Staying on Track with Location Services

Session 303

Jay Bruins
Software Engineer

What We'll Cover

What We'll Cover

- Overview

What We'll Cover

- Overview
- Improvements

What We'll Cover

- Overview
- Improvements
- New APIs

What We'll Cover

- Overview
- Improvements
- New APIs
- Tips, tricks, and myths

CoreLocation

Refresher

Why Is Location Powerful?

Why Is Location Powerful?

Location provides context

Why Is Location Powerful?

Location provides context

- iOS: Inherently mobile

Why Is Location Powerful?

Location provides context

- iOS: Inherently mobile
- Enhances the user experience

Why Is Location Powerful?

Location provides context

- iOS: Inherently mobile
- Enhances the user experience
- More intuitive apps

Why Is Location Powerful?

Location provides context

- iOS: Inherently mobile
- Enhances the user experience
- More intuitive apps
- Enables new use cases

Location Technologies

Location Technologies

- Cellular positioning

Location Technologies

- Cellular positioning
- Wi-Fi positioning

Location Technologies

- Cellular positioning
- Wi-Fi positioning
- Global Navigation Satellite Systems (GNSS)

Location Technologies

- Cellular positioning
- Wi-Fi positioning
- Global Navigation Satellite Systems (GNSS)
  - NavStar (GPS)

Location Technologies

- Cellular positioning
- Wi-Fi positioning
- Global Navigation Satellite Systems (GNSS)
  - NavStar (GPS)
  - GLONASS

CoreLocation API

CoreLocation API

```
- (void)setupLocation
{
    self.manager = [CLLocationManager new];
    self.manager.delegate = self;
    self.manager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
}
- (void)startLocation
{
    [self.manager startUpdatingLocation];
}
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    self.mapView.userPosition = newLocation;
}
```

CoreLocation API

```
- (void)setupLocation
{
    self.manager = [CLLocationManager new];
    self.manager.delegate = self;
    self.manager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
}
- (void)startLocation
{
    [self.manager startUpdatingLocation];
}
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    self.mapView.userPosition = newLocation;
}
```

CoreLocation API

```
- (void)setupLocation
{
    self.manager = [CLLocationManager new];
    self.manager.delegate = self;
    self.manager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
}
- (void)startLocation
{
    [self.manager startUpdatingLocation];
}
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    self.mapView.userPosition = newLocation;
}
```

CoreLocation API

```
- (void)setupLocation
{
    self.manager = [CLLocationManager new];
    self.manager.delegate = self;
    self.manager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
}
- (void)startLocation
{
    [self.manager startUpdatingLocation];
}
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    self.mapView.userPosition = newLocation;
}
```

CoreLocation API

```
- (void)setupLocation
{
    self.manager = [CLLocationManager new];
    self.manager.delegate = self;
    self.manager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
}
- (void)startLocation
{
    [self.manager startUpdatingLocation];
}
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    self.mapView.userPosition = newLocation;
}
```

CoreLocation API

```
- (void)setupLocation
{
    self.manager = [CLLocationManager new];
    self.manager.delegate = self;
    self.manager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
}
- (void)startLocation
{
    [self.manager startUpdatingLocation];
}
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
           fromLocation:(CLLocation *)oldLocation
{
    self.mapView.userPosition = newLocation;
}
```

What's New?

Improvements to Location Services

What's New?

What's New?

- Improved accuracy

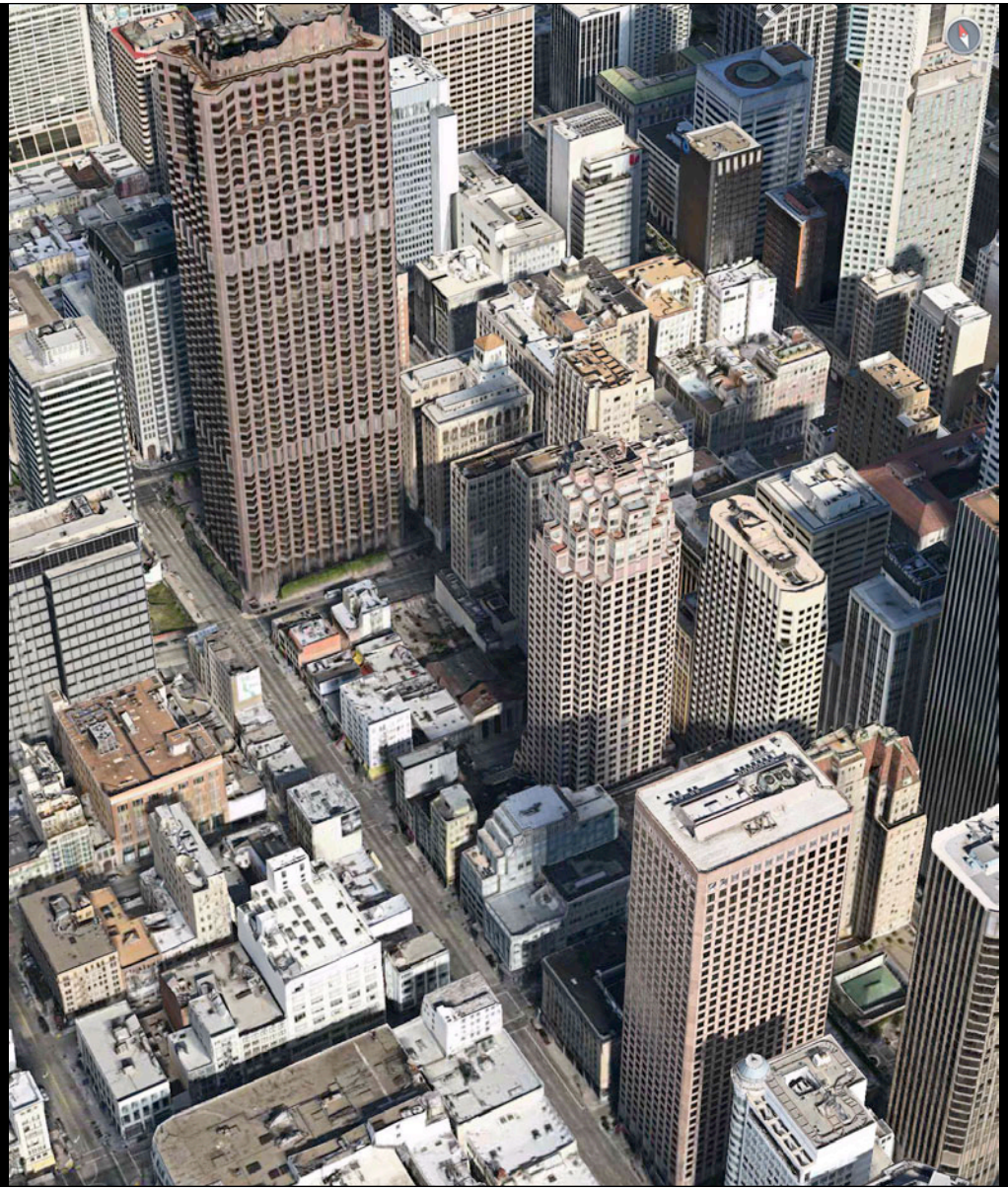
What's New?

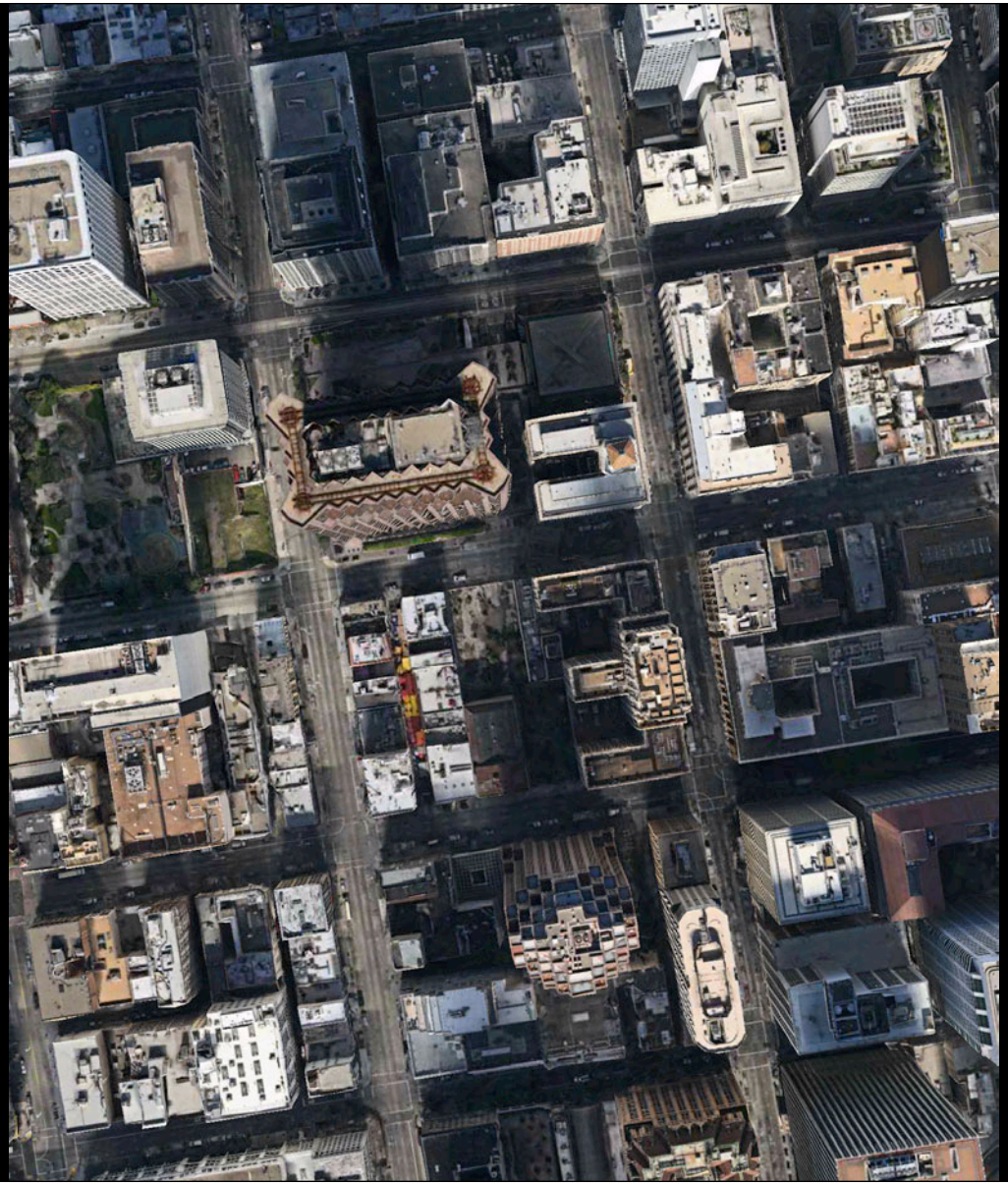
- Improved accuracy
- Improved availability

What's New?

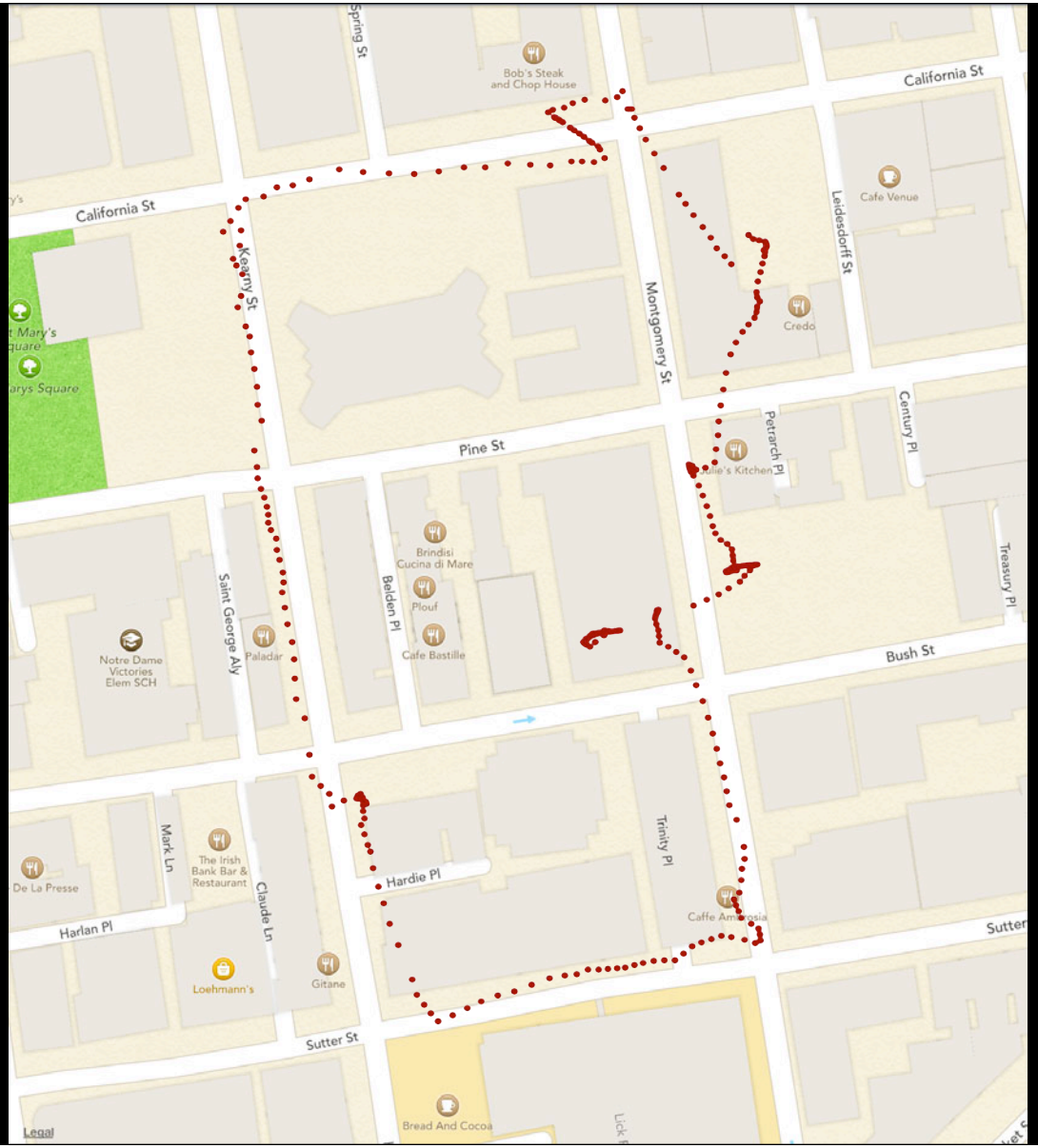
- Improved accuracy
- Improved availability
- Lowered power consumption

