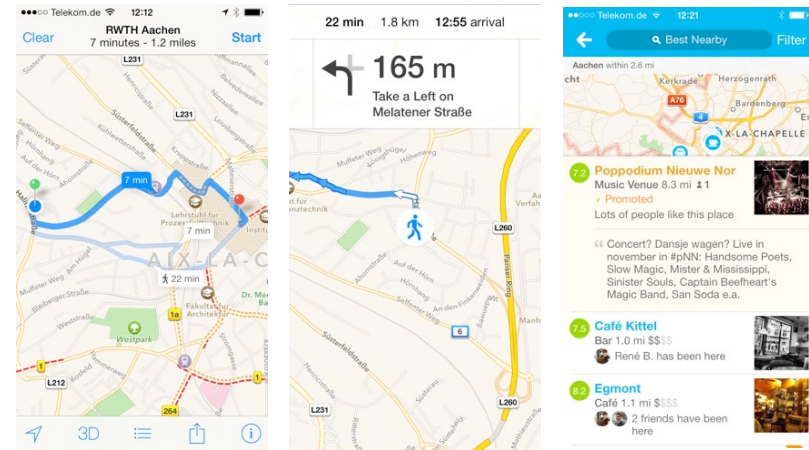


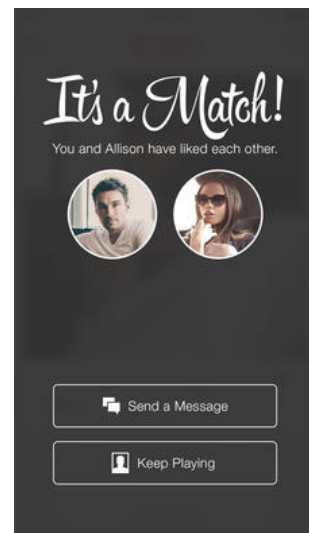
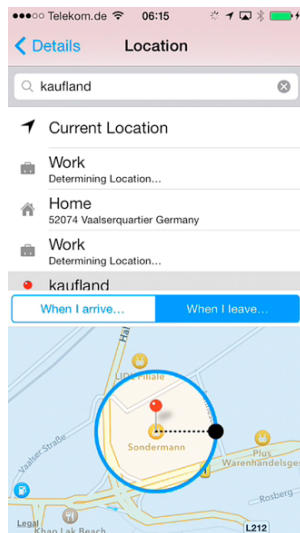
# Why Using Location and Map?

## iPhone Application Programming LI2: Location and Maps

Chat Wacharamanatham  
Media Computing Group  
RWTH Aachen University  
Winter Semester 2013/2014  
<http://hci.rwth-aachen.de/iphone>



