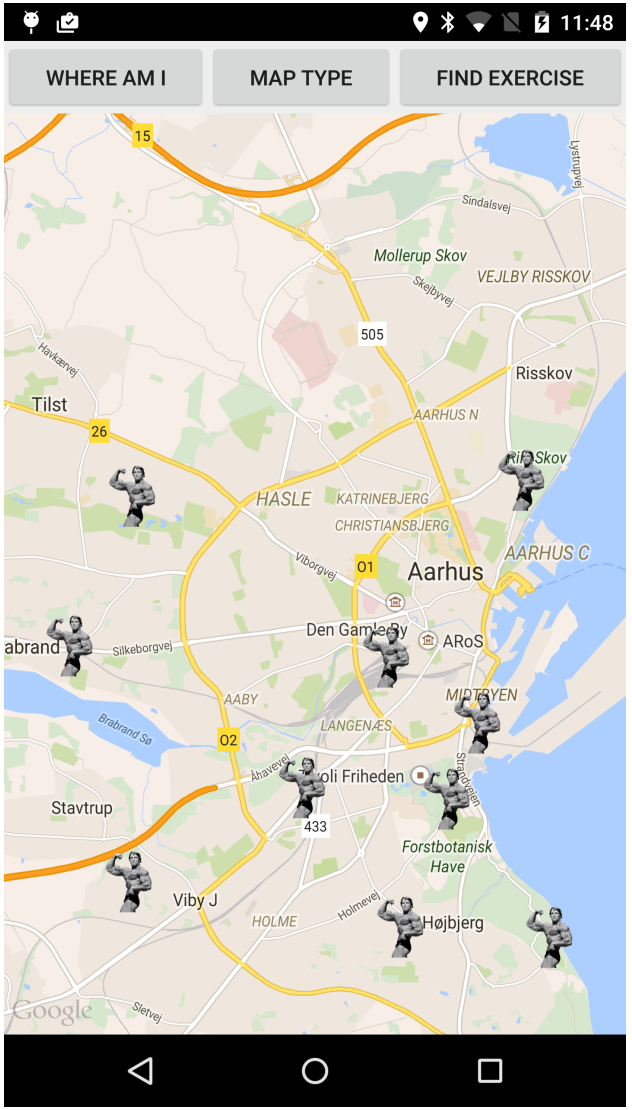
History

Computational resources
CPU, memory, storage
External resources/services

Location-Awareness

- Location Based Apps and Services


11:49
The Arnie Exercise Finder
ALL IT TAKES IS ALL YOU'VE GOT
Current Location:
Unknown
Status: location=unknown tracking=false
EXIT
TOOGLE TRACKING
FIND EXERCISES