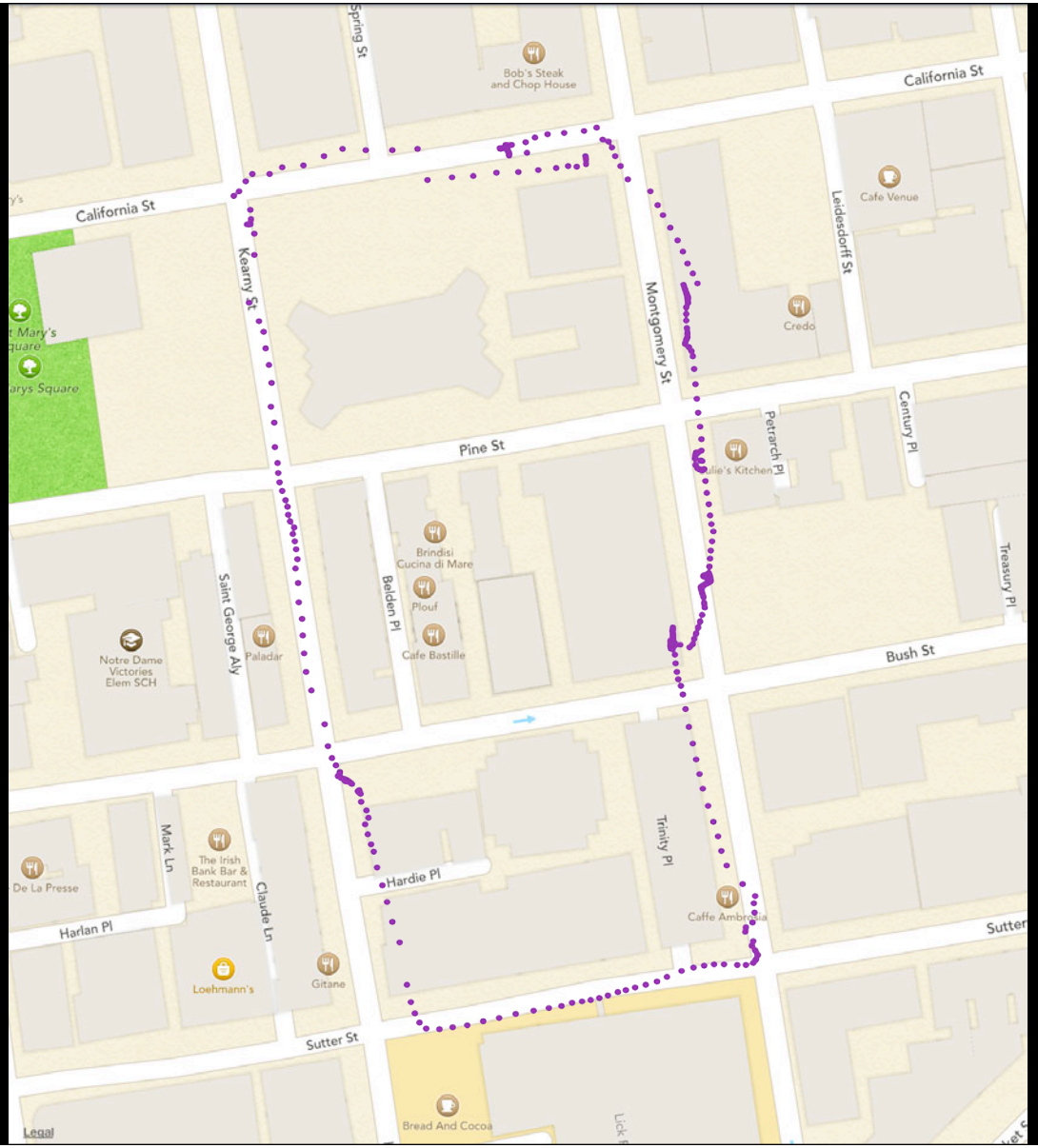
Improved Accuracy

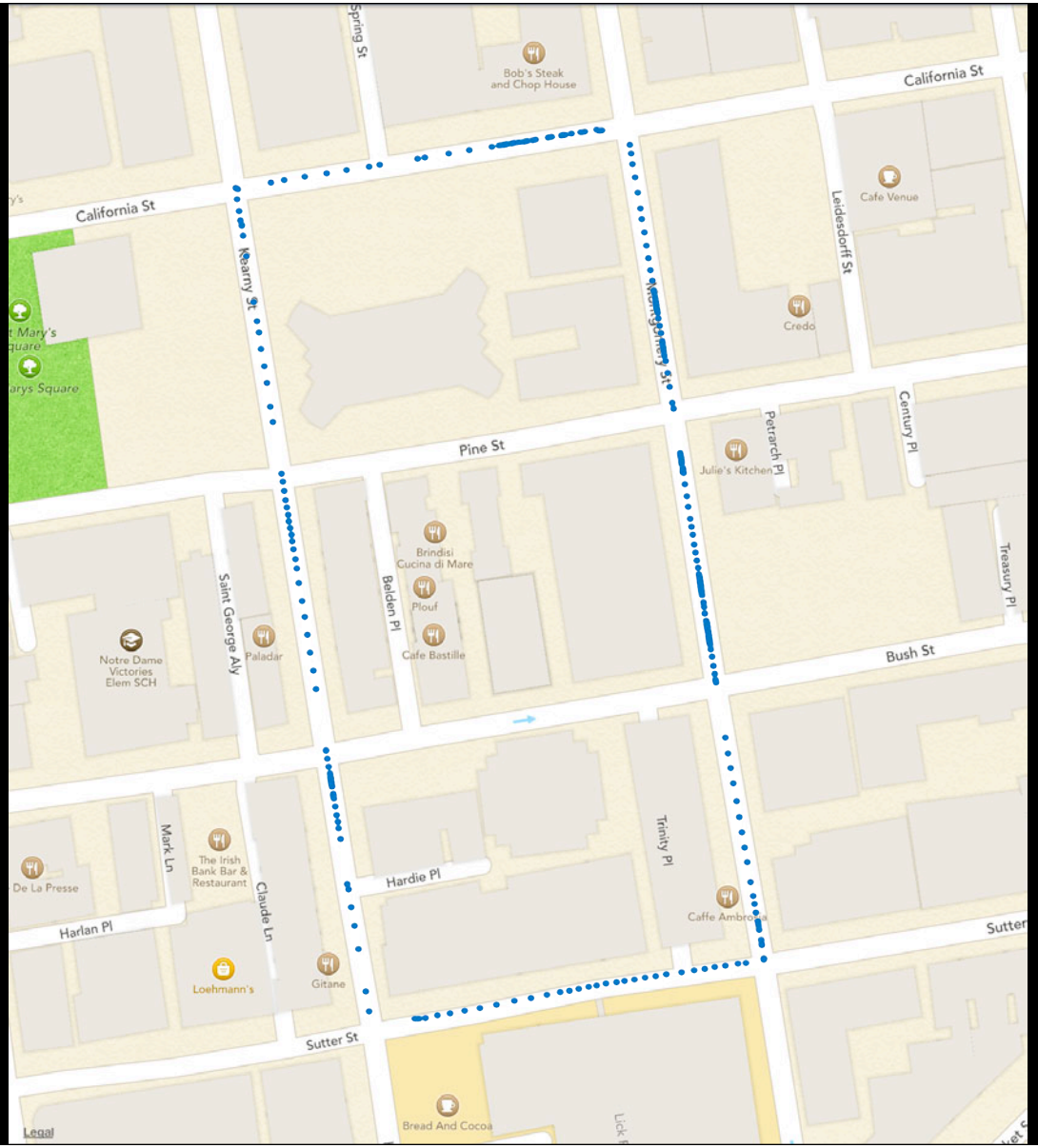




Spring St
Bob's Steak and Chop House
California St
Cafe Venue
Leidesdorff St
Kearny St
Montgomery St
Credo
Pine St
Petrarch Pl
Century Pl
Julie's Kitchen
Brindisi Cucina di Mare
Plouf
Cafe Bastille
Belden Pl
Saint George Aly
Paladar
Notre Dame Victories Elem SCH
Treasury Pl
Bush St
Mark Ln
The Irish Bank Bar & Restaurant
Claude Ln
De La Presse
Harlan Pl
Hardie Pl
Trinity Pl
Caffe Ambrosia
Loehmann's
Gitane
Sutter St
Bread And Cocoa
Legal


Bob's Steak and Chop House
California St
Spring St
Cafe Venue
Leidesdorff St
Kearny St
Montgomery St
Credo
Pine St
Petrarch Pl
Century Pl
Brindisi Cucina di Mare
Plouf
Belden Pl
Saint George Aly
Paladar
Cafe Bastille
Notre Dame Victories Elem SCH
Treasury Pl
Bush St
Mark Ln
The Irish Bank Bar & Restaurant
Claude Ln
Trinity Pl
Hardie Pl
De La Presse
Harlan Pl
Loehmann's
Gitane
Sutter St
Bread And Cocoa
Legal


Bob's Steak and Chop House
California St
Spring St
Cafe Venue
Leidesdorff St
Kearny St
Montgomery St
Credo
Pine St
Petrarch Pl
Century Pl
Brindisi Cucina di Mare
Plouf
Cafe Bastille
Belden Pl
Saint George Aly
Paladar
Notre Dame Victories Elem SCH
Treasury Pl
Bush St
Trinity Pl
Mark Ln
The Irish Bank Bar & Restaurant
De La Presse
Claude Ln
Hardie Pl
Harlan Pl
Loehmann's
Gitane
Sutter St
Bread And Cocoa
Legal


Spring St
Bob's Steak and Chop House
California St
Cafe Venue
Leidesdorff St
Kearny St
Montgomery St
Credo
Pine St
Julie's Kitchen
Petrarch Pl
Century Pl
Treasury Pl
Brindisi Cucina di Mare
Plouf
Cafe Bastille
Belden Pl
Saint George Aly
Paladar
Notre Dame Victories Elem SCH
Bush St
Trinity Pl
Mark Ln
The Irish Bank Bar & Restaurant
Claude Ln
De La Presse
Hardie Pl
Harlan Pl
Caffe Ambrosi
Loehmann's
Gitane
Sutter St
Bread And Cocoa
Legal