# Why Using Location and Map?



Core Location

- Determine where you are
- Geocoding



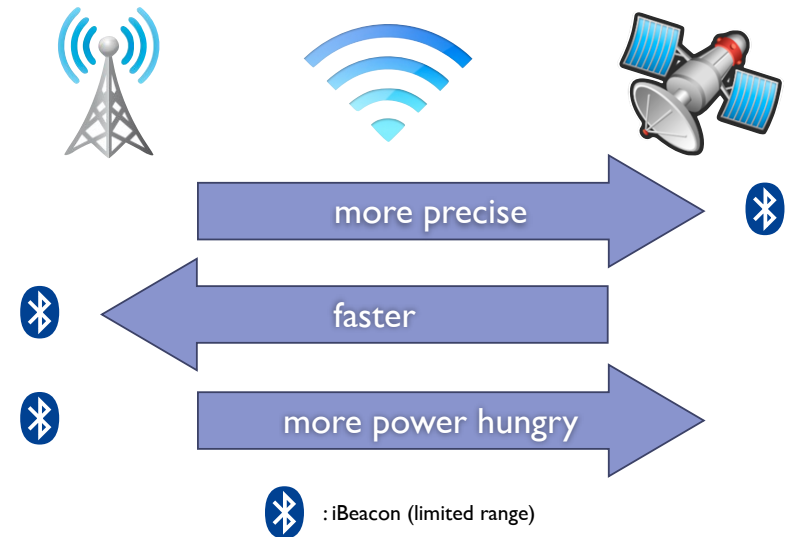
Map Kit

- Showing where you are
- Search, and routing

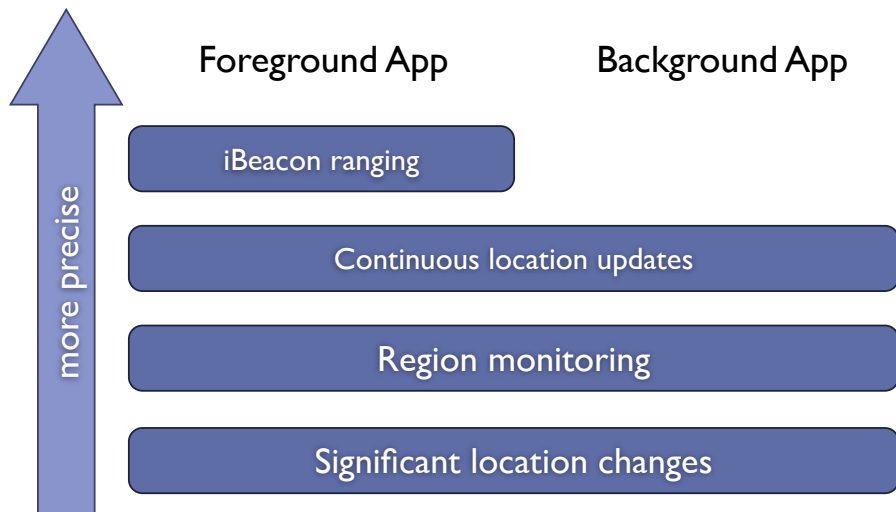


## Core Location

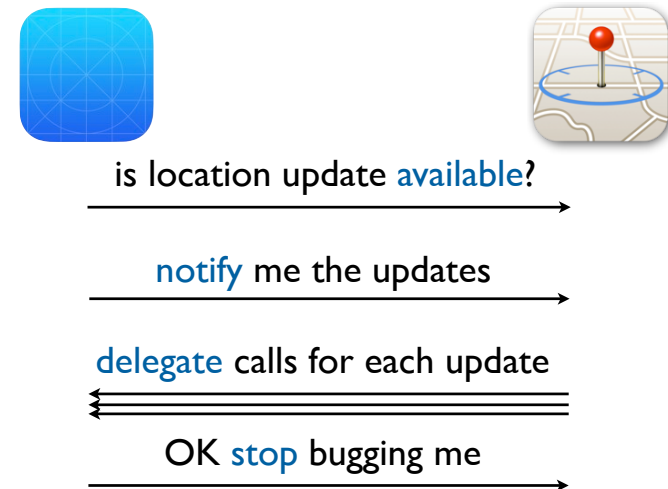
## Technology



## Location Monitoring



## Core Location Pattern



# Check for Availability

```
[CLLocationManager authorizationStatus]
[CLLocationManager locationServicesEnabled]
[CLLocationManager headingAvailable]
[CLLocationManager isMonitoringAvailableForClass:
    [CLCircularRegion class] OR
    [CLBeaconRegion class]]
[CLLocationManager isRangingAvailable]
```

- Reasons why location services are not available
  - User disabled the services in the settings app
  - Authorization to use the services denied for your app
  - Airplane mode

# Getting Location Data

```
// Get a reference to the Location Manager
self.locationManager = [[CLLocationManager alloc] init];
// and assign ourselves as delegate
self.locationManager.delegate = self;

// Start acquiring position data (start draining power)
[self.locationManager startUpdatingLocation];

#pragma mark CLLocationManagerDelegate methods
- (void) locationManager:(CLLocationManager *)manager
  didUpdateToLocation:(CLLocation *)newLocation
  fromLocation:(CLLocation *)oldLocation
{
    NSLog(@"New location: %@", newLocation);
}

- (void) locationManager:(CLLocationManager *)manager
  didFailWithError:(NSError *)error
{
    NSLog(@"An error occurred while trying to retrieve the current location");
}
```

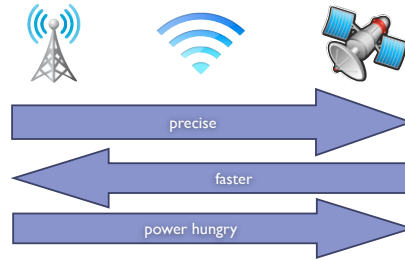
# CLLocation Object

- Coordinate (lat, long) and altitude (m)
- Horizontal and vertical accuracy (m)
- Timestamp
- Course (degree) and speed (m/s)
- distanceFromLocation: (m)

# Demo

# Location Manager Settings

- Desired accuracy
  - Best for navigation
  - Best
  - Nearest ten meters
  - Hundred meters
  - Kilometer
  - Three kilometers
- Distance Filter
  - The minimum distance a device must move laterally before an update event is generated.