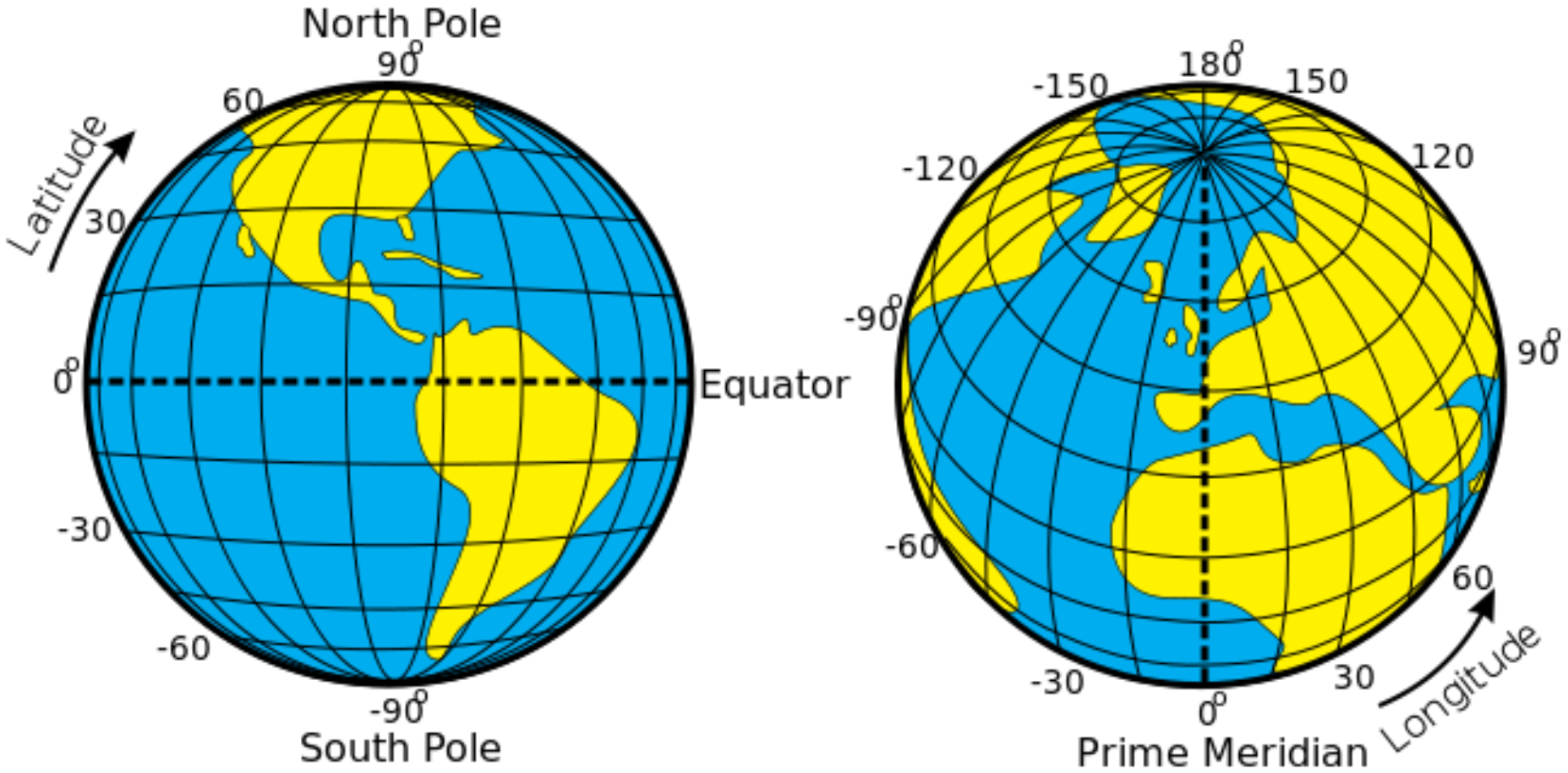

11:48
WHERE AM I
MAP TYPE
FIND EXERCISE
Tilst
Risskov
Aarhus
Stavtrup
Viby J
Højbjerg
Forstbotanisk Have
Mollerup Skov
VEJLBY RISSKOV
HASLE
KATRINEBJERG
CHRISTIANSBJERG
AARHUS C
AARHUS N
Den Gamle By
ARoS
AABY
LANGENÆS
MIDTBYEN
HOLME
Google

Location Technologies

- GPS
- Wifi
- Cell towers
- Other
  - External sensors
  - Beacons
  - Computer vision
  - …

https://developer.android.com/google/play-services/location.html

Latitude and Longitude



Source: Wikipedia - http://en.wikipedia.org/wiki/File:Latitude_and_Longitude_of_the_Earth.svg