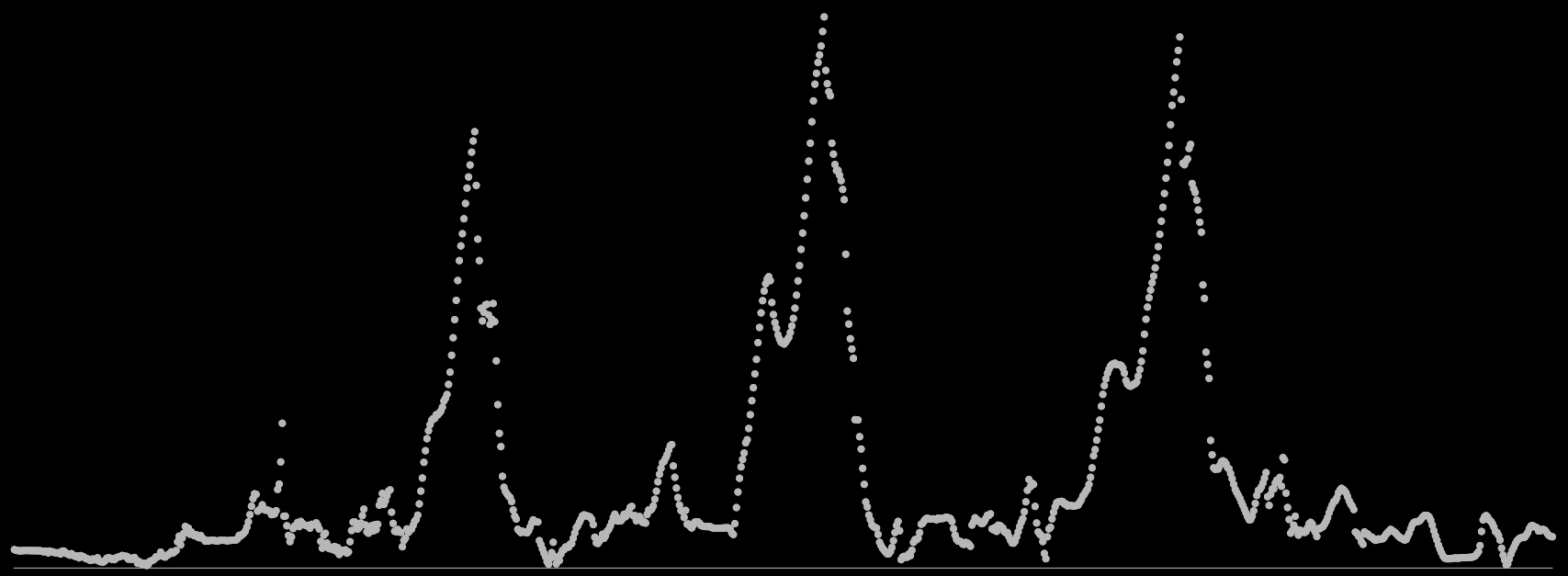
Wi-Fi–Aided Location

Wi-Fi–Aided Location

Reduced horizontal outliers

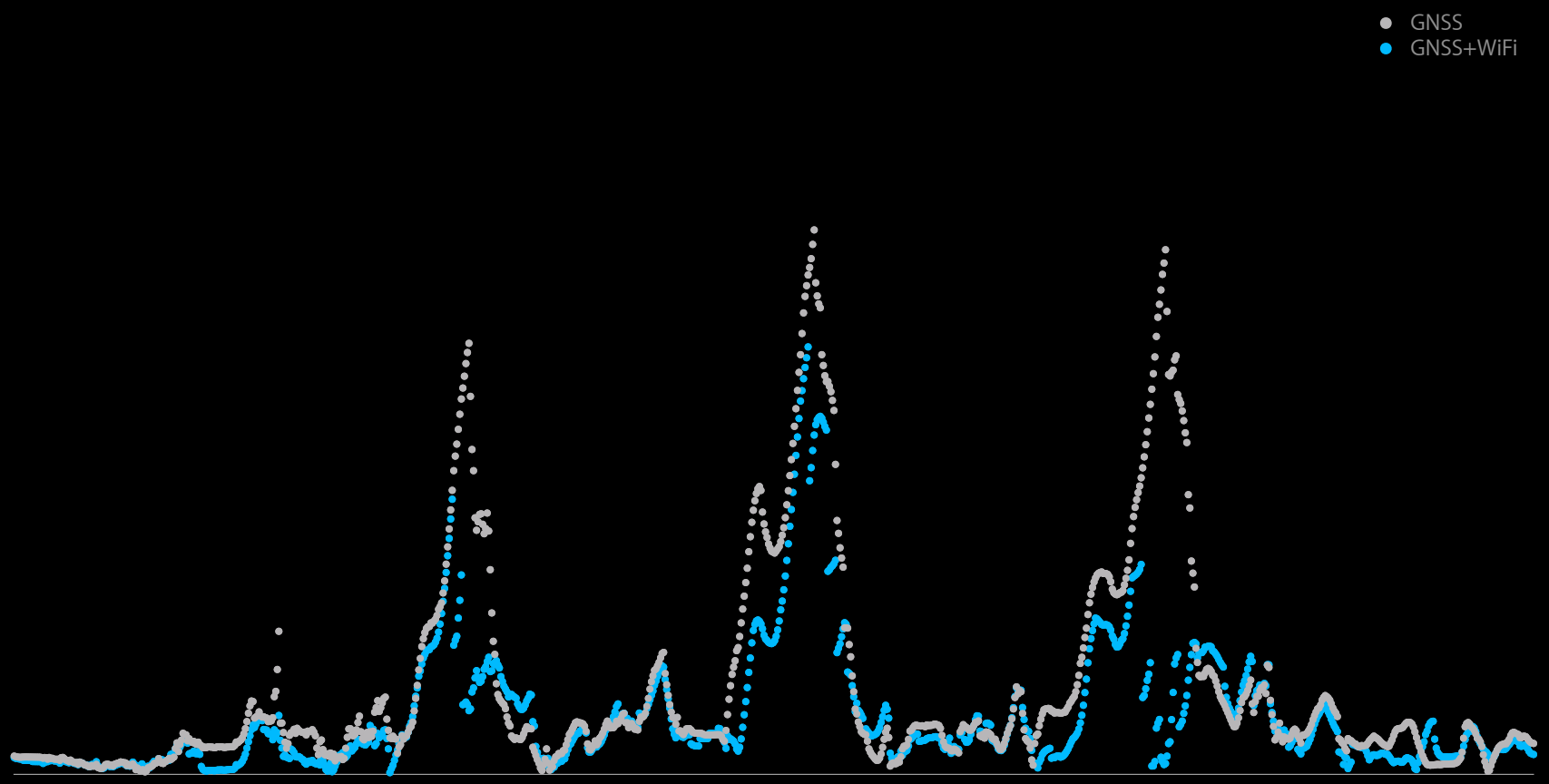
Wi-Fi–Aided Location

Reduced horizontal outliers



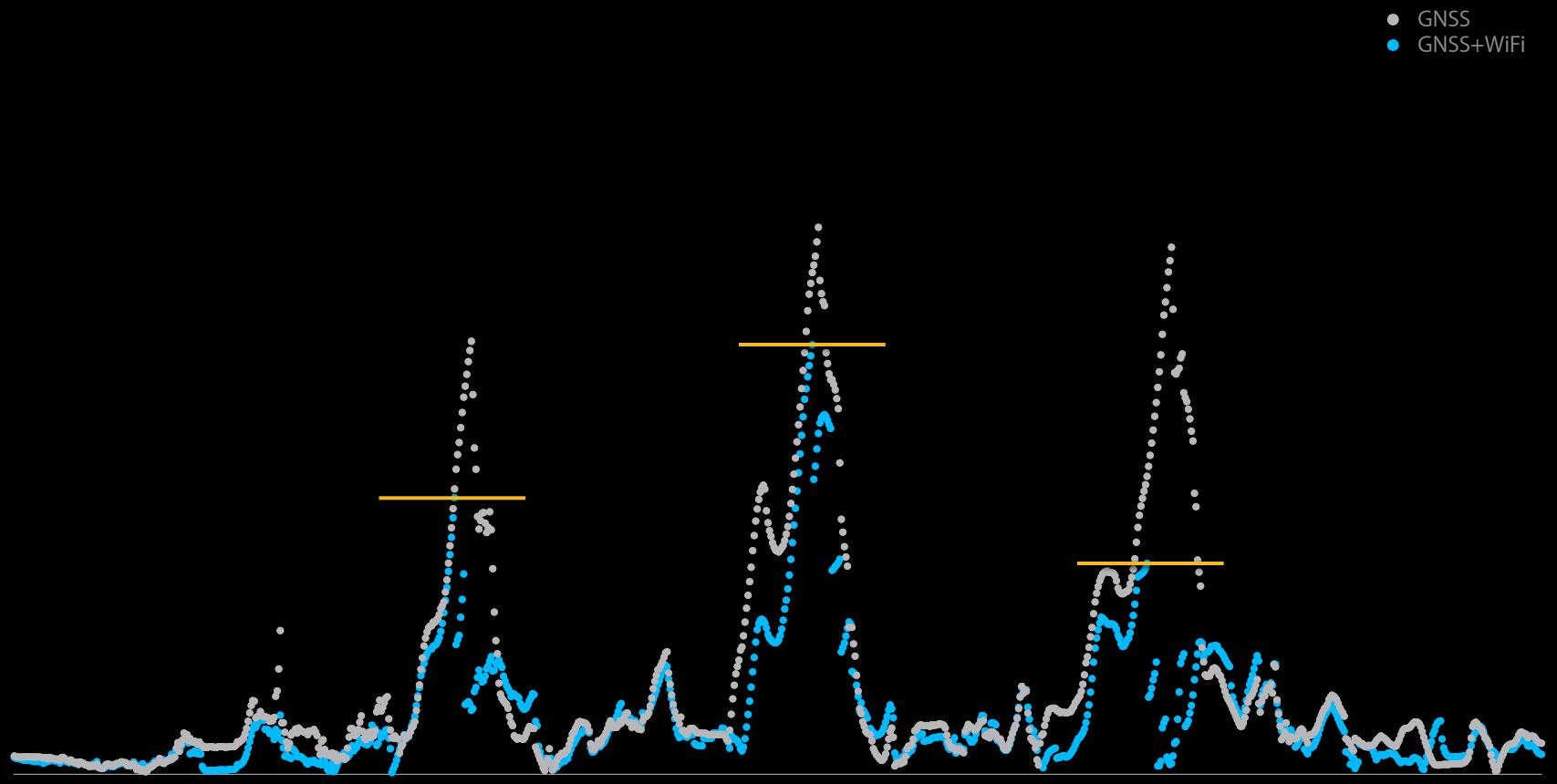
Wi-Fi–Aided Location

Reduced horizontal outliers

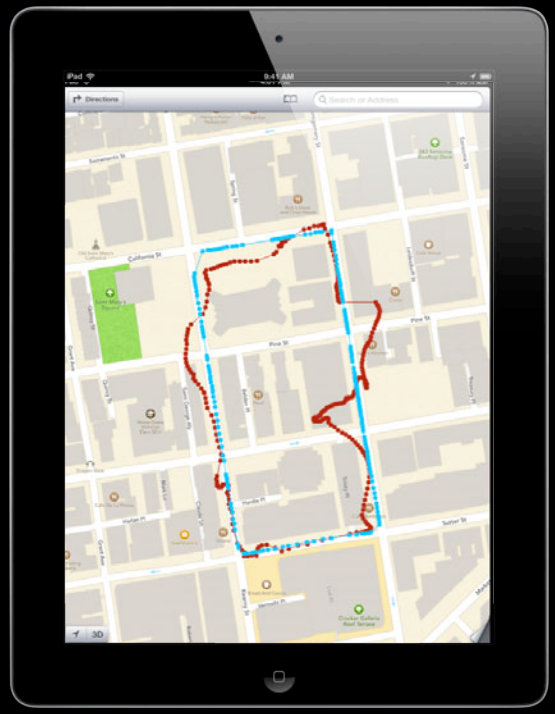


Wi-Fi–Aided Location

Reduced horizontal outliers

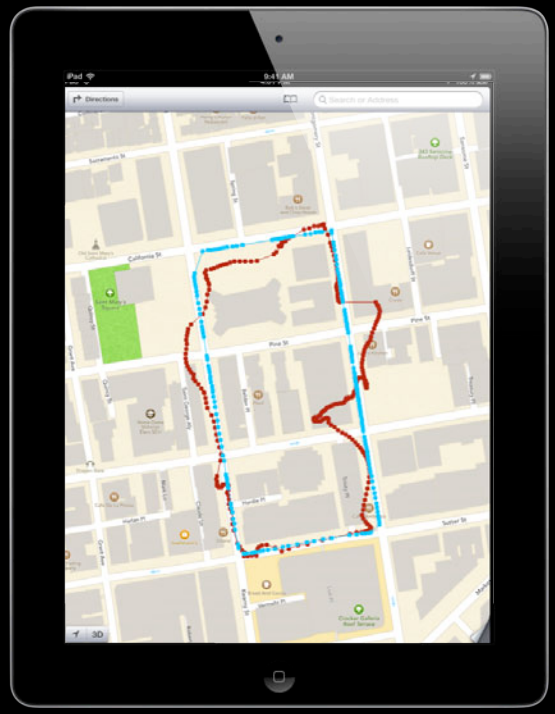


Map-Aided Location



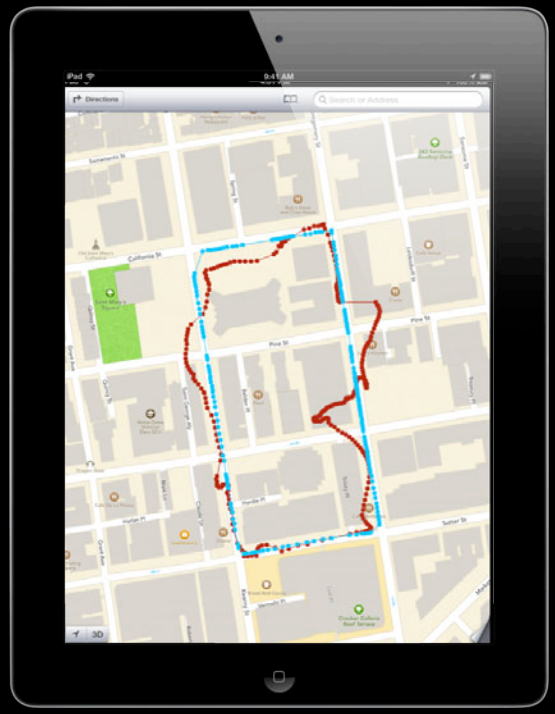
Map-Aided Location

- Location and course



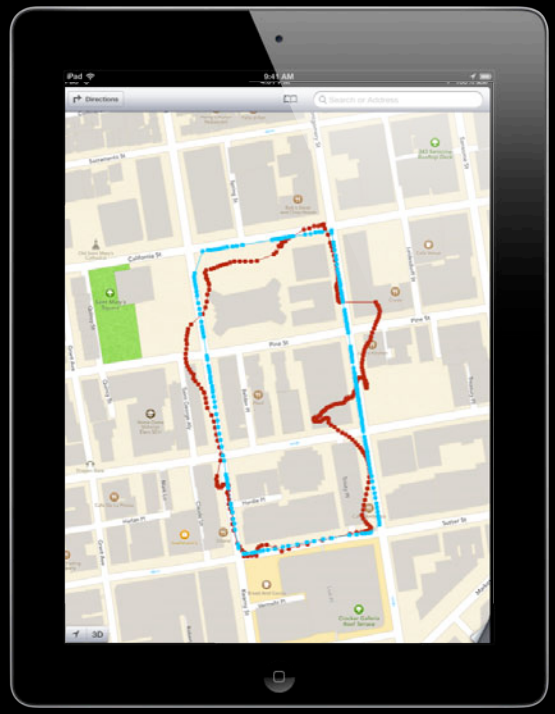
Map-Aided Location

- Location and course
- Driving



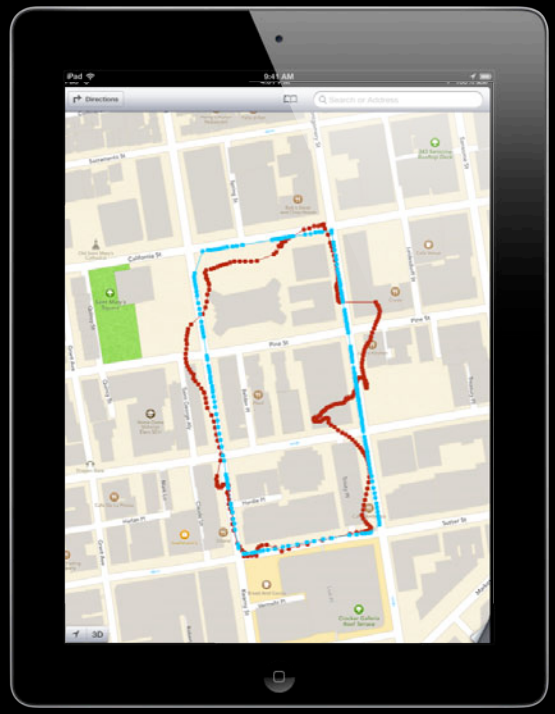
Map-Aided Location

- Location and course
- Driving
- Vector map data



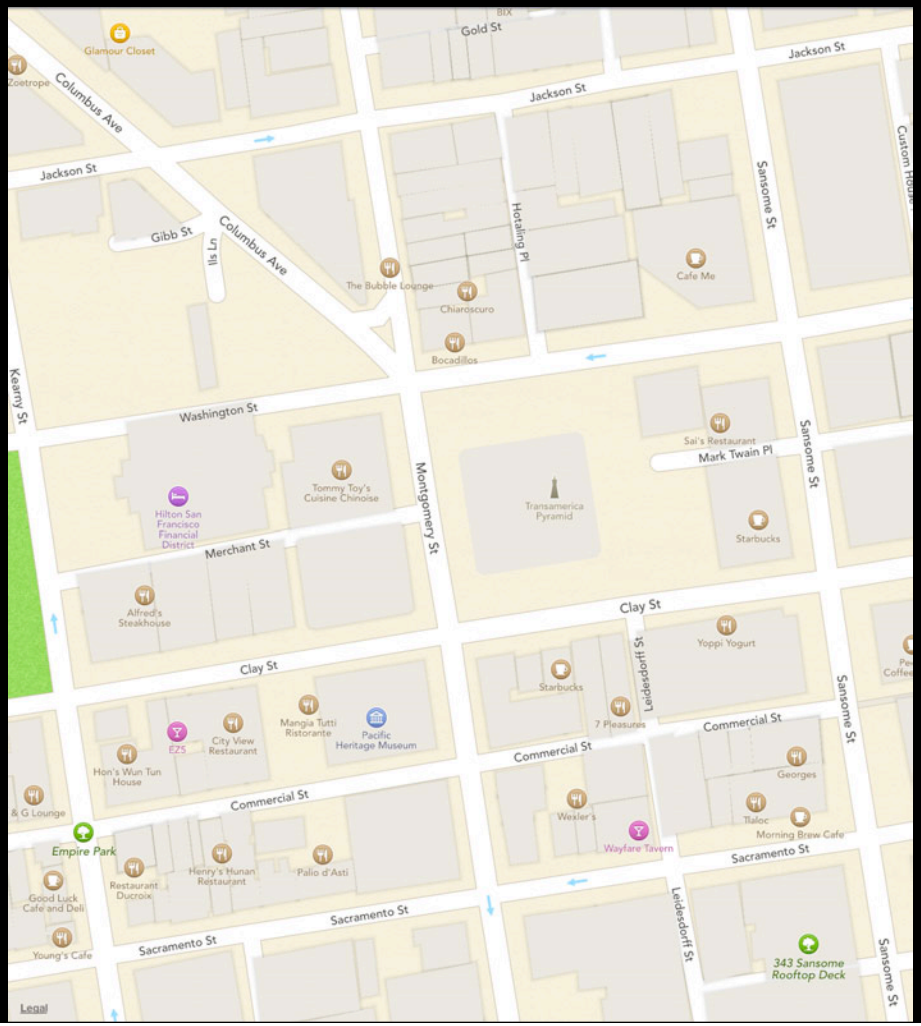
Map-Aided Location

- Location and course
- Driving
- Vector map data
- Improve if possible

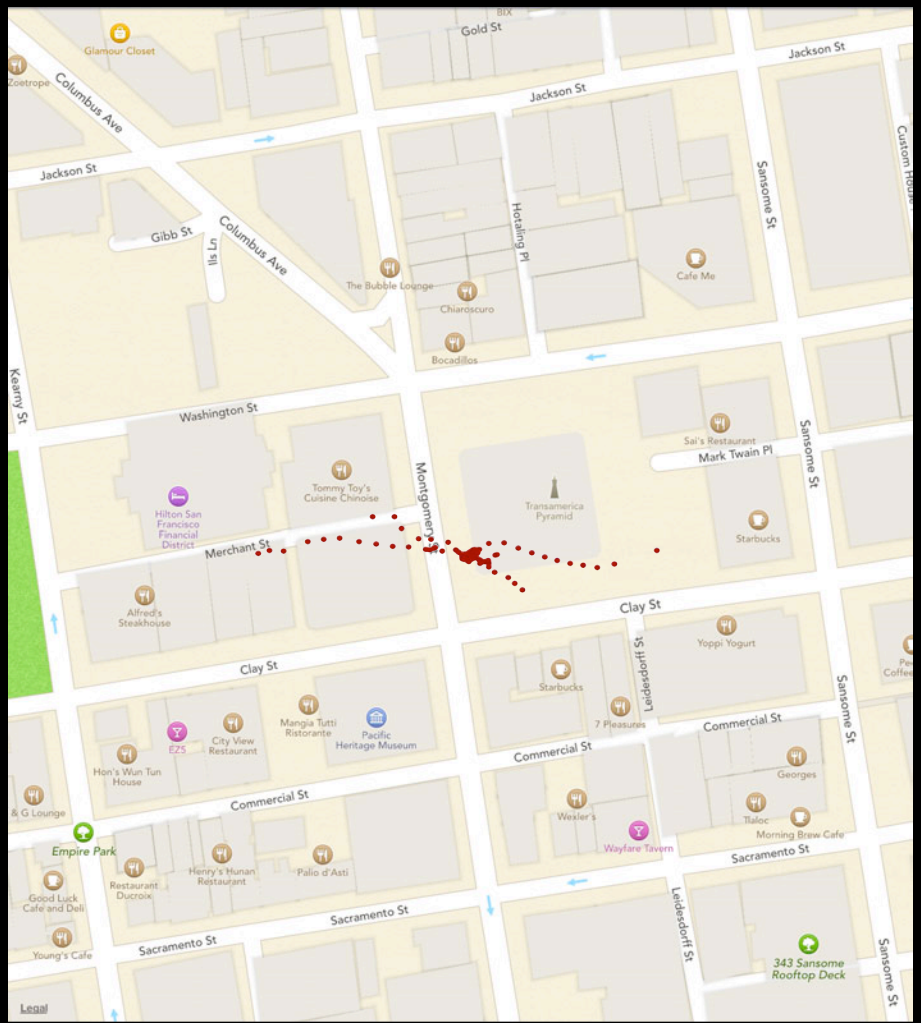


Improved Turns

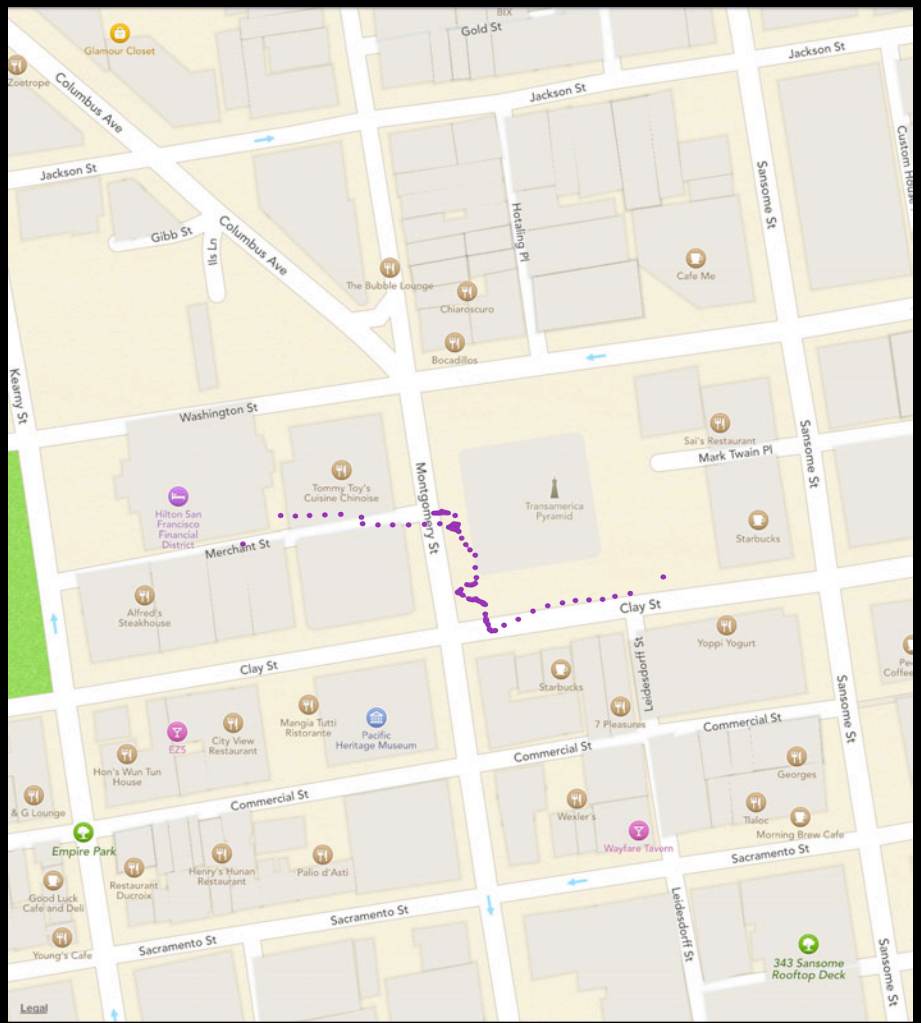
Improved Turns



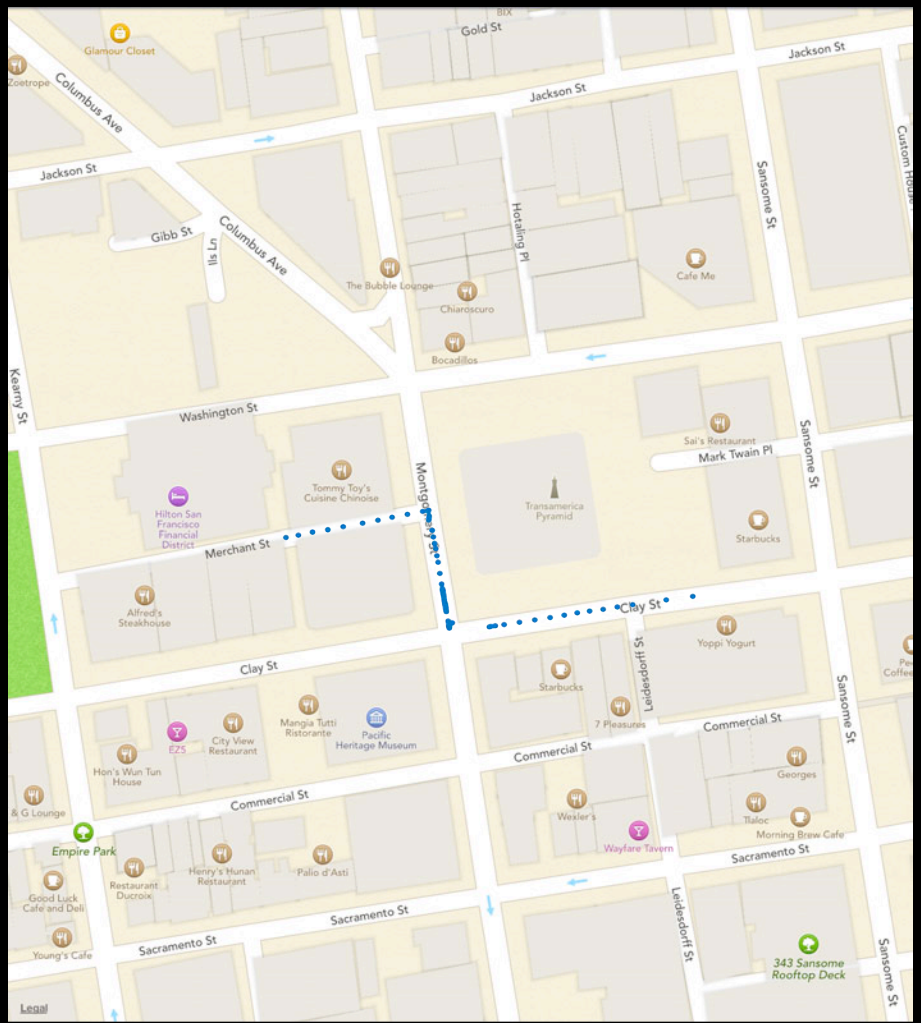
Improved Turns



Improved Turns

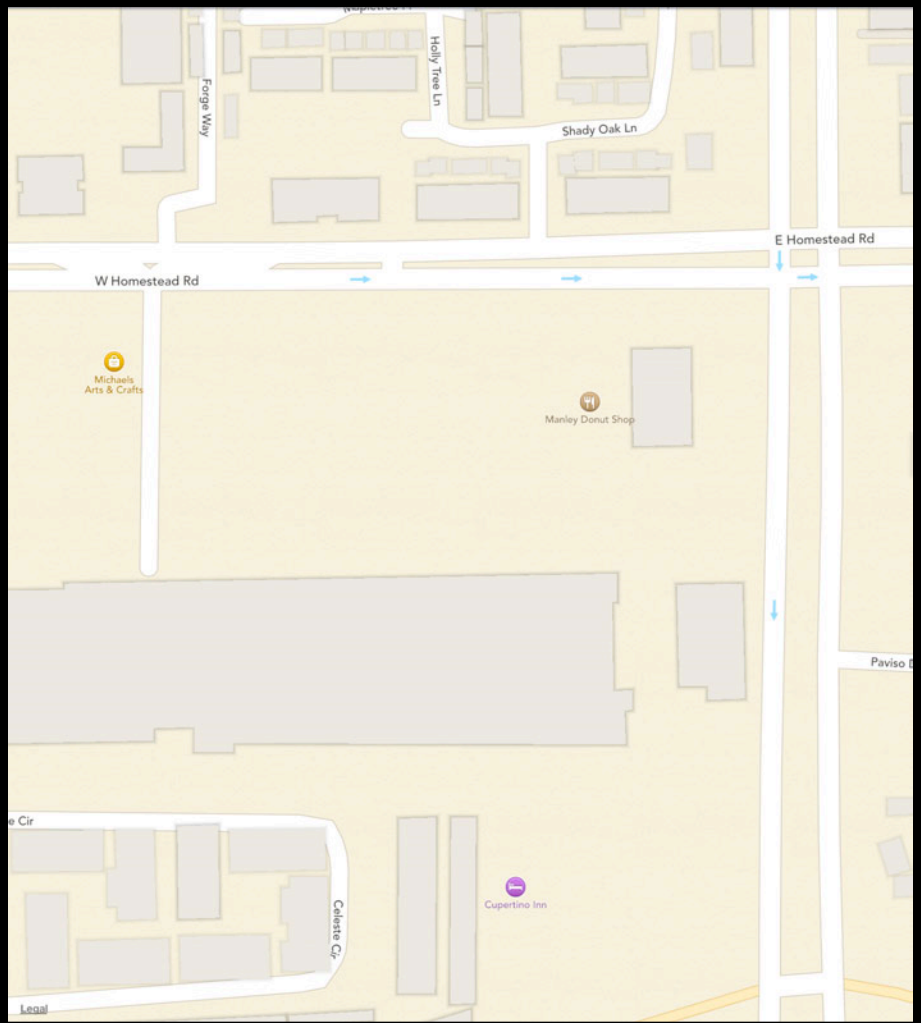


Improved Turns

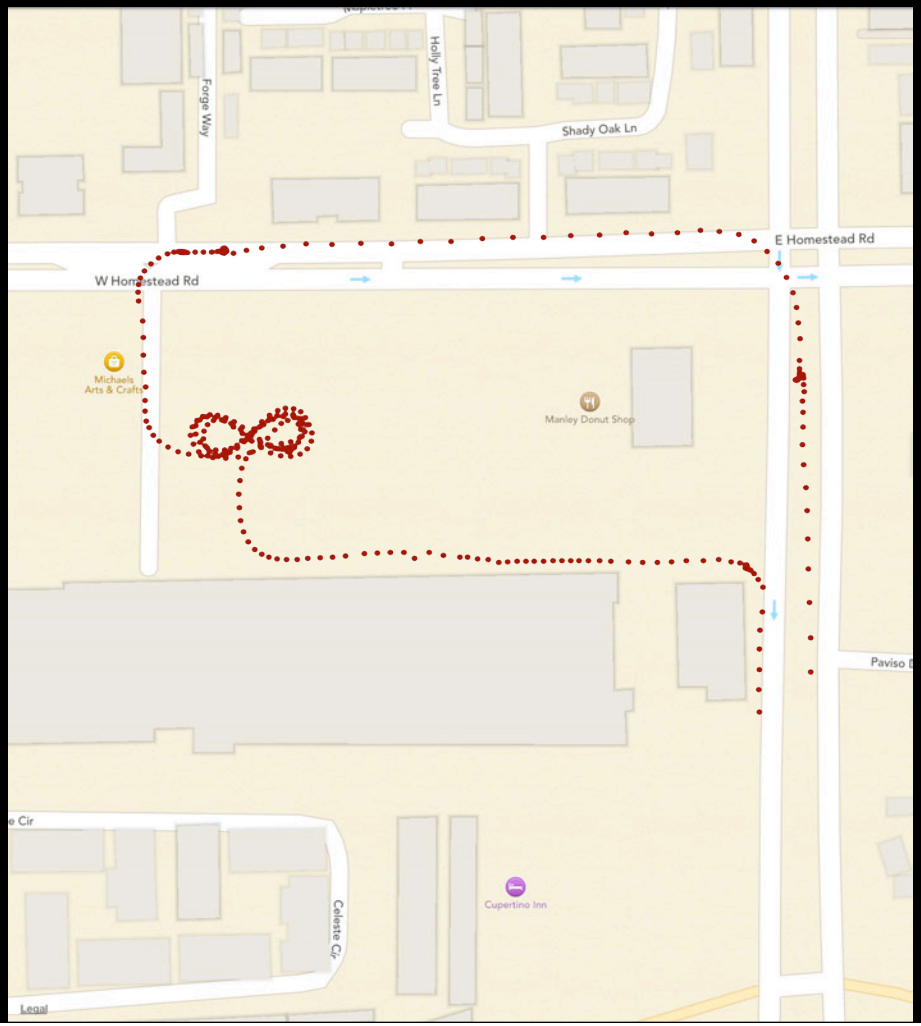


Do No Harm

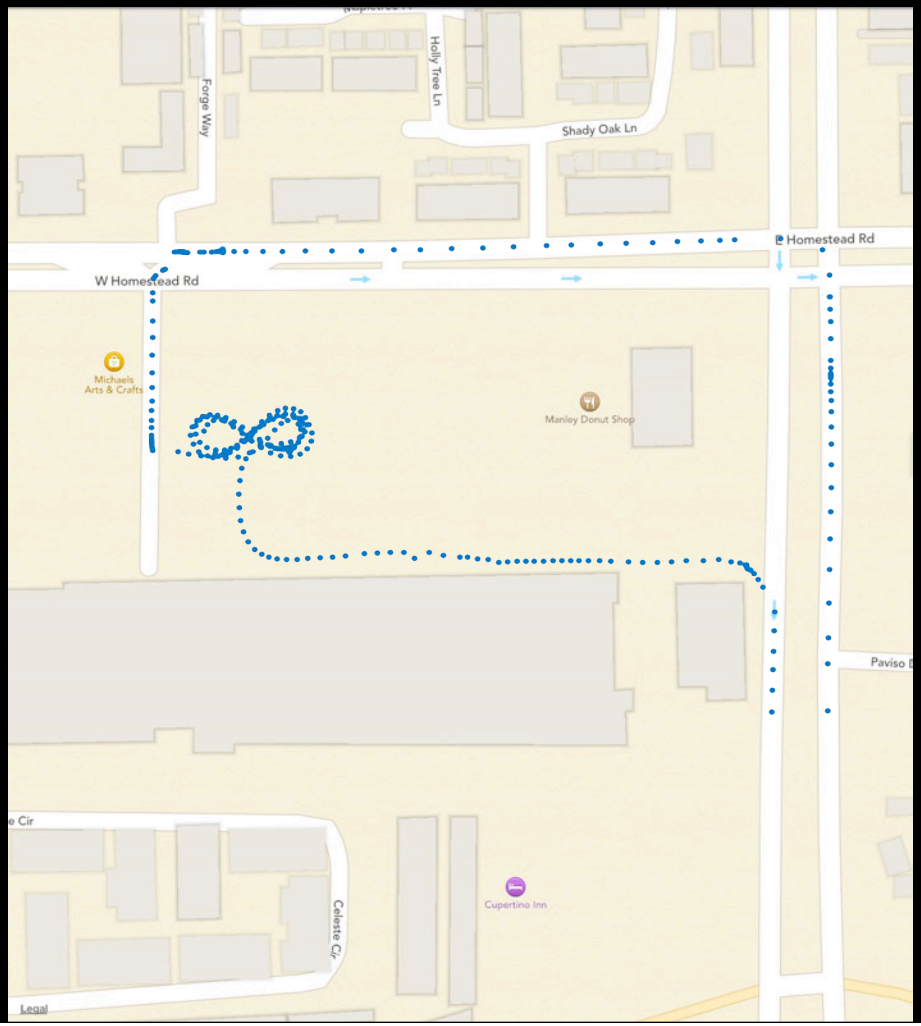
Do No Harm


Holly Tree Ln
Forge Way
Shady Oak Ln
E Homestead Rd
W Homestead Rd
Michaels
Arts & Crafts
Manley Donut Shop
Cupertino Inn
Celeste Cir
Legal

Do No Harm


Forge Way
Holly Tree Ln
Shady Oak Ln
E Homestead Rd
W Homestead Rd
Michaels
Arts & Crafts
Manley Donut Shop
Cupertino Inn
Celeste Cir
Legal

Do No Harm


Forge Way
Holly Tree Ln
Shady Oak Ln
E Homestead Rd
W Homestead Rd
Michaels
Arts & Crafts
Manley Donut Shop
Cupertino Inn
Celeste Cir
Legal

API Availability on OS X

API Availability on OS X

- Geocoding

API Availability on OS X

- Geocoding
- Region monitoring

Reverse Geocoding

Transforms a coordinate into an address

Reverse Geocoding

Transforms a coordinate into an address

37.78338, -122.403354

Reverse Geocoding

Transforms a coordinate into an address

800 Howard St.
San Francisco, CA

37.78338, -122.403354

Forward Geocoding

Transforms an address into a coordinate

Forward Geocoding

Transforms an address into a coordinate

800 Howard St.
San Francisco, CA

Forward Geocoding

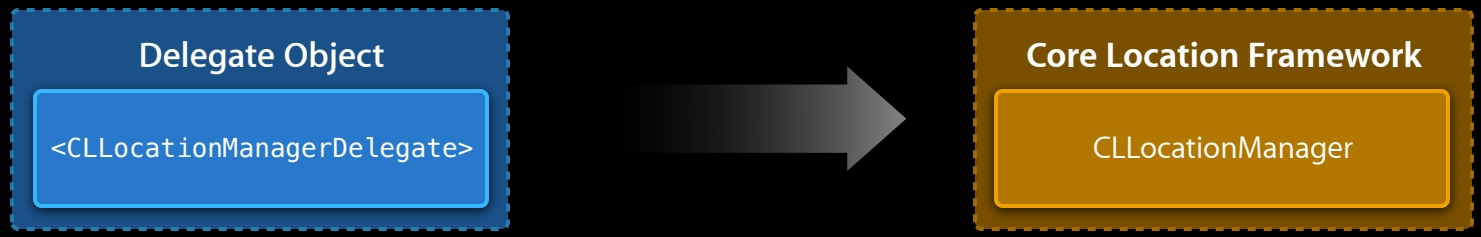
Transforms an address into a coordinate

800 Howard St.
San Francisco, CA

37.78338, -122.403354

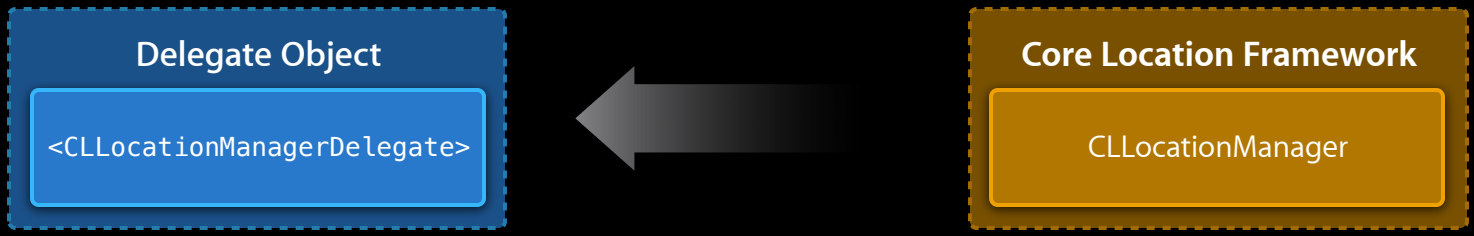
Starting Region Monitoring

```
CLLocationCoordinate2D coord = CLLocationCoordinate2DMake(37.332426, -122.030404);
CLRegion *region = [[CLRegion alloc] initCircularRegionWithCenter:coord
                                                           radius:100.0
                                                       identifier:@"Apple Inc."];
```




```
startMonitoringForRegion:(CLRegion *)region;
```

Receiving Region Monitoring Events




```
locationManager:didEnterRegion:

locationManager:didExitRegion:
```

Reminders on OS X

Example

Example

```
- (void)installReminder
{
    NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
    CLGeocoder *geocoder = [CLGeocoder new];
    [geocoder geocodeAddressString:address
                 completionHandler: ^(NSArray *placemarks, NSError *error){
        [self.locationManager startMonitoringForRegion:[placemark region]];
    }];
}
- (void)locationManager:(CLLocationManager *)manager
          didEnterRegion:(CLRegion *)region
{
    NSAlert *alert = [NSAlert new];