# Significant Location Changes

```
- (IBAction) startSignificantChangeUpdates:(id)sender {
    [self.locationManager startMonitoringSignificantLocationChanges];
}

- (IBAction) stopSignificantChangeUpdates:(id)sender {
    [self.locationManager stopMonitoringSignificantLocationChanges];
}

#pragma mark CLLocationManagerDelegate methods
- (void) locationManager:(CLLocationManager *)manager
  didUpdateToLocation:(CLLocation *)newLocation
  fromLocation:(CLLocation *)oldLocation
{
    NSLog(@"New location: %@", newLocation);
}

- (void) locationManager:(CLLocationManager *)manager
  didFailWithError:(NSError *)error
{
    NSLog(@"An error occurred while trying to retrieve the current location");
}
```

# Monitoring Regions

- Monitor geographic boundary crossing
- Runs in the background
- System launches your app in the background
- Circular regions (GPS) or iBeacon regions (Bluetooth LE)

# Monitoring Regions

```
- (BOOL) registerRegionWithOrigin:(CLLocationCoordinate2D)origin
  radius:(CLLocationDistance)rad
  andIdentifier:(NSString *)id
{
    // Check if region monitoring services are available and enabled
    if (![CLLocationManager isMonitoringAvailableForClass:
        [CLCircularRegion class]])
        return NO;

    // Clamp the radius to the maximum value
    CLLocationDistance radius = rad;
    if (radius > self.locationManager.maximumRegionMonitoringDistance)
        radius = self.locationManager.maximumRegionMonitoringDistance;

    // Create the region and add it to the list of monitored regions
    CLRegion *region = [[CLRegion alloc] initWithCircularRegionWithCenter:origin
        radius:radius
        identifier:identifier];

    region.notifyOnEntry = YES;
    region.notifyOnExit = NO;
    [self.locationManager startMonitoringForRegion:region
        desiredAccuracy:kCLLocationAccuracyHundredMeters];

    return YES;
}
```

# CLLocationManager Delegate

```
- (void)locationManager:(CLLocationManager *)manager
  didEnterRegion:(CLRegion *)region

- (void)locationManager:(CLLocationManager *)manager
  didExitRegion:(CLRegion *)region

- (void)locationManager:(CLLocationManager *)manager
  monitoringDidFailForRegion:(CLRegion *)region
  withError:(NSError *)error

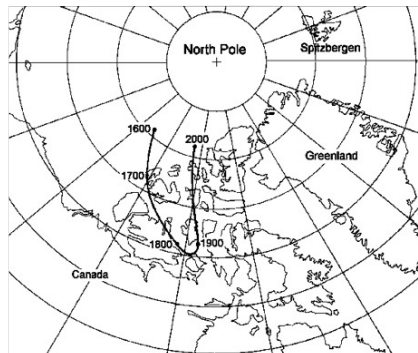
- (void)locationManager:(CLLocationManager *)manager
  didDetermineState:(CLRegionState)state
  forRegion:(CLRegion *)region
```

## iBeacon Demo



## Heading Information

- Magnetic north & true north
- Integrated magnetometer
- GPS-Hardware



## Getting Heading

```
- (IBAction) startUpdatingHeading:(id)sender;
{
  // locationManager is a CLLocationManager
  self.locationManager.delegate = self;

  // enable location services to get true heading
  // not needed if magnetic north is sufficient
  self.locationManager.distanceFilter = 1000;
  self.locationManager.desiredAccuracy =
  kCLLocationAccuracyKilometer;
  [self.locationManager startUpdatingLocation];

  if ([CLLocationManager headingAvailable]) {
    self.locationManager.headingFilter = 0;
    [self.locationManager startUpdatingHeading];
  }
}
```

