Locations, Location, Locations

A detailed look at Location Services

Ian Whiffin (@BlakDouble)
www.doubleblak.com
18th July 2020

Even though iOS Location Services has been around for a number of years now, there still seems to be some confusion about how and why it does what it does and how reliable the data is. I have testified in court a few times in relation to the data obtained from Location Services but since I recently incorporated this type of data into ArtEx, I thought now was a good time to do a really in-depth study of this technology.

This is a large post, but rather than break it up into smaller pieces, I thought I'd just segment it instead. Please bear with it and click the buttons to reveal additional information.

Also please note that I am not an expert in the mathematics required to calculate location information. What I am presenting in parts of this blog is simplified data to the best of my understanding as a result of research I have done. If you see anything that is incorrect please do not hesitate in contacting me and I will make the required changes. Ultimately, this post is about Location Services, but I couldn't write it without going over some other aspects of how it works.

To make this a little easier, I've also made this into 2 PDF documents for easier printing. The download links are available at the bottom of the page.

Assisted-GPS (A-GPS)

To my knowledge, all Smart Phones are Assisted-GPS devices (A-GPS) rather than straight GPS devices like in-car Sat-Nav systems are.

"Assisted-GPS" is the term used for the collection of technologies used by devices to determine their location.

Using Satellite GPS is the most accurate method for determining location but is not always practical (as described later), therefore other technologies already built into smart phones can be employed instead. Each technique has its own pros and con’s. An overview of each technology is shown below:

Cell Siting

Pros : Fast, Very Low Power
Cons : Lowest accuracy

Cell Siting estimates the location of a device by using the cell signal from the cell tower which it is currently connected to and those around it in a process called **Trilateration**.

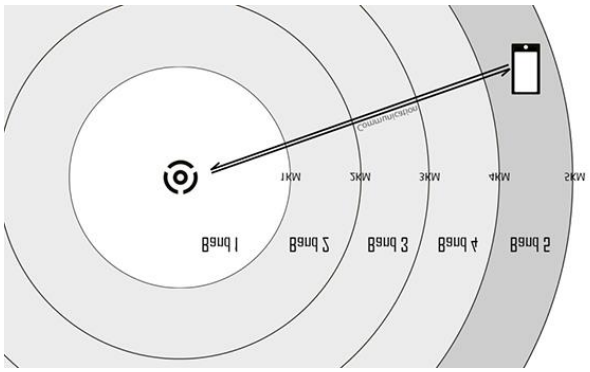
The accuracy of Cell Siting is drastically affected by the density of the towers. More Towers = Better Accuracy.

A device can uses the Received Signal Strength Indicator (RSSI) to calculate it's approximate distance from the cell tower to which it is connected. The RSSI signal should be fairly consistant (notwithstanding obstacles) and so the lower the signal the further the device is from the tower.

Cell Towers constantly send out a signal which is picked up by the device. The device evaluates all of the signals received and selects the best quality signal to connect with. This is the tower that would be used if you made a phone call or sent a text message etc.

As mentioned, RSSI can be affected by obstacles, effectively weakening the signal which is why the distance is only an approximation.

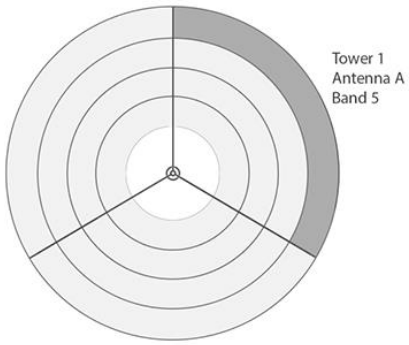
For ease of understanding, I have grouped the distances in my examples into bands. So Under 1KM is Band 1, between 1KM and 2KM is