    [alert setMessageText:@"Welcome to the Loop!"];
    [alert runModal];
}
```

Example

```
- (void)installReminder
{
    NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
    CLGeocoder *geocoder = [CLGeocoder new];
    [geocoder geocodeAddressString:address
                 completionHandler: ^(NSArray *placemarks, NSError *error){
        [self.locationManager startMonitoringForRegion:[placemark region]];
    }];
}
- (void)locationManager:(CLLocationManager *)manager
          didEnterRegion:(CLRegion *)region
{
    NSAlert *alert = [NSAlert new];
    [alert setMessageText:@"Welcome to the Loop!"];
    [alert runModal];
}
```

Example

```
- (void)installReminder
{
    NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
    CLGeocoder *geocoder = [CLGeocoder new];
    [geocoder geocodeAddressString:address
                 completionHandler: ^(NSArray *placemarks, NSError *error){
        [self.locationManager startMonitoringForRegion:[placemark region]];
    }];
}
- (void)locationManager:(CLLocationManager *)manager
          didEnterRegion:(CLRegion *)region
{
    NSAlert *alert = [NSAlert new];
    [alert setMessageText:@"Welcome to the Loop!"];
    [alert runModal];
}
```

Example

```
- (void)installReminder
{
    NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
    CLGeocoder *geocoder = [CLGeocoder new];
    [geocoder geocodeAddressString:address
                 completionHandler: ^(NSArray *placemarks, NSError *error){
        [self.locationManager startMonitoringForRegion:[placemark region]];
    }];
}
- (void)locationManager:(CLLocationManager *)manager
         didEnterRegion:(CLRegion *)region
{
    NSAlert *alert = [NSAlert new];
    [alert setMessageText:@"Welcome to the Loop!"];
    [alert runModal];
}
```

Example

```
- (void)installReminder
{
   NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
   CLGeocoder *geocoder = [CLGeocoder new];
   [geocoder geocodeAddressString:address
                completionHandler: ^(NSArray *placemarks, NSError *error){
       [self.locationManager startMonitoringForRegion:[placemark region]];
   }];
}
- (void)locationManager:(CLLocationManager *)manager
         didEnterRegion:(CLRegion *)region
{
   NSAlert *alert = [NSAlert new];
   [alert setMessageText:@"Welcome to the Loop!"];
   [alert runModal];
}
```

Example

```
- (void)installReminder
{
   NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
   CLGeocoder *geocoder = [CLGeocoder new];
   [geocoder geocodeAddressString:address
                completionHandler: ^(NSArray *placemarks, NSError *error){
      [self.locationManager startMonitoringForRegion:[placemark region]];
   }];
}
- (void)locationManager:(CLLocationManager *)manager
         didEnterRegion:(CLRegion *)region
{
   NSAlert *alert = [NSAlert new];
   [alert setMessageText:@"Welcome to the Loop!"];
   [alert runModal];
}
```

Example

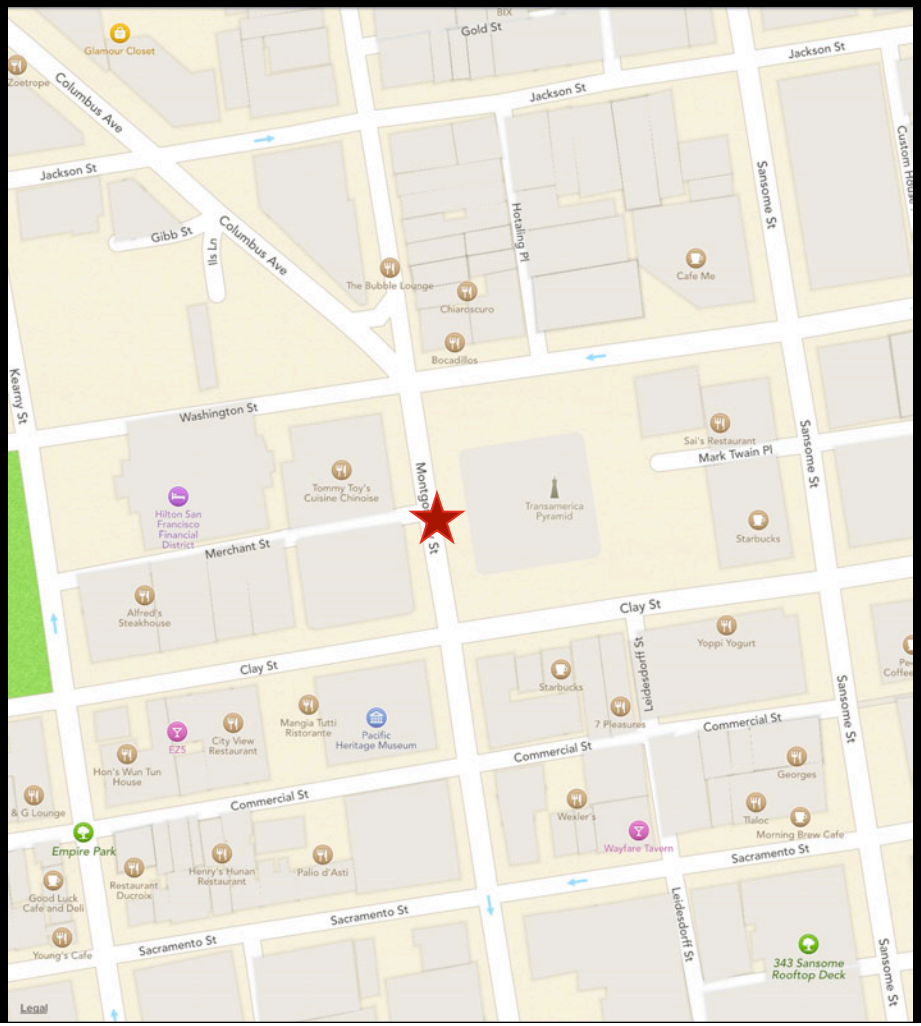
```
- (void)installReminder
{
   NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
   CLGeocoder *geocoder = [CLGeocoder new];
   [geocoder geocodeAddressString:address
                completionHandler: ^(NSArray *placemarks, NSError *error){
      [self.locationManager startMonitoringForRegion:[placemark region]];
   }];
}
- (void)locationManager:(CLLocationManager *)manager
         didEnterRegion:(CLRegion *)region
{
   NSAlert *alert = [NSAlert new];
   [alert setMessageText:@"Welcome to the Loop!"];
   [alert runModal];
}
```