# CLLocation Manager Delegate

```
@property (assign) CLLocationDirection heading;

#pragma mark CLLocationManagerDelegate methods
- (void)locationManager:(CLLocationManager *)manager
  didUpdateHeading:(CLHeading *)newHeading {

  // check if heading is valid (>0)
  if (newHeading.headingAccuracy < 0)
    return;
  // check if true heading is valid, otherwise use magnetic heading
  if (newHeading.trueHeading > 0) {
    self.heading = newHeading.trueHeading;
  }
  else {
    self.heading = newHeading.magneticHeading;
  }
}
```

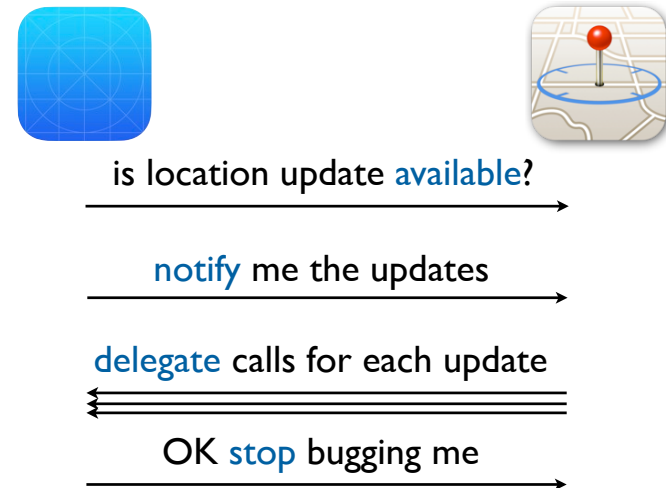
# Saving Battery Power

- Turn off location services when not used
- Use the significant location change service whenever possible
- Use low-resolution accuracy values
  - Unless doing so would impair your application
- Turn off location services if the accuracy does not increase

# Multitasking

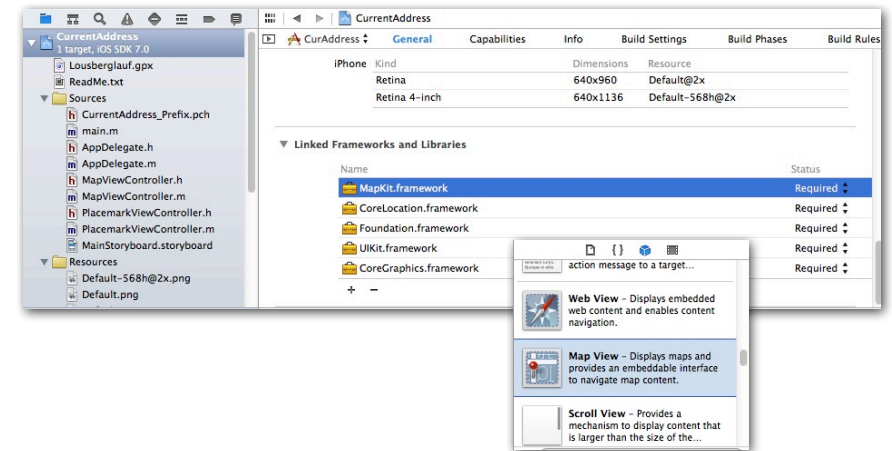
- User can disable multitasking across the device, or disable multitasking for your app, or quit the app from app switcher
- UIBackgroundModes key with location value
  - Must start location in foreground
  - Or use significant location change
- Deferring location updates (phone off, GPS on = 40% battery saving)