Using Location

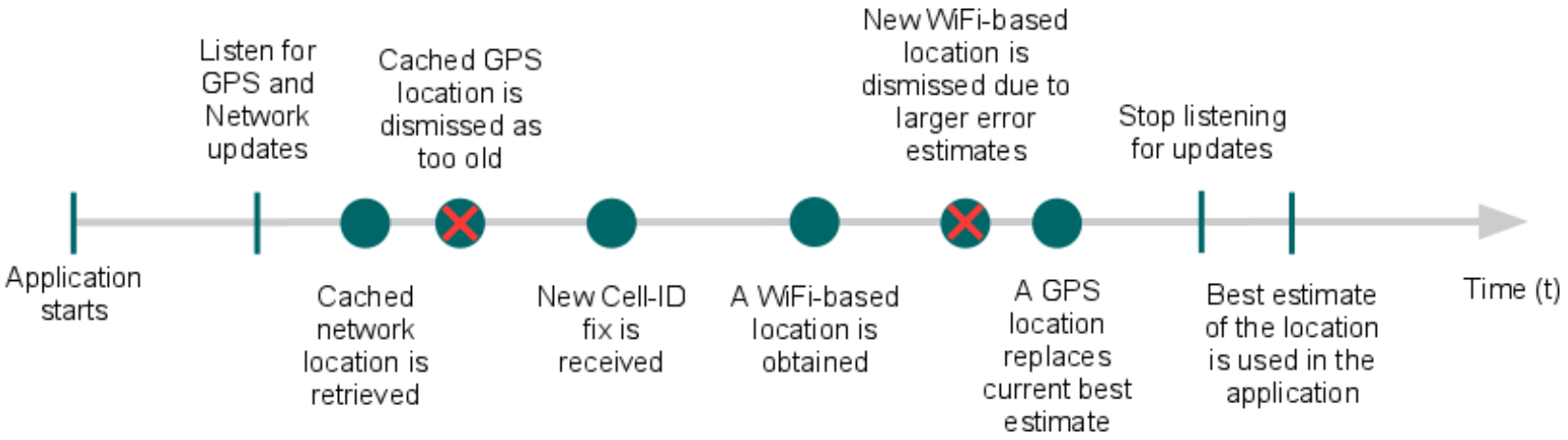
- Where the user is now
  - Current/last known location
- Where is the user going
  - Tracking, Location change updates
- When the user goes somewhere specific
  - Proximity alert, geofencing

Challenges of Location

- “Multitude of location sources
  - GPS, Cell-ID, and Wi-Fi can each provide a clue to users location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency.
- User movement
  - Because the user location changes, you must account for movement by re-estimating user location every so often.
- Varying accuracy
  - Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source”

http://developer.android.com/guide/topics/location/strategies.html

Location / Time



https://developer.android.com/guide/topics/location/strategies.html

Costs and Tradeoffs

- Battery life
- Network usage (especially when roaming)
- Accuracy
- Speed

Location-Aware Apps

- Two core ways of implementing
  - Location API (android.location)
  - Google Play services: Location APIs
- Many strategies
  - Should be tailored to the user experience of the app
  - Should be optimized (battery!)

ANDROID.LOCATION

Permissions

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

Location Manager

- Get `LocationManager`
  - `getSystemService(Context.LOCATION_SERVICE)`
- Check for available location providers
  - `LocationManager.GPS_PROVIDER`
  - `LocationManager.NETWORK_PROVIDER`
- Set up with desired criteria
  - Frequency (minimum time between notifications)
  - Distance (minimum change in distance)
  - `Criteria`
- Register callback `LocationListener`
  - `requestLocationUpdates(…)`
  - `removeUpdates(…)`

The `LocationListener` Interface

- Location Changes
  - `onLocationChanged(Location location)`
- Providers availability (by user)
  - `onProviderEnabled(String provider)`
  - `onProviderDisabled(String provider)`
- Status changes on providers
  - `onStatusChanged(String provider, int status, Bundle extras)`

`Location` object

- Can have hold the following data
  - Latitude
  - Longitude
  - Altitude
  - Timestamp
  - Bearing
  - Speed
  - Provider