Example

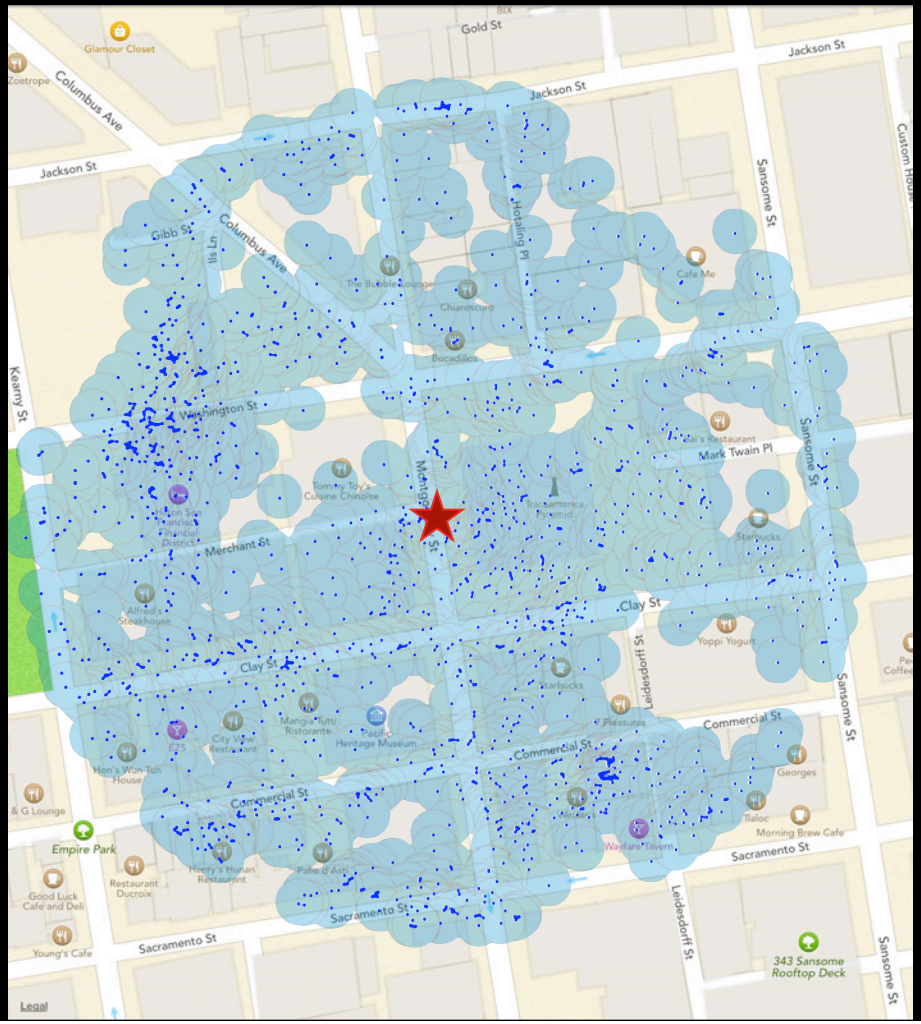
```
- (void)installReminder
{
   NSString *address = @"2 Infinite Loop, Cupertino, CA 95014";
   CLGeocoder *geocoder = [CLGeocoder new];
   [geocoder geocodeAddressString:address
                completionHandler: ^(NSArray *placemarks, NSError *error){
      [self.locationManager startMonitoringForRegion:[placemark region]];
   }];
}
- (void)locationManager:(CLLocationManager *)manager
         didEnterRegion:(CLRegion *)region
{
   NSAlert *alert = [NSAlert new];
   [alert setMessageText:@"Welcome to the Loop!"];
   [alert runModal];
}
```

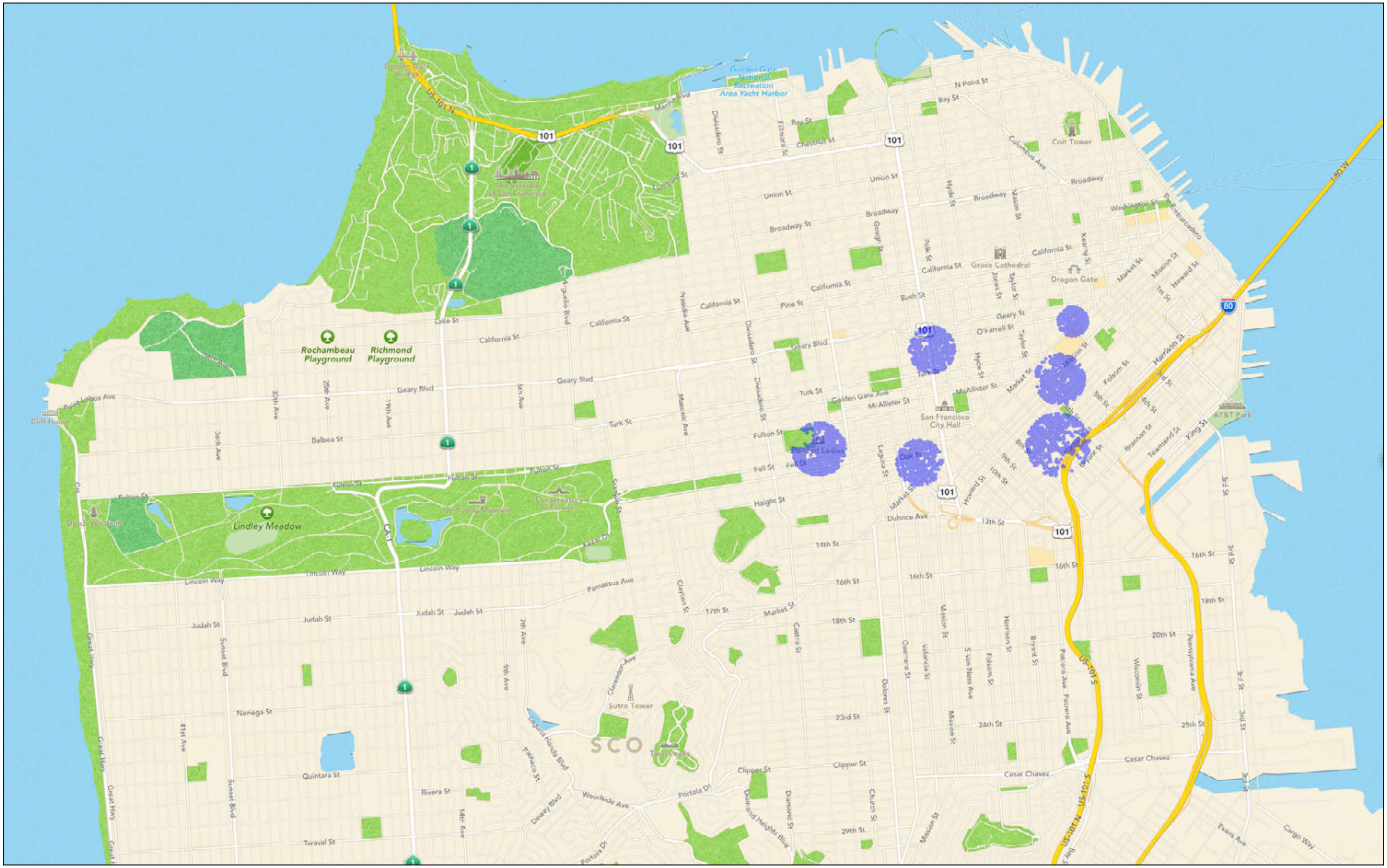
Wi-Fi Availability

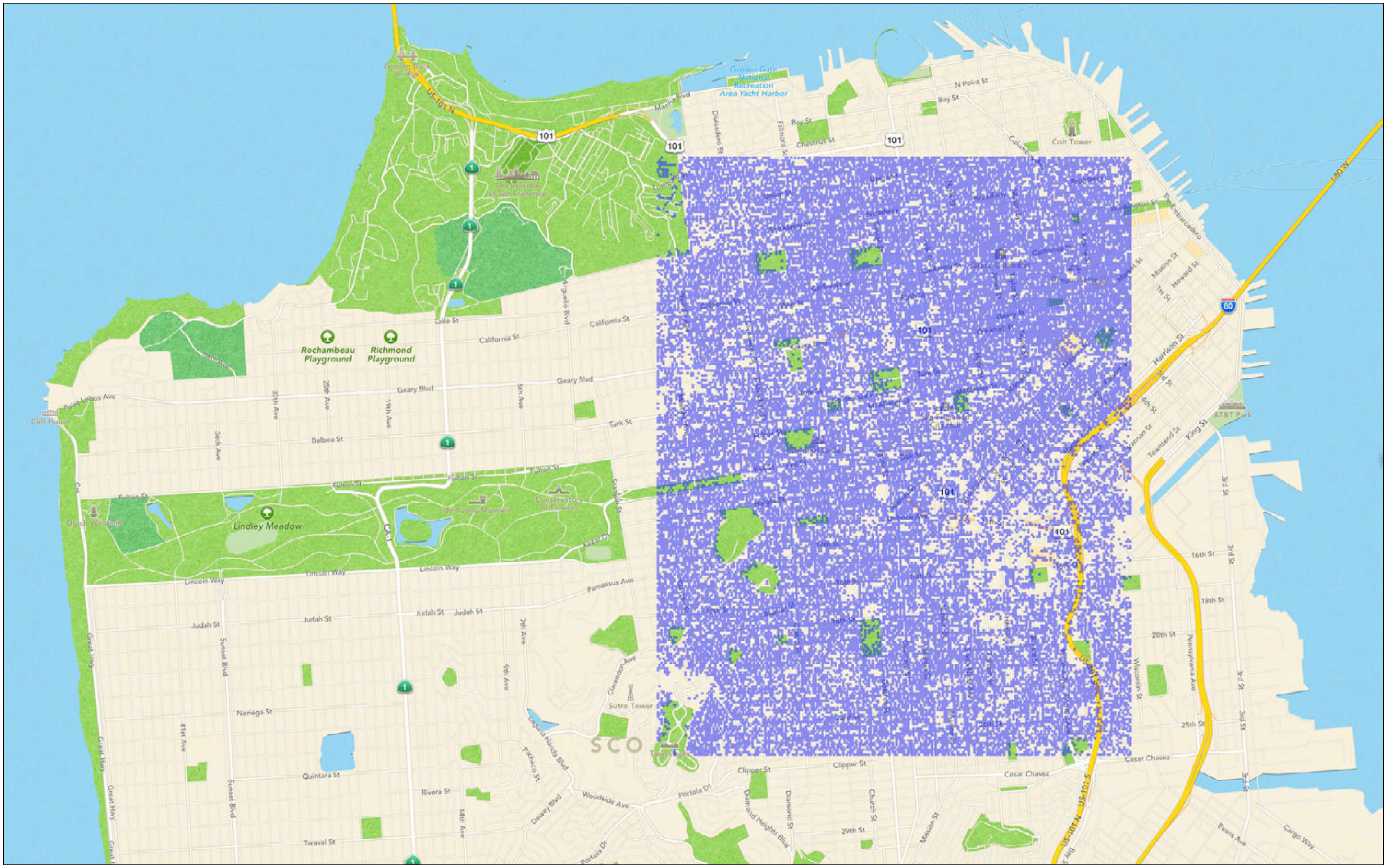
Wi-Fi Availability



Wi-Fi Availability




Golden Gate National Recreation Area Yacht Harbor
US-101 N
101
The Presidio of San Francisco
Rochambeau Playground
Richmond Playground
Lindley Meadow
de Young Museum
Conservatory of Flowers
Cliff House
Dutch Windmill
Sutro Tower
Coit Tower
Grace Cathedral
Dragon Gate
San Francisco City Hall
Painted Ladies
AT&T Park
N Point St
Bay St
Chestnut St
Union St
Broadway
California St
Pine St
Bush St
Geary St
Geary Blvd
O'Farrell St
Turk St
Golden Gate Ave
McAllister St
Fulton St
Fell St
Oak St
Haight St
Balboa St
Lincoln Way
Judah St
Noriega St
Quintara St
Taraval St
Rivera St
Lake St
Point Lobos Ave
Parnassus Ave
Market St
Mission St
Howard St
Folsom St
Harrison St
Brannan St
Townsend St
King St
Duboce Ave
13th St
14th St
16th St
17th St
18th St
20th St
23rd St
24th St
25th St
29th St
Clipper St
Cesar Chavez
Portola Dr
Woodside Ave
Laguna Honda Blvd
Clarendon Ave
Dewey Blvd
Diamond Heights Blvd
Divisadero St
Fillmore St
Gough St
Polk St
Hyde St
Jones St
Taylor St
Mason St
Kearny St
Columbus Ave
Washington St
The Embarcadero
Laguna St
Castro St
Church St
Dolores St
Guerrero St
Valencia St
S Van Ness Ave
Bryant St
Potrero Ave
Wisconsin St
Pennsylvania Ave
3rd St
Evans Ave
Cargo Way
Great Hwy
Sunset Blvd
Arguello Blvd
Masonic Ave
Presidio Ave
Stanyan St
Kezar Dr
30th Ave
25th Ave
19th Ave
41st Ave
36th Ave
14th Ave
9th Ave
7th Ave
6th Ave
Marina Blvd
Lombard St
US-101 S
I-80 W
80
SCO



US-101 N
101
Golden Gate National Recreation Area Yacht Harbor
Marina Blvd
Divisadero St
Fillmore St
Chestnut St
Bay St
N Point St
Coit Tower
The Presidio of San Francisco
Lake St
California St
Arguello Blvd
Rochambeau Playground
Richmond Playground
Geary Blvd
Point Lobos Ave
Cliff House
30th Ave
25th Ave
19th Ave
6th Ave
Turk St
Balboa St
36th Ave
Fulton St
Lindley Meadow
Dutch Windmill
de Young Museum
Conservatory of Flowers
Kezar Dr
Stanyan St
Lincoln Way
Parnassus Ave
Judah St
Sunset Blvd
7th Ave
9th Ave
Clarendon Ave
Sutro Tower
Great Hwy
Noriega St
41st Ave
Quintara St
Rivera St
14th Ave
Laguna Honda Blvd
Pacheco St
Dewey Blvd
Woodside Ave
Portola Dr
Taraval St
Clipper St
Diamond Heights Blvd
Diamond St
Church St
29th St
Mission St
Cesar Chavez
US-101 N
US-101 S
Harrison St
Mission St
Howard St
1st St
Townsend St
King St
AT&T Park
I-80 W
80
3rd St
16th St
18th St
20th St
Pennsylvania Ave
Wisconsin St
25th St
Evans Ave
Cargo Way


10× More Access Points


10× More Access Points
60× More Coverage Area

Wi-Fi Tile Advantages

Wi-Fi Tile Advantages

- Instant locations (Passbook, Weather, etc.)

Wi-Fi Tile Advantages

- Instant locations (Passbook, Weather, etc.)
- Improved availability on Wi-Fi–only devices

Wi-Fi Tile Advantages

- Instant locations (Passbook, Weather, etc.)
- Improved availability on Wi-Fi–only devices
  - Region monitoring

Wi-Fi Tile Advantages

- Instant locations (Passbook, Weather, etc.)
- Improved availability on Wi-Fi–only devices
  - Region monitoring
  - Photo geo-tagging

Wi-Fi Tile Advantages

- Instant locations (Passbook, Weather, etc.)
- Improved availability on Wi-Fi–only devices
  - Region monitoring
  - Photo geo-tagging
- International 3G roaming

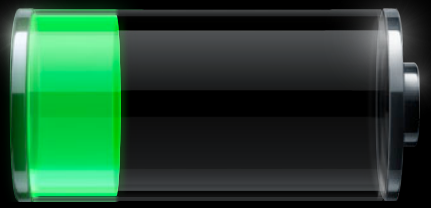
Wi-Fi Tile Advantages

- Instant locations (Passbook, Weather, etc.)
- Improved availability on Wi-Fi–only devices
  - Region monitoring
  - Photo geo-tagging
- International 3G roaming
- 10x better data usage

API Changes

Power

Power



Power

Power

The Problem

The Problem

- App running indefinitely

The Problem

- App running indefinitely
- User not moving

The Problem

- App running indefinitely
- User not moving
- Location not available

The Problem

- App running indefinitely
- User not moving
- Location not available
- Activity stopped

AutoPause

AutoPause

- App becomes pausable when backgrounded

AutoPause

- App becomes pausable when backgrounded
- Pause criteria

AutoPause

- App becomes pausable when backgrounded
- Pause criteria
  - User not moving

AutoPause

- App becomes pausable when backgrounded
- Pause criteria
  - User not moving
  - Device cannot obtain a location fix

AutoPause

- App becomes pausable when backgrounded
- Pause criteria
  - User not moving
  - Device cannot obtain a location fix
  - User discontinues activity

AutoPause API

AutoPause API

- Enabled?