# Core Location Pattern



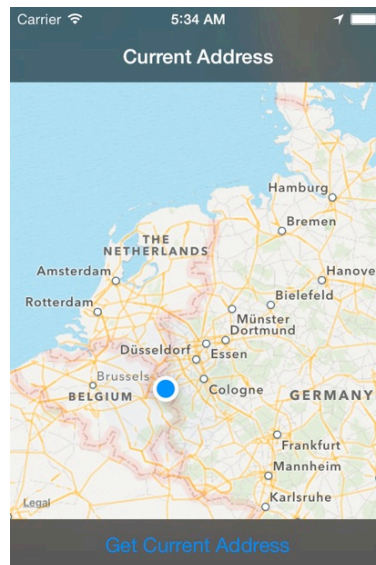


## MapKit



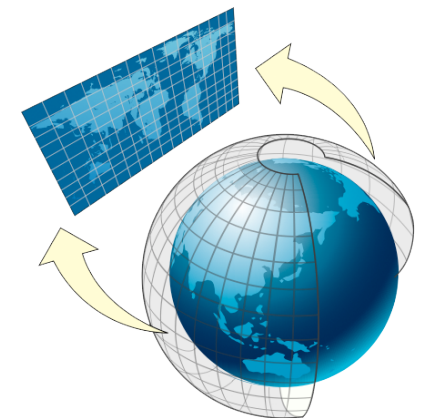
## MKMapView

- MKMapViewDelegate
- User location
- User heading
- Annotations



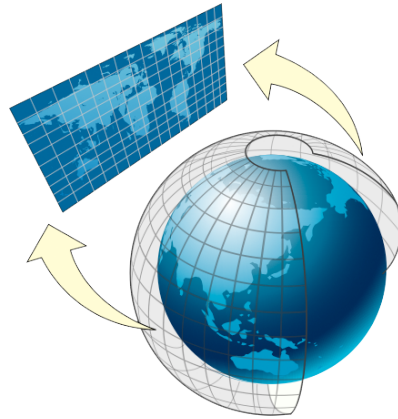
## Map Geometry

- Mercator projection
- Useful for navigation
- Lines of constant course
- Distorts areas far from the equator



# MapKit Units

- Map points (x,y)
  - MKMapPoint
  - MKMapSize
  - MKMapRect
- Map coordinates (lat, long)
  - CLLocationCoordinate2D
  - MKCoordinateSpan
  - MKCoordinateRegion



# Unit Conversion

Convert from	Convert to	Routines
Map coordinates	Points	convertCoordinate: toPointToView: (MKMapView) convertRegion: toRectToView: (MKMapView)
Map coordinates	Map points	MKMapPointForCoordinate (MapKit)
Map Points	Map coordinates	MKCoordinateForMapPoint (MapKit) MKCoordinateRegionForMapRect (MapKit)
Map Points	Points	pointForMapPoint: (MKOverlayRenderer) rectForMapRect: (MKOverlayRenderer)
Points	Map coordinates	convertPoint: toCoordinateFromView: (MKMapView) convertRect: toRegionFromView: (MKMapView)
Points	Map points	mapPointForPoint: (MKOverlayRenderer) mapRectForRect: (MKOverlayRenderer)

# Demo

# Using Maps

```

self.mapView.showsUserLocation = YES;

// Change map mode to satellite
self.mapView.mapType = MKMapTypeSatellite;

// Create the coordinate of the RWTH main building
CLLocationCoordinate2D rwth =
    CLLocationCoordinate2DMake(50.77761254285463, 6.077756881713867);

// Show an area of 400m x 400m around the main building
self.mapView.region =
    MKCoordinateRegionMakeWithDistance(rwth, 400, 400);
    
```



# Using Maps

```
- (IBAction) moveLeft:(id)sender;
{
    // Get the current center
    CLLocationCoordinate2D mapCenter = self.mapView.centerCoordinate;

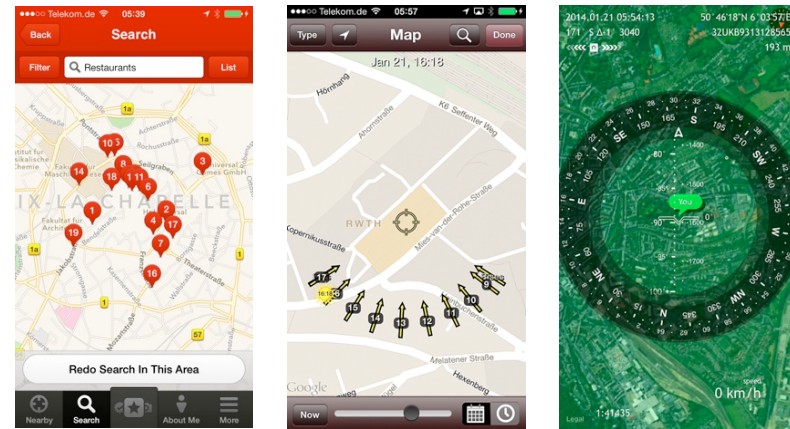
    // Move it to the left by half the width of the frame
    mapCenter = [self.mapView convertPoint:CGPointMake(1,
        (self.mapView.frame.size.height/2.0))
        toCoordinateFromView:self.mapView];

    // Make the transition animated to let the user follow what happens
    [self.mapView setCenterCoordinate:mapCenter animated:YES];

    // Zoom out
    MKCoordinateRegion region = self.mapView.region;

    region.span.longitudeDelta *= 2.0;
    region.span.latitudeDelta *= 2.0;
    [self.mapView setRegion:region animated:YES];
}
```