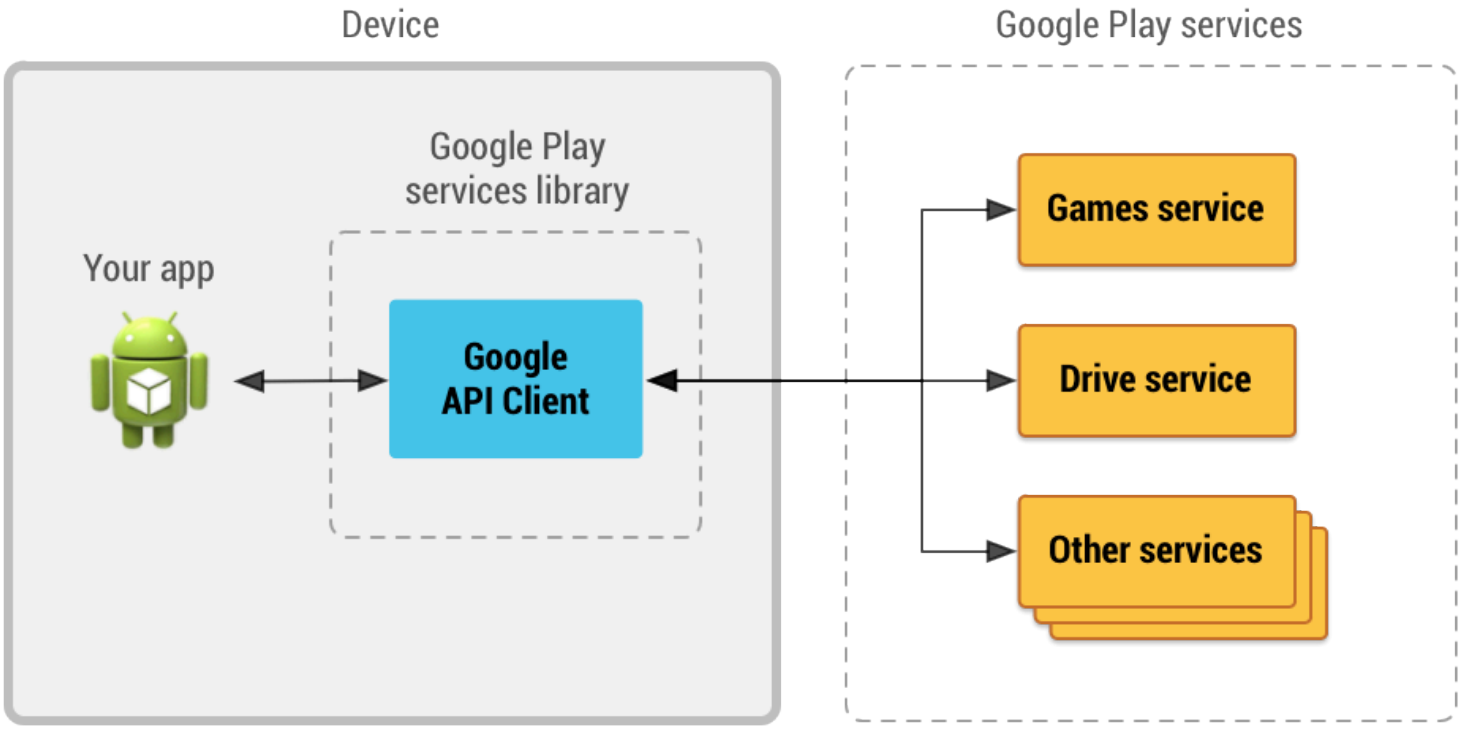
Location in Action

FUSED LOCATION PROVIDER

Fused Location Provider

- Complete service available through Google Play services
- Google recommends using this
  - Handles a lot of the problems with handling different location providers
  - Optimize battery use

Google API Client



https://developer.android.com/google/auth/api-client.html

Getting Google API Client

```
GoogleApiClient client =

new GoogleApiClient.Builder(context)
            .addApi(LocationServices.API)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .build()
```

https://developer.android.com/reference/com/google/android/gms/location/FusedLocationProviderApi.html

`GoogleApiClient` Callbacks

- Connect/disconnect for use
  - Call `connect()` in `onStart()`
  - Call `disconnect()` in `onStop()`
- Implement and handle callbacks
  - `GoogleApiClient.ConnectionCallbacks`
  - `GoogleApiClient.OnConnectionFailedListener`

Fused Location Provider

- No active tracking
  - `getLastLocation(GoogleApiClient)`
- Tracking in foreground
  - `requestLocationUpdates(GoogleApiClient, LocationRequest, LocationListener)`
- Tracking in background
  - `requestLocationUpdates(GoogleApiClient, LocationRequest, LocationListener, PendingIntent)`

The `LocationRequest`

- Used to specify criteria for location updates QoS (Quality of Service)
- Desired interval
  - `setInterval(…)`
- No faster than this interval
  - `setFastsInterval(…)`
- Priority (affects battery)
  - `setPriority(…)`

Priority levels and accuracy

Constants		
int	PRIORITY_BALANCED_POWER_ACCURACY	Used with `setPriority(int)` to request "block" level accuracy.
int	PRIORITY_HIGH_ACCURACY	Used with `setPriority(int)` to request the most accurate locations available.
int	PRIORITY_LOW_POWER	Used with `setPriority(int)` to request "city" level accuracy.
int	PRIORITY_NO_POWER	Used with `setPriority(int)` to request the best accuracy possible with zero additional power consumption.