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
```

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
    CLActivityTypeVehicularNavigation
```

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
    CLActivityTypeVehicularNavigation
    CLActivityTypeFitness
```

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
    CLActivityTypeVehicularNavigation
    CLActivityTypeFitness
    CLActivityTypeOther
```

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
    CLActivityTypeVehicularNavigation
    CLActivityTypeFitness
    CLActivityTypeOther
```

- Handle pause events

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
    CLActivityTypeVehicularNavigation
    CLActivityTypeFitness
    CLActivityTypeOther
```

- Handle pause events

```
– locationManagerDidPauseLocationUpdates:
```

AutoPause API

- Enabled?

```
@property BOOL pausesLocationUpdatesAutomatically
```

- Activity detection

```
@property CLActivityType activityType
    CLActivityTypeVehicularNavigation
    CLActivityTypeFitness
    CLActivityTypeOther
```

- Handle pause events

```
– locationManagerDidPauseLocationUpdates:
– locationManagerDidResumeLocationUpdates:
```

Demo

Mike Dal Santo
Software Engineer

How to Respond

How to Respond

- Take no action

How to Respond

- Take no action
- Notify the user

How to Respond

- Take no action
- Notify the user
- Stop location updates

Other Responses

Other Responses

- Disable AutoPause

Other Responses

- Disable AutoPause

```
self.manager.pausesLocationUpdatesAutomatically = NO;
```

Other Responses

- Disable AutoPause

```
self.manager.pausesLocationUpdatesAutomatically = NO;
```

- Restart location

Other Responses

- Disable AutoPause

```
self.manager.pausesLocationUpdatesAutomatically = NO;
```

- Restart location

```
[self.manager startUpdatingLocation];
```

Other Responses

- Disable AutoPause

```
self.manager.pausesLocationUpdatesAutomatically = NO;
```

- Restart location

```
[self.manager startUpdatingLocation];
```

- Resume yourself

Other Responses

- Disable AutoPause

```
self.manager.pausesLocationUpdatesAutomatically = NO;
```

- Restart location

```
[self.manager startUpdatingLocation];
```

- Resume yourself

```
[self.manager startMonitoringForRegion:currentLocation];
```

Handling Location Updates

Handling Location Updates

- New delegate method

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSArray *)locations
```

Handling Location Updates

- New delegate method
  - `- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations`
- Deprecated delegate method
  - `- (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation fromLocation:(CLLocation *)oldLocation`

Authorization

Help the user make an informed decision

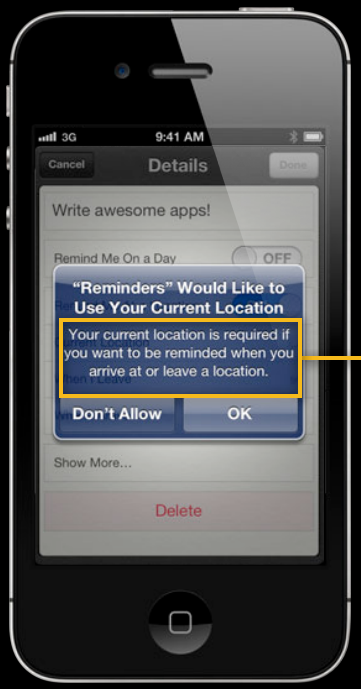
Authorization

Help the user make an informed decision

Permission Dialog
Shown once per application

Authorization

Help the user make an informed decision



Purpose property
Tell the user why your application needs to use location services

Authorization

Help the user make an informed decision

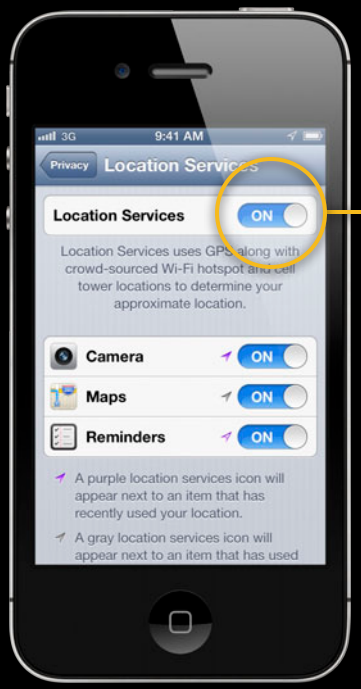
“Don’t Allow”
Plan for it

Authorization

The user is in control

Authorization

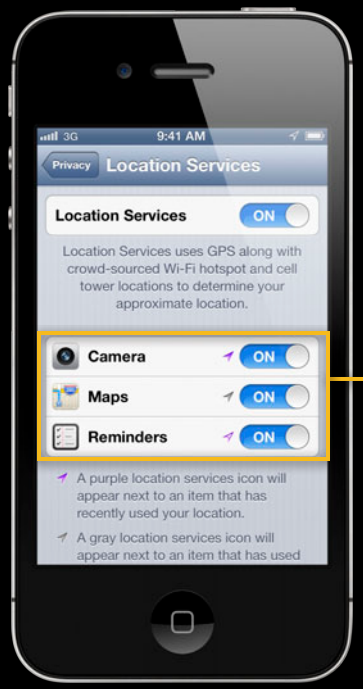
The user is in control



Enabled switch
Retrieve value from location manager class function

Authorization

The user is in control

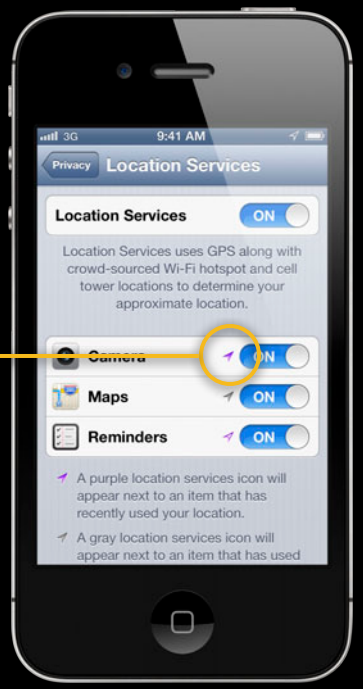


List of applications
Applications appear after requesting location services

Authorization

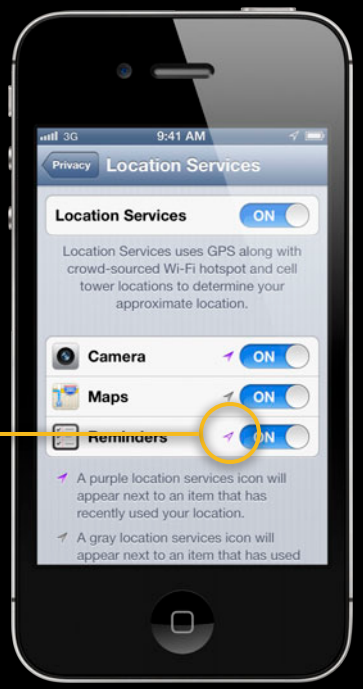
The user is in control

Location arrow
Displayed if the application has requested location within the last minute



Authorization

The user is in control

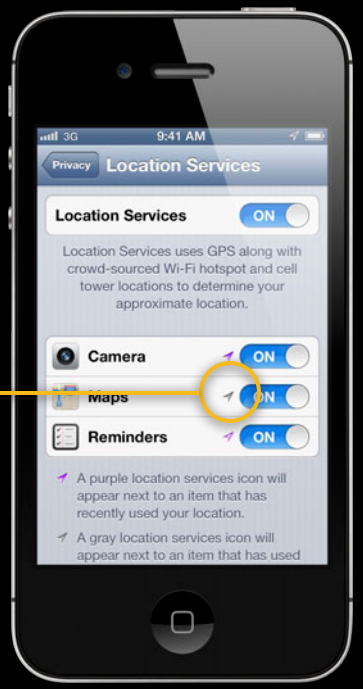


Location arrow
Displayed if the application is region monitoring

Authorization

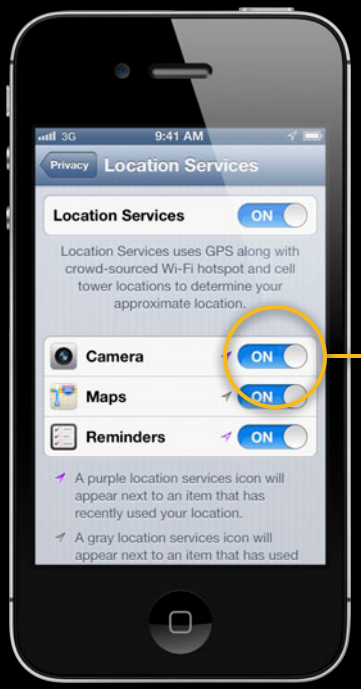
The user is in control

Location arrow
Displayed if the application has requested location in the last 24 hours



Authorization

The user is in control



Approval switch
Check approval status
using CLLocationManager

Authorization

The user is in control

Status bar icon
Visible when an application is using the user's location

Getting User Permission

Getting User Permission

Getting User Permission

- Standardized usage descriptions in Info.plist

```
NSLocationUsageDescription
```

Getting User Permission

- Standardized usage descriptions in Info.plist

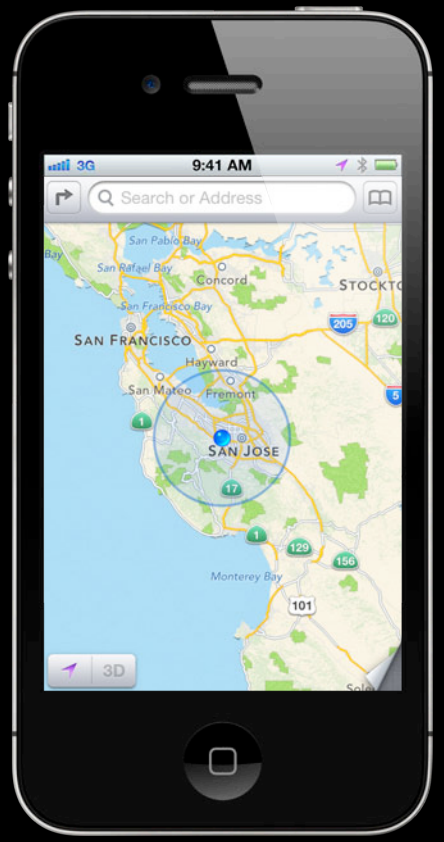
```
NSLocationUsageDescription
```

- Deprecated existing API

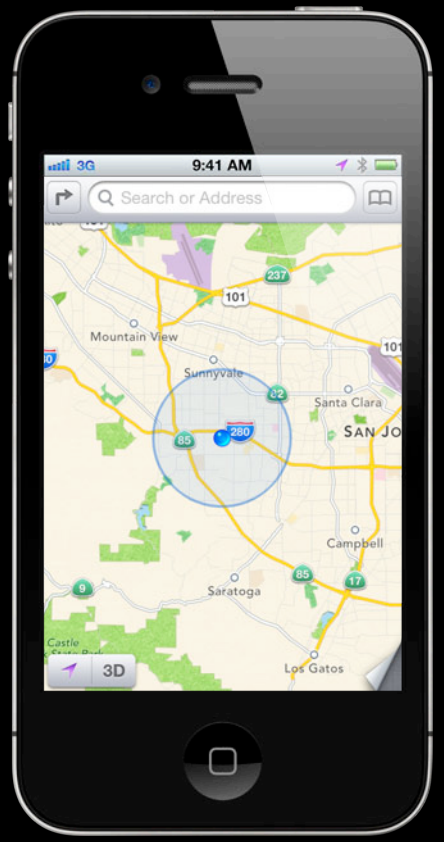
```
@property NSString *purpose
```

Tips, Tricks, and Myths

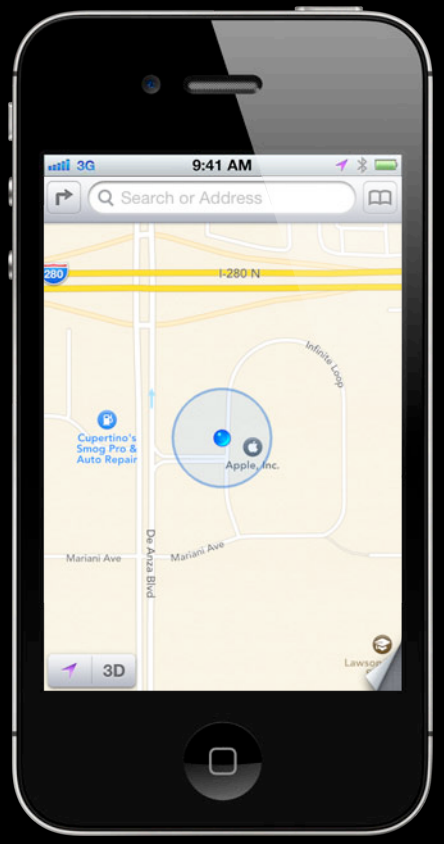
Myth: Maps Loads Location Faster



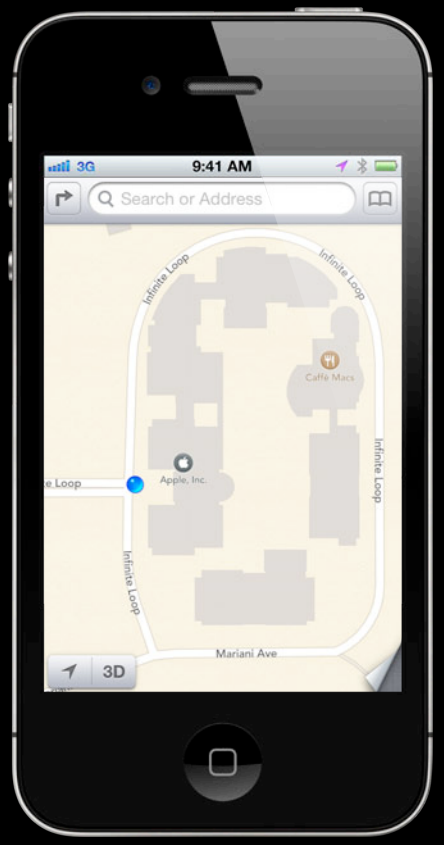
Myth: Maps Loads Location Faster



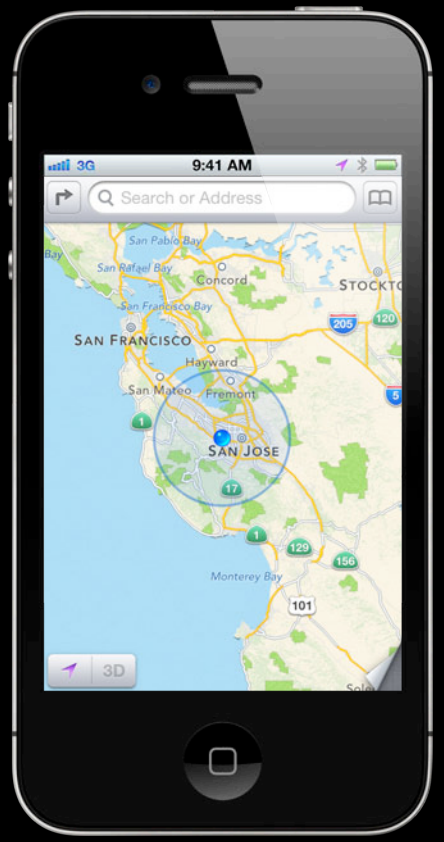
Myth: Maps Loads Location Faster



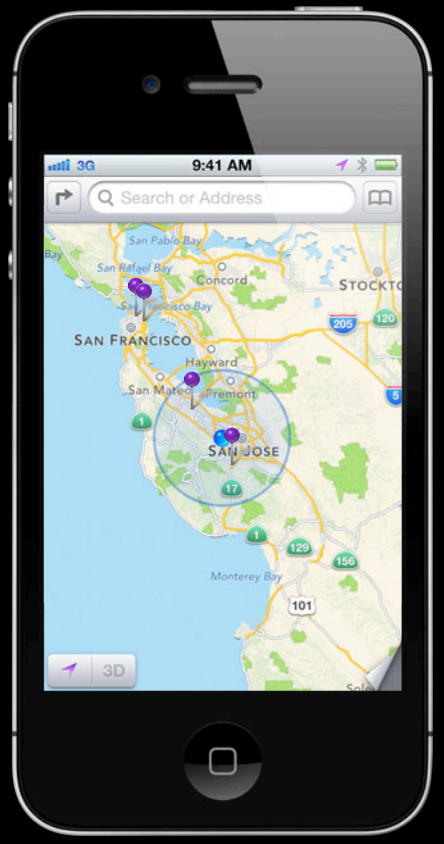
Myth: Maps Loads Location Faster



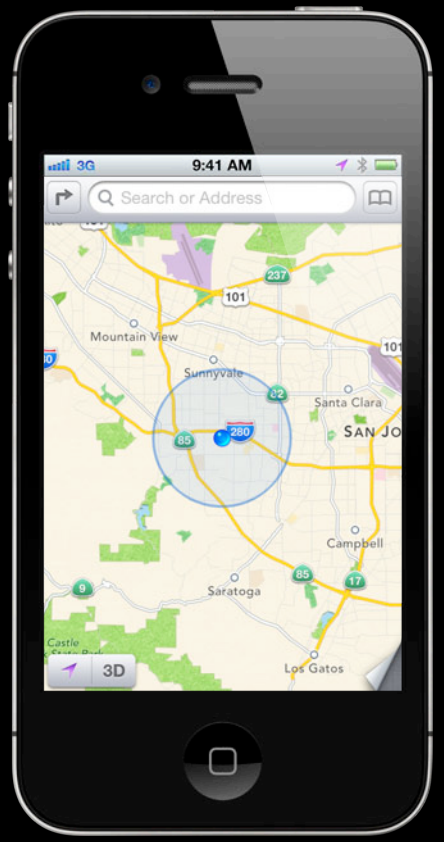
Myth: Maps Loads Location Faster



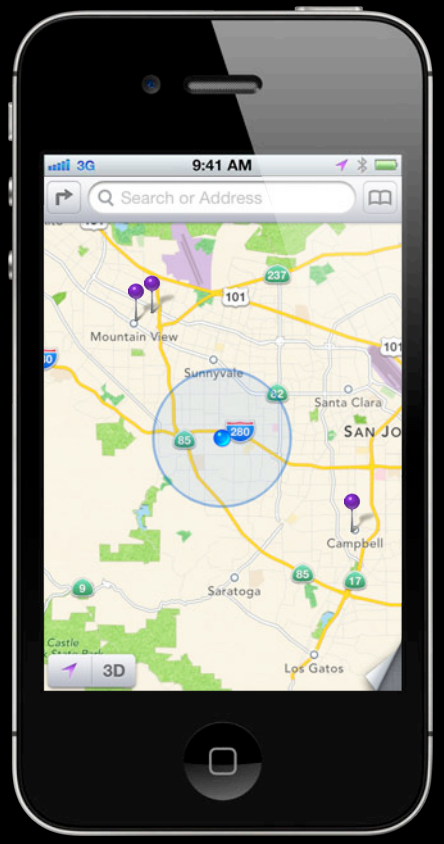
Myth: Maps Loads Location Faster



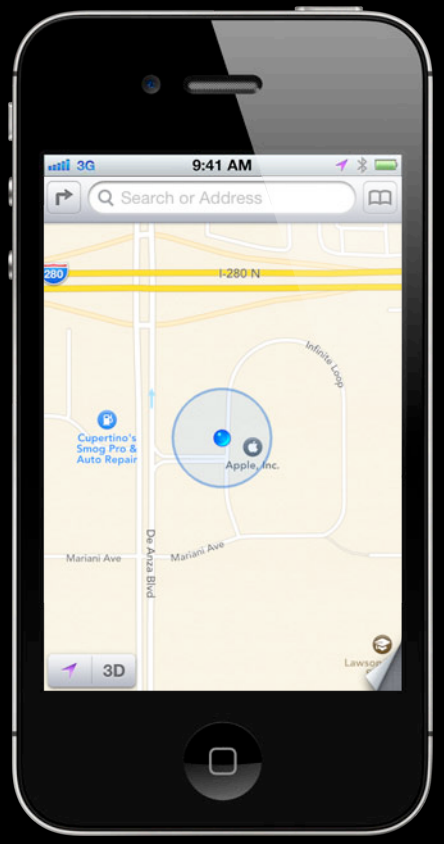
Myth: Maps Loads Location Faster



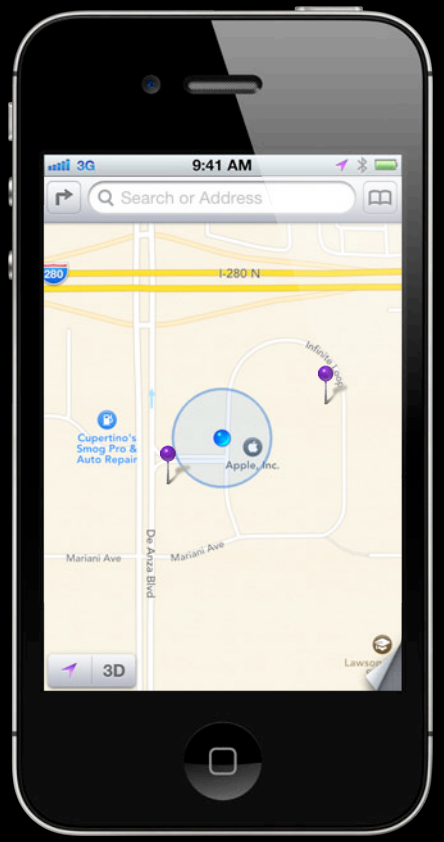
Myth: Maps Loads Location Faster



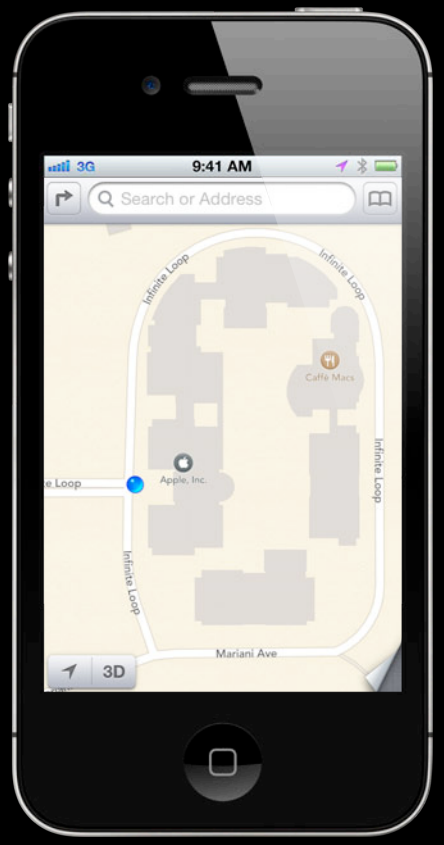
Myth: Maps Loads Location Faster



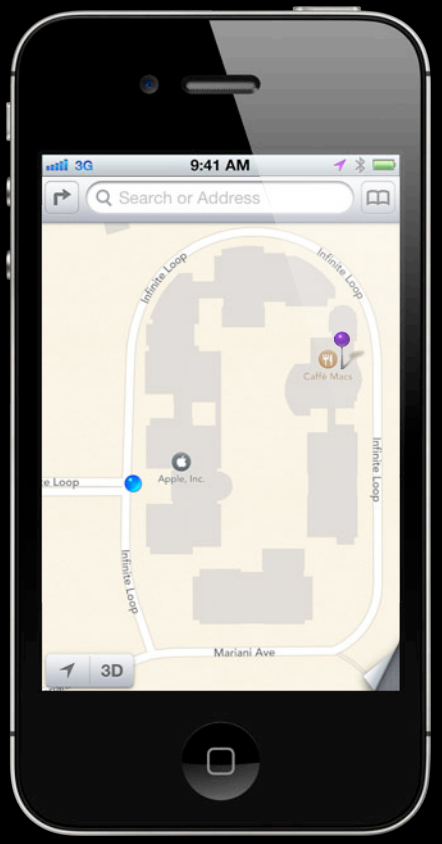
Myth: Maps Loads Location Faster



Myth: Maps Loads Location Faster

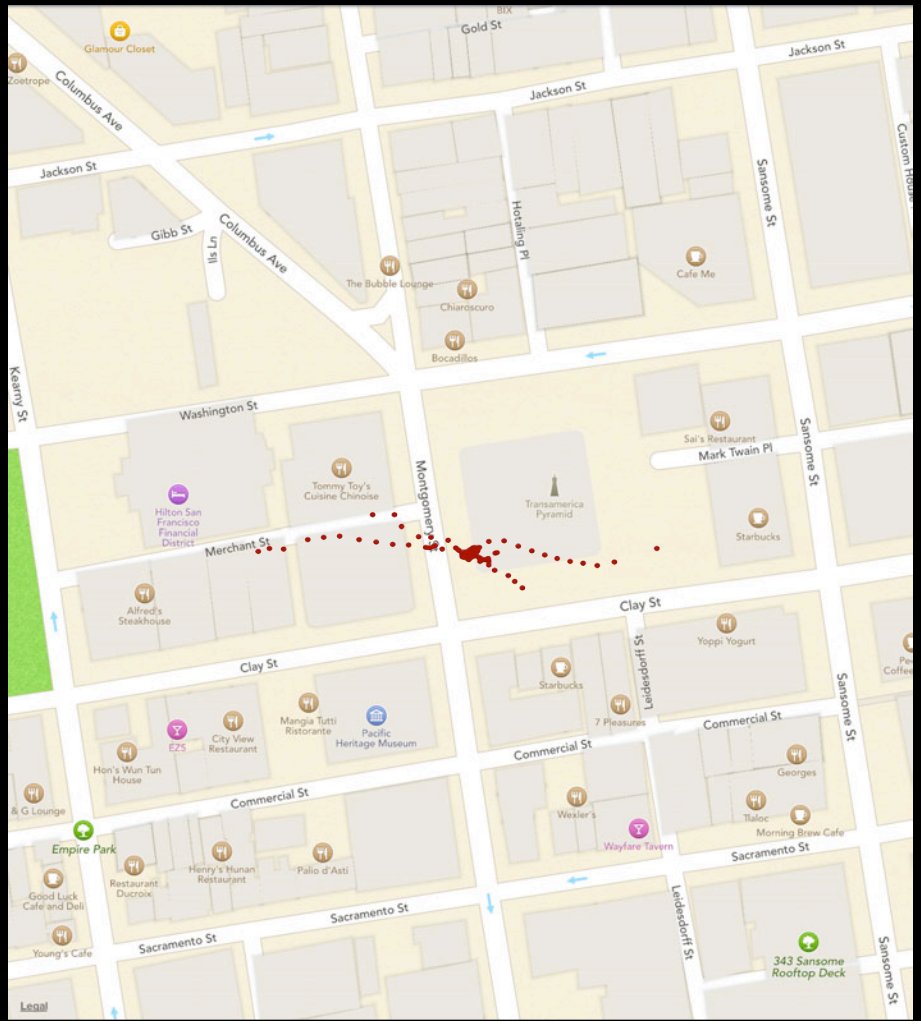


Myth: Maps Loads Location Faster

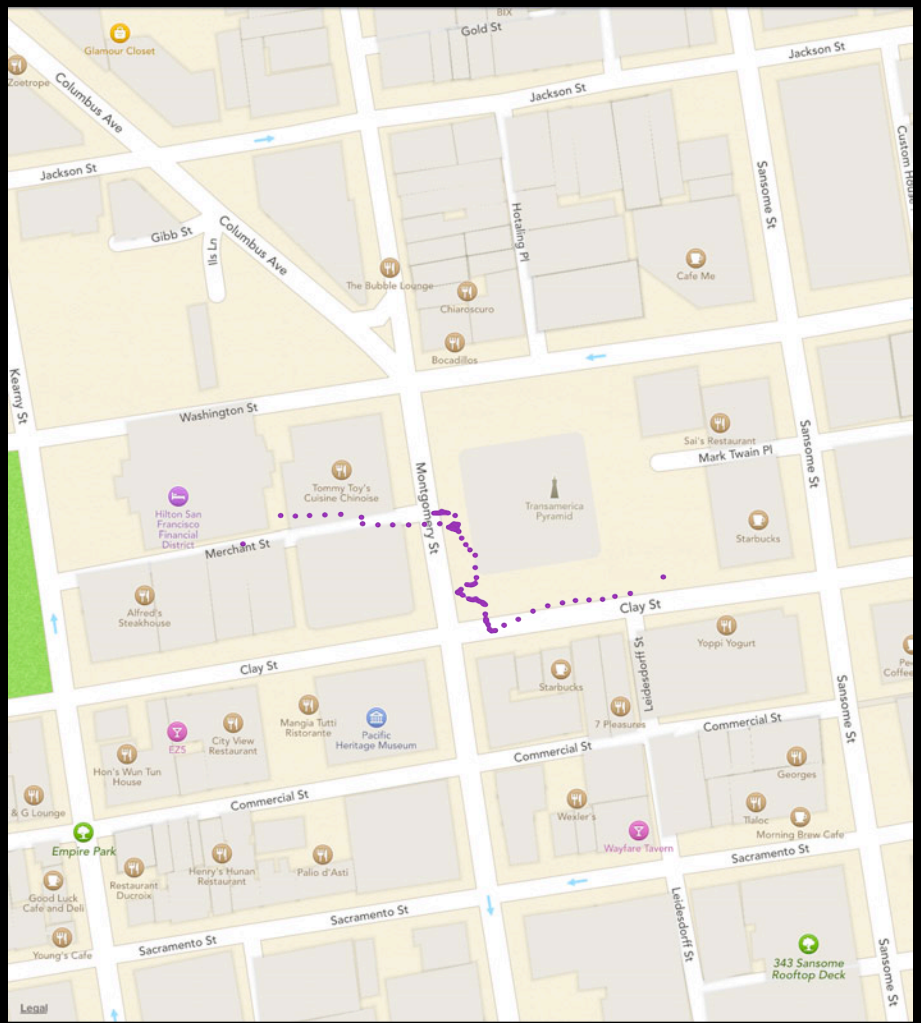


Myth: Turn Off Wi-Fi to Improve Accuracy

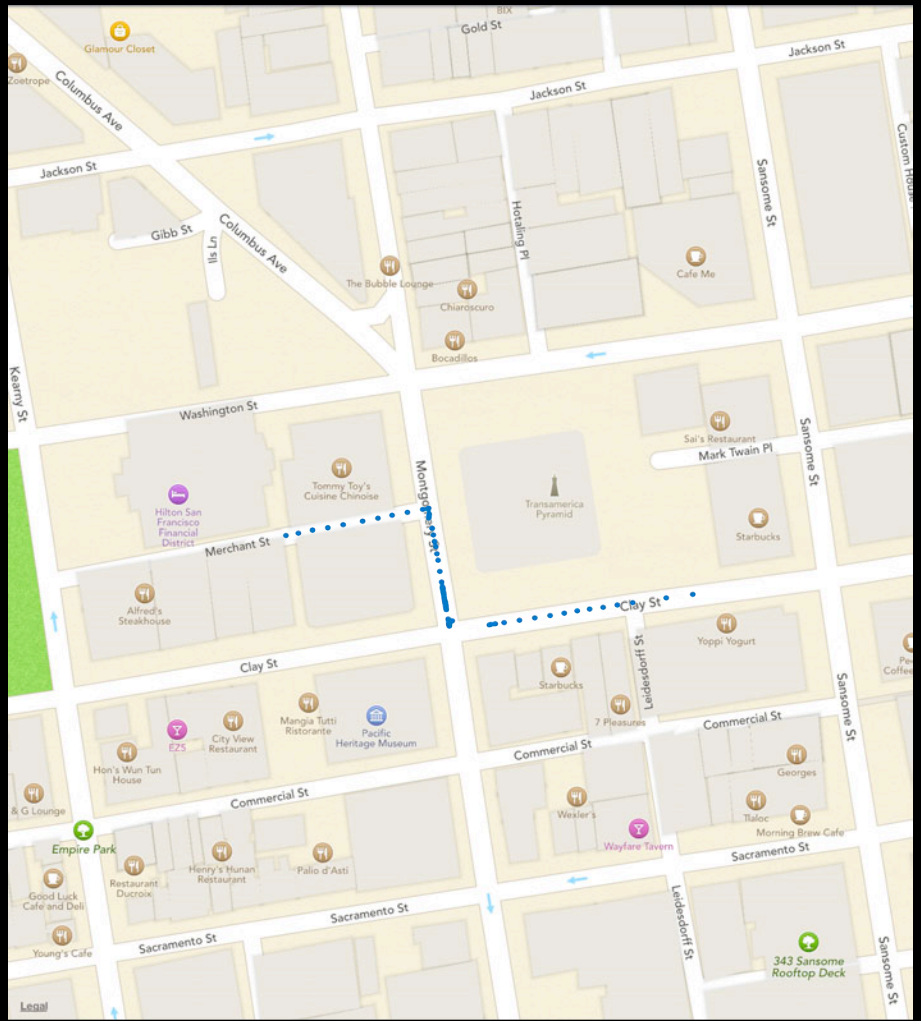
Myth: Turn Off Wi-Fi to Improve Accuracy



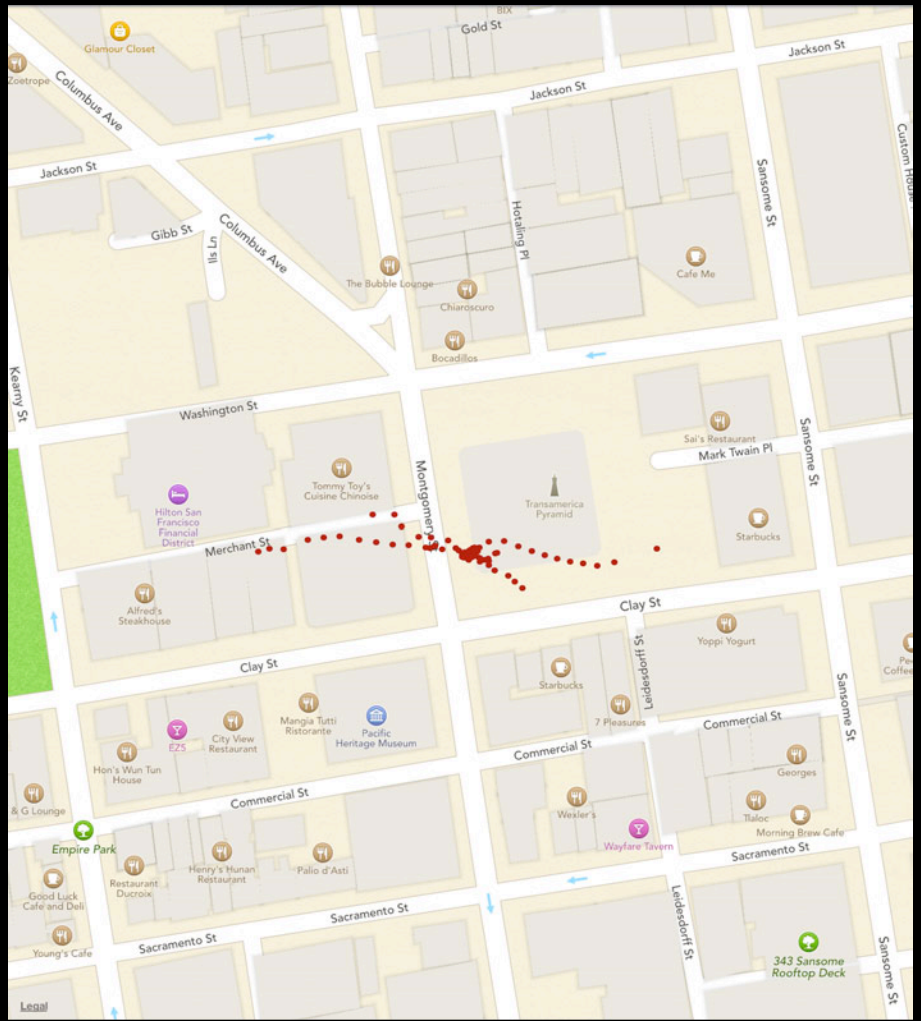
Myth: Turn Off Wi-Fi to Improve Accuracy



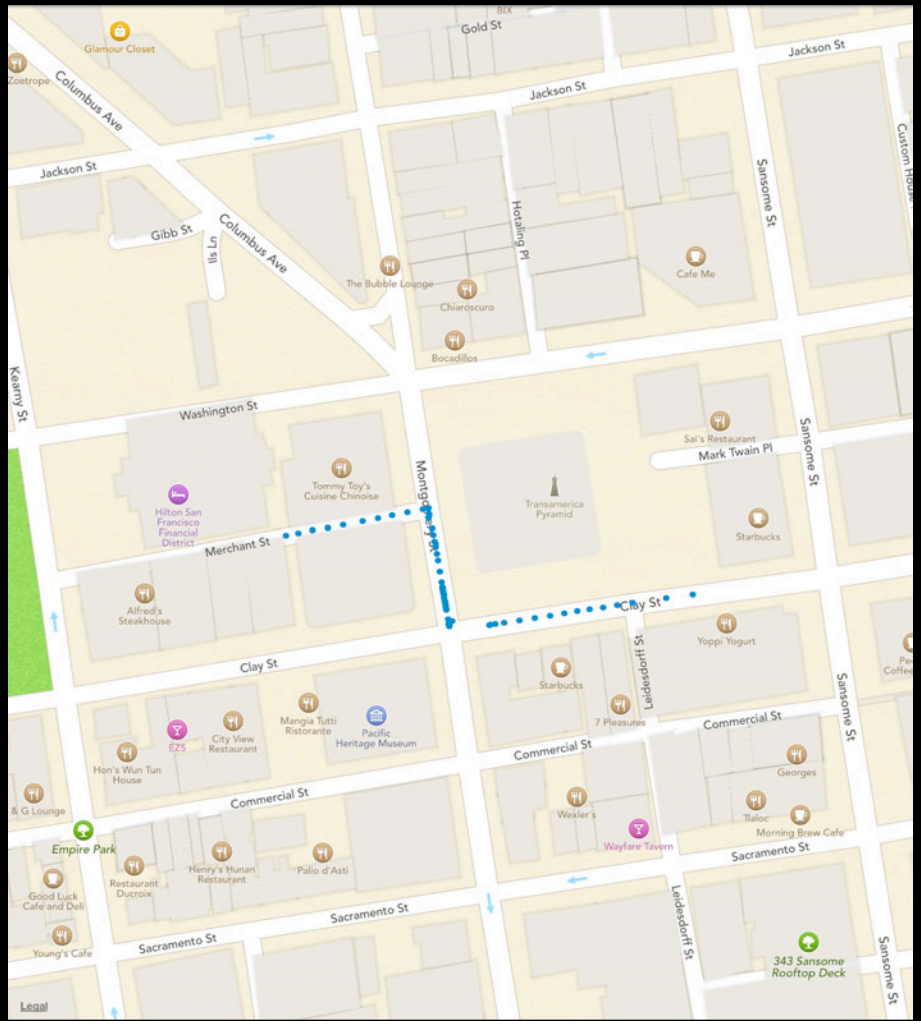
Myth: Turn Off Wi-Fi to Improve Accuracy

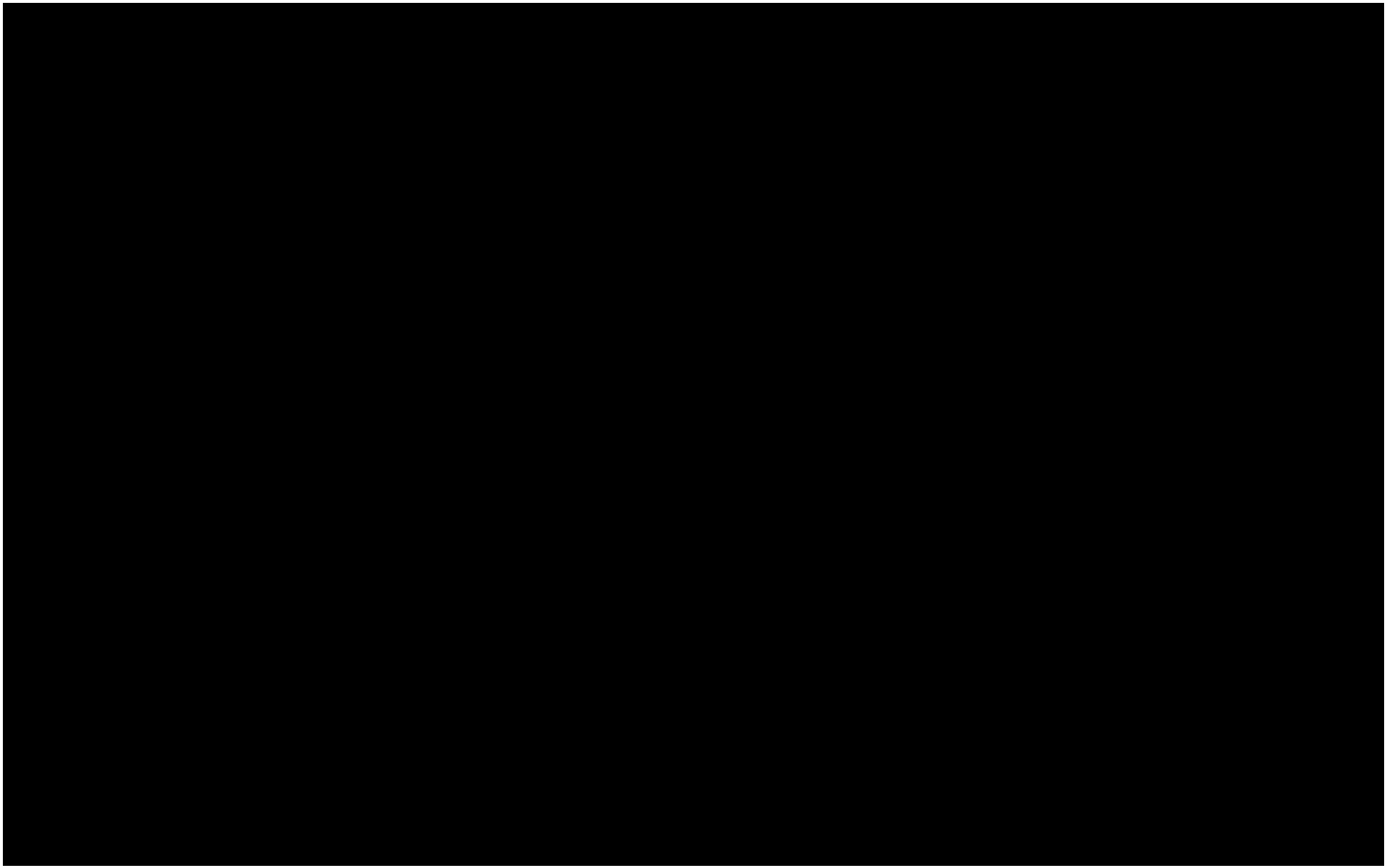


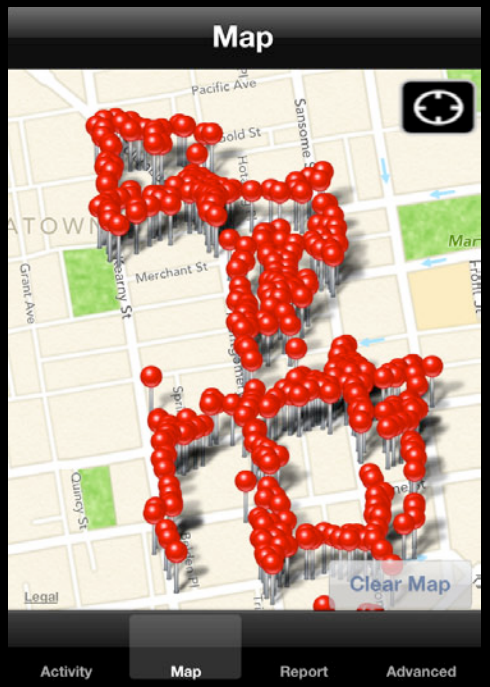
Myth: Turn Off Wi-Fi to Improve Accuracy

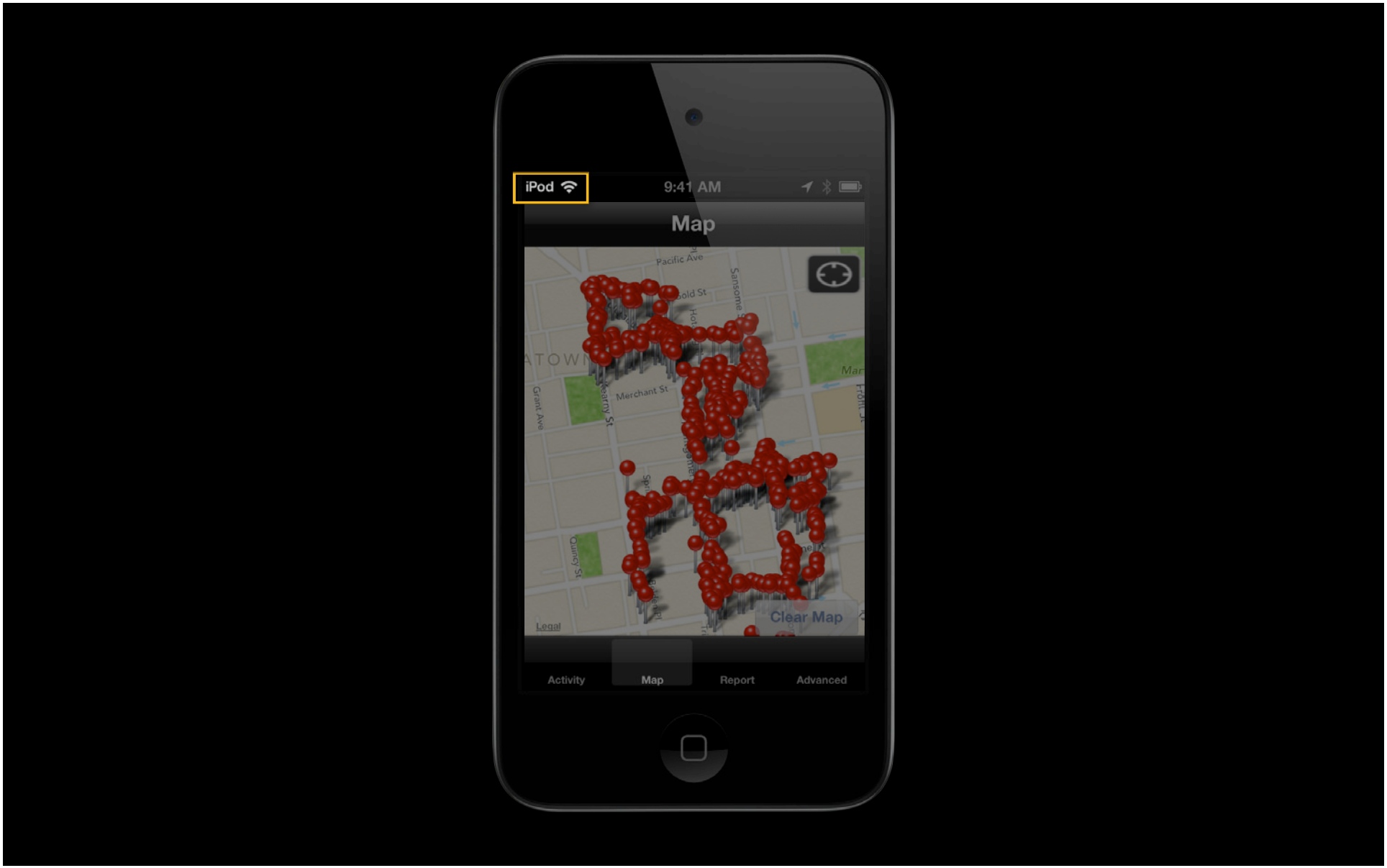


Myth: Turn Off Wi-Fi to Improve Accuracy






Map
Pacific Ave
Merchant St
Grant Ave
Kearny St
Quincy St
Legal
Clear Map
Activity
Map
Report
Advanced


iPod
9:41 AM
Map
Pacific Ave
Merchant St
Grant Ave
Kearny St
Quincy St
Legal
Clear Map
Activity
Map
Report
Advanced

Support an Offline User Experience

Support an Offline User Experience

- Support poor coverage areas

Support an Offline User Experience

- Support poor coverage areas
- Minimize data

Support an Offline User Experience

- Support poor coverage areas
- Minimize data
- Save user battery

Support an Offline User Experience

- Support poor coverage areas
- Minimize data
- Save user battery
- Improve app performance

Support an Offline User Experience

- Support poor coverage areas
- Minimize data
- Save user battery
- Improve app performance
- Support iPod touches and Wi-Fi iPads

Support an Offline User Experience

- Support poor coverage areas
- Minimize data
- Save user battery
- Improve app performance
- Support iPod touches and Wi-Fi iPads
- Data Protection APIs

Reasons to Run in the Background

Reasons to Run in the Background

- Navigation

Reasons to Run in the Background

- Navigation
- Fitness

Reasons to Run in the Background

- Navigation
- Fitness
- Reminders

Reasons to Run in the Background

- Navigation
- Fitness
- Reminders
- Automatic check-in

Run All the Time

Run All the Time