# Adding Annotations



- Custom class Implements the `MKAnnotation` protocol

# Annotations

```
@interface MapManAnnotation : NSObject <MKAnnotation> {
    CLLocationCoordinate2D coordinate;
    NSString *title;
    NSString *subtitle;
}

- (id)initWithLocation:(CLLocationCoordinate2D)aCoordinate
    title:(NSString *)theTitle
    subtitle:(NSString *)theSubtitle;

@end
```

# Annotations

```
@implementation MapManAnnotation

@synthesize coordinate,title, subtitle;

- (id)initWithLocation:(CLLocationCoordinate2D)aCoordinate
    title:(NSString *)theTitle
    subtitle:(NSString *)theSubtitle;
{
    self = [super init];
    if (self != nil) {
        coordinate = aCoordinate;
        title = theTitle;
        subtitle = theSubtitle;
    }
    return self;
}

@end
```

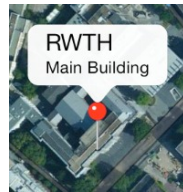
# Annotations

```
- (IBAction) addAnnotation:(id)sender;
{
    // Create the coordinate of the RWTH main building
    CLLocationCoordinate2D rwth =
        CLLocationCoordinate2DMake(50.77761254285463, 6.077756881713867);

    // Create an annotation object
    MapManAnnotation *ma =
        [[MapManAnnotation alloc] initWithLocation:rwth];
    ma.title = @"RWTH";
    ma.subtitle = @"Main Building";

    [self.mapView addAnnotation:(id)ma];

    ...
}
```



# Annotations

```
#pragma mark MapView delegate methods
- (MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:
(id <MKAnnotation>)annotation
{
    if ([annotation.title isEqualToString:@"Rathaus"])
    {
        MKAnnotationView* aView = [[MKAnnotationView alloc]
            initWithAnnotation:annotation
            reuseIdentifier:@"Rathaus"];
        aView.image = [UIImage imageNamed:@"Rathaus.png"];
        aView.centerOffset = CGPointMake(10., -20.);
        return aView;
    }
    return nil;
}
```



# Overlays

```
- (IBAction)addOverlay:(id)sender;
{
    // Define an overlay that covers the CS building.
    CLLocationCoordinate2D points[6];
    points[0] = CLLocationCoordinate2DMake(50.779396749979426, 6.058316230773926);
    points[1] = CLLocationCoordinate2DMake(50.77815527465925, 6.059163808822632);
    points[2] = CLLocationCoordinate2DMake(50.77787712539619, 6.061438322067261);
    points[3] = CLLocationCoordinate2DMake(50.779247503323184, 6.060891151428223);
    points[4] = CLLocationCoordinate2DMake(50.7791186081004, 6.06020450592041);
    points[5] = CLLocationCoordinate2DMake(50.77976986453611, 6.059743165969849);

    MKPolygon* csBuilding = [MKPolygon polygonWithCoordinates:points count:6];

    csBuilding.title = @"CS Building";

    [self.mapView addOverlay:csBuilding level:MKOverlayLevelAboveRoads];
}
```

# Overlay Renderer

```
- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
rendererForOverlay:(id<MKOverlay>)overlay;
{
    MKOverlayRenderer *renderer;
    UIColor *cyan = [[UIColor cyanColor] colorWithAlphaComponent:
0.3];
    UIColor *blue = [[UIColor blueColor] colorWithAlphaComponent:
0.8];
    if ([overlay.title isEqualToString:@"CS Building"])
    {
        MKPolygonRenderer* csRenderer = [[MKPolygonRenderer alloc]
            initWithPolygon:(MKPolygon*)overlay];
        csRenderer.fillColor = cyan;
        csRenderer.strokeColor = blue;
        csRenderer.lineWidth = 2;
        renderer = csRenderer;
    }
    return renderer;
}
```