https://developer.android.com/reference/com/google/android/gms/location/LocationRequest.html

LocationListener Interface

- Similar to `android.location`, this is where you get notifications of location changes
- Implement callback to act on changes
  - `onLocationChanged(Location location)`

MAPS

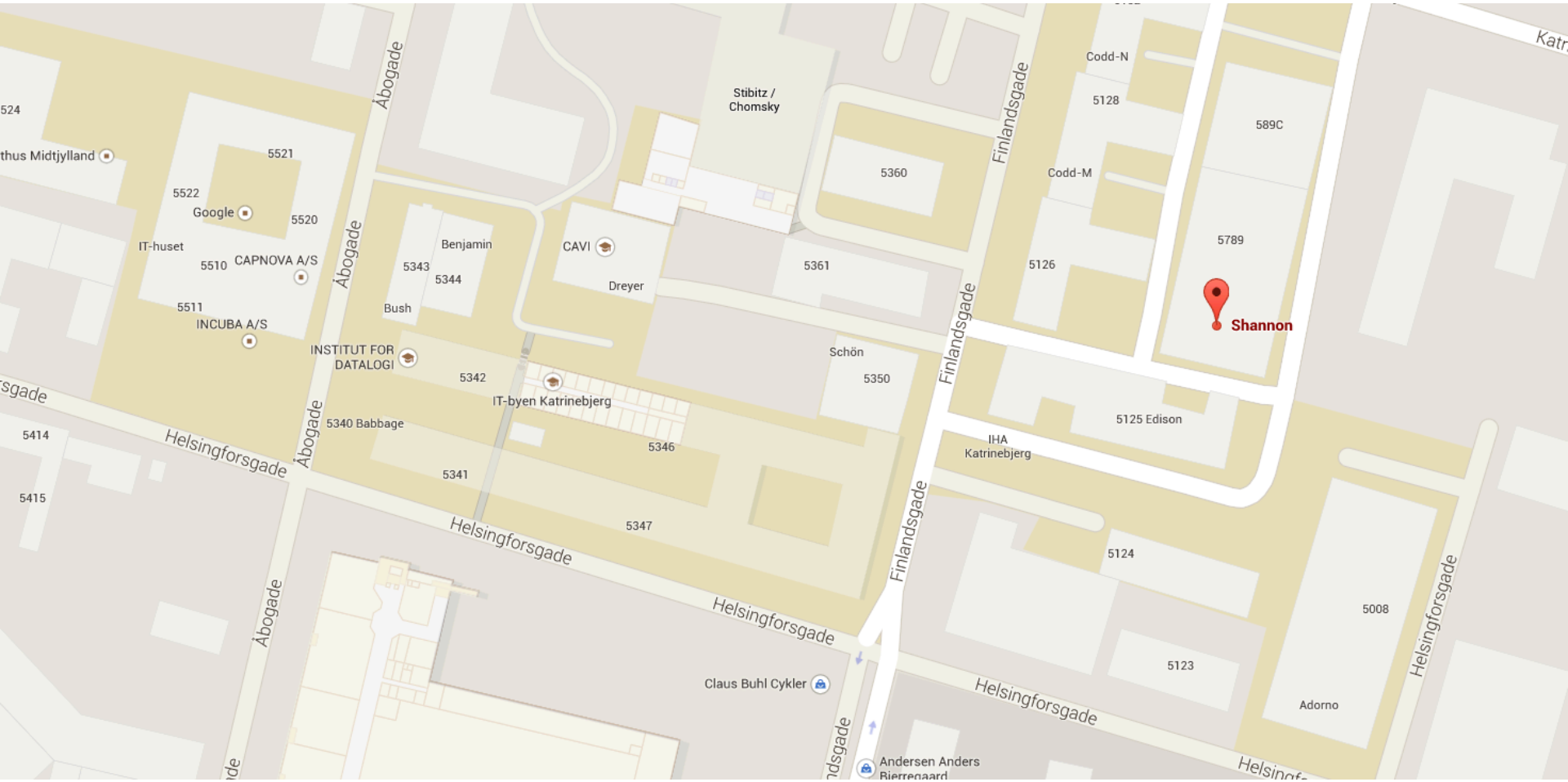
Setting it up

- Need Google Play services
- Google developer account

Map Strategies

- Dynamic
  - Generates maps as needed
  - Requires internet for downloading
- Static
  - Has e.g. vector graphics maps stored on the device
  - Uses pictures (“fixed” tiles)
- Hybrid
  - Using best available
  - Caching maps

Google Maps


Åbogade
Finlandsgade
Helsingforsgade
Stibitz / Chomsky
Codd-N
Codd-M
5128
589C
5360
5361
5126
5789
Shannon
5521
5522
Google
5520
IT-huset
5510
CAPNOVA A/S
5511
INCUBA A/S
5343
5344
Benjamin
Bush
CAVI
Dreyer
INSTITUT FOR DATALOGI
5342
IT-byen Katrinebjerg
5340 Babbage
5341
5346
5347
Schön
5350
5125 Edison
IHA Katrinebjerg
5124
5123
5008
Adorno
5414
5415
Claus Buhl Cykler
Andersen Anders

GOOGLE PLAY SERVICES & GOOGLE APIS

Google APIs

- Need Google Developer Account
  - Create online
  - Need unique authentication
    - Your SHA1 key used for signing apps
    - Application package name
    - Your API key
  - Define in manifest

Manifest for Maps

- Permissions

```
<uses-permission
android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
```

- Meta-data

```
<meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
<meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="@string/google_maps_key" />
```

Creating Google Dev Account

- Create and setup at
  - https://console.developers.google.com

Google APIs in Action

USING GOOGLE MAPS

MapFragment

Define gms.maps.SupportMapFragment in your layout XML

```
<fragment
   android:layout_width="match_parent”
   android:layout_height="match_parent"
   android:id="@+id/map”
   android:name="com.google.android.gms.maps
                          .SupportMapFragment”
 />
```

The GoogleMap

- Use Fragment to get `GoogleMap` instance

```
GoogleMap mMap =
((SupportMapFragment)
getSupportFragmentManager()
.findFragmentById(R.id.map))
.getMap()
```

- `GoogleMap` object used for manipulating the map view behavior and content

Change Camera/View

- `moveCamera(…)`
- Use `CameraUpdateFactory`
  - `newLatLng(…)`
  - `newLatLngBounds(…)`
  - `newLatLngZoom(…)`
  - `zoomIn(…)`
  - `zoomOut(…)`
  - `newCameraPosition(…)`
  - …

Change `MapType`

- Dynamically change the type of map shown
  - Normal
  - Satellite
  - Hybrid
  - Terrain

Create Custom Markers

- `addMarker(…)`
- Use `MarkerOptions` to set:
  - Position
  - Title
  - Snippet
  - Icon
  - Alpha
  - Rotation
  - Anchor
  - Flat
  - Draggable

Overlays and Drawing

- Overlays
  - `GroundOverlay`
  - `TileOverlay`
- Drawing
  - Polygons
  - Poly lines

Custom Behavior

- Override callbacks for interactions with map view in fragment
  - `OnMapClickListener`
  - `OnMarkerClickListener`
  - `OnMarkerDragListener`
  - …

GEOCODING AND GEOFENCING

Geocoding

- Translates between GPS coordinates and street addresses
  - Reverse:

    (latitude, longitude) -> Address
  - Forward:

    Address -> (latitude, longitude)
- Requires internet connection
  - A blocking call, so do in separate thread!

Reverse Geocoding

- `Geocoder.getFromLocation(lat, long, max results)`
  - Returns list of Address objects.
  - Different level of details depending on lookup result, e.g. country, street, number, …
  - Null if none found

Forward Geocoding

- `Geocoder.getFromLocationName(addressString, max results)`
  - Returns results as list of Address objects (including GPS coordinates) if found
  - Search can be limited to a certain rectangular area for map based applications

Proximity Alerts

- Do “something” when user is enters or leaves from within a certain distance of a certain location
  - Centerpoint (GPS) + radius (meters) = circle
  - Using `PendingIntent` to define what should happen
  - Use:

```
LocationManager.addProximityAlert(
lat, long, radius, expiration
time, yourPendingIntent)
```

TOOLS & TRICKS

Using the Emulator

- Faking location for test
  - Set location
  - Use gpx traces
  - Use KML
- Using maps
  - Get Google Play Services apk and Google Play Store apk (e.g APKMirror.com)
  - Install on emulator using adb

Maps in Action

EXERCISES