```
<key>UIBackgroundModes</key>
<array>
    <string>location</string>
</array>
```

Run All the Time

```
<key>UIBackgroundModes</key>
<array>
    <string>location</string>
</array>
<key>Required background modes</key>
<array>
    <string>App registers for location updates</string>
</array>
```

Run All the Time

```
<key>UIBackgroundModes</key>
<array>
    <string>location</string>
</array>
<key>Required background modes</key>
<array>
    <string>App registers for location updates</string>
</array>
```

Run When Necessary

Run When Necessary

- Significant location changes

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

```
-startMonitoringForRegion:
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

```
-startMonitoringForRegion:
-stopMonitoringForRegion:
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

```
-startMonitoringForRegion:
-stopMonitoringForRegion:
-locationManager:didEnterRegion:
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

```
-startMonitoringForRegion:
-stopMonitoringForRegion:
-locationManager:didEnterRegion:
-locationManager:didExitRegion:
```

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

```
-startMonitoringForRegion:
-stopMonitoringForRegion:
-locationManager:didEnterRegion:
-locationManager:didExitRegion:
```

- Run as long as you need to

Run When Necessary

- Significant location changes

```
-startMonitoringSignificantLocationChanges
-stopMonitoringSignificantLocationChanges
-locationManager:didUpdateLocations:
```

- Monitor specific regions

```
-startMonitoringForRegion:
-stopMonitoringForRegion:
-locationManager:didEnterRegion:
-locationManager:didExitRegion:
```

- Run as long as you need to

```
-beginBackgroundTaskWithExpirationHandler:
```

Start Responsibly

Start Responsibly

```
- (void)startLocation
{
    UIApplication *app = [UIApplication sharedApplication];
    self.bkgdTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [self stopLocation];
    }];
    [self startLocationTimer];
    [self.locationManager startUpdatingLocations];
    self.updatingLocations = YES;
}
```

Start Responsibly

```
- (void)startLocation
{
    UIApplication *app = [UIApplication sharedApplication];
    self.bkgdTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [self stopLocation];
    }];
    [self startLocationTimer];
    [self.locationManager startUpdatingLocations];
    self.updatingLocations = YES;
}
```

Start Responsibly

```
- (void)startLocation
{
    UIApplication *app = [UIApplication sharedApplication];
    self.bkgdTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [self stopLocation];
    }];
    [self startLocationTimer];
    [self.locationManager startUpdatingLocations];
    self.updatingLocations = YES;
}
```

Start Responsibly

```
- (void)startLocation
{
    UIApplication *app = [UIApplication sharedApplication];
    self.bkgdTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [self stopLocation];
    }];
    [self startLocationTimer];
    [self.locationManager startUpdatingLocations];
    self.updatingLocations = YES;
}
```

Start Responsibly

```
- (void)startLocation
{
    UIApplication *app = [UIApplication sharedApplication];
    self.bkgdTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [self stopLocation];
    }];
    [self startLocationTimer];
    [self.locationManager startUpdatingLocations];
    self.updatingLocations = YES;
}
```

Start Responsibly

```
- (void)startLocation
{
    UIApplication *app = [UIApplication sharedApplication];
    self.bkgdTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [self stopLocation];
    }];
    [self startLocationTimer];
    [self.locationManager startUpdatingLocations];
    self.updatingLocations = YES;
}
```

Handling Updates

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
    didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Handling Updates

```
- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSMutableArray *)newLocations
{
    if ([[newLocations lastObject] horizontalAccuracy] > self.userMinimum)
    {
        if (self.locationStarted) {
            // wait for a better location
        } else {
            [self startLocation];
        }
    } else {
        [self.view updateWithLocations:newLocations];
        [self stopLocation];
    }
}
```

Summary

Summary

- Horizontal accuracy is better than ever

Summary

- Horizontal accuracy is better than ever
- Location is more available than ever

Summary

- Horizontal accuracy is better than ever
- Location is more available than ever
- Power consumption is lower than ever

Summary

Summary

- Run only as needed

Summary

- Run only as needed
- Tell the user why

Summary

- Run only as needed
- Tell the user why
- Specify your user's activity

Summary

- Run only as needed
- Tell the user why
- Specify your user's activity
- Leave Wi-Fi on

More Information

Paul Marcos
App Services Evangelist
pmarcos@apple.com

Documentation
Location Awareness Programming Guide
http://developer.apple.com/library/ios

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Getting Around Using Map Kit	Nob Hill Tuesday 9:00AM
Understanding Core Motion	Pacific Heights Friday 10:15AM

Labs

Core Location Lab	App Services Lab B Wednesday 3:15PM
Core Motion Lab	Graphics, Media & Games Lab D Friday 11:30AM


WWDC2012

The last 3 slides after the logo are intentionally left blank for all presentations.

The last 3 slides after the logo are intentionally left blank for all presentations.

The last 3 slides after the logo are intentionally left blank for all presentations.