## MapKit

## MapKit



## GoogleMaps



## OpenStreetMap



Map Image Data

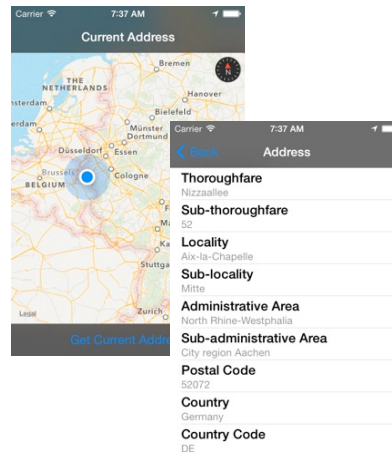
Semantic Data

<https://developers.google.com/maps/documentation/ios/>

<http://wiki.openstreetmap.org/wiki/Develop>

## Geocoding

- Convert coordinate to an address or
- Convert address to coordinates
- More user-friendly
- CLGeocoder
  - Limited resources
  - Requires live network connection



## Reverse Geocoding

```
self.geocoder = [[CLGeocoder alloc] init];

[self.geocoder
 reverseGeocodeLocation:self.mapView.userLocation.location
 completionHandler:^(NSArray *placemarks, NSError *error)
 {
     self.placemark = [placemarks objectAtIndex:0];

     NSLog(@"%@", placemark.thoroughfare);
     NSLog(@"%@", placemark.subThoroughfare);
     NSLog(@"%@", placemark.locality);
     NSLog(@"%@", placemark.subLocality);
     NSLog(@"%@", placemark.administrativeArea);
     NSLog(@"%@", placemark.subAdministrativeArea);
     NSLog(@"%@", placemark.postalCode);
     NSLog(@"%@", placemark.country);
     NSLog(@"%@", placemark.ISOCountryCode);

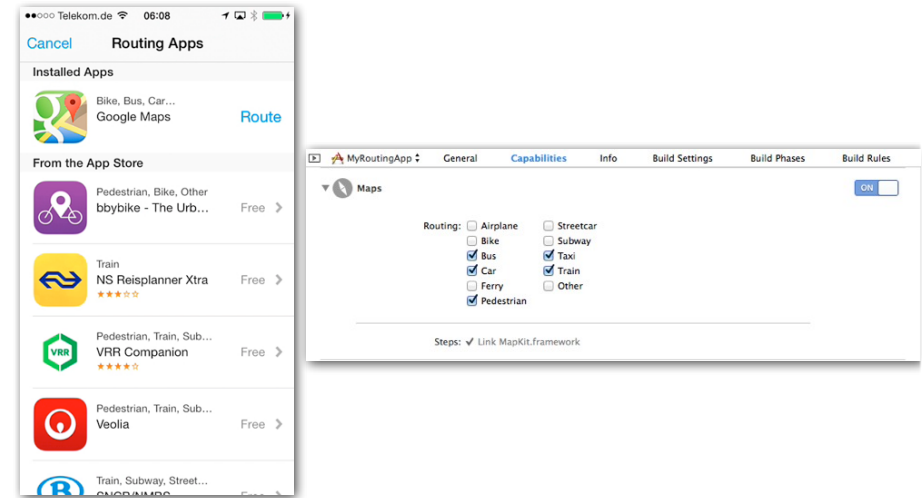
 }];
```

# Natural Language Search

```
- (IBAction)findPizza:(id)sender
{
    // request object
    MKLocalSearchRequest *request = [[MKLocalSearchRequest alloc] init];
    request.naturalLanguageQuery = @"Pizza";
    request.region = self.mapView.region;

    // search object
    MKLocalSearch *search = [[MKLocalSearch alloc] initWithRequest:request];
    [search startWithCompletionHandler:^(MKLocalSearchResponse *response,
                                       NSError *error)
    {
        NSMutableArray *placemarks = [NSMutableArray array];
        for (MKMapItem *item in response.mapItems) {
            [placemarks addObject:item.placemark];
        }
        [self.mapView showAnnotations:placemarks animated:NO];
    }];
}
```

# Integrating with iOS Routing



## Summary

- Core Location
- Compass
- MapKit
  
- Reading Assignment:
  - Location Awareness Programming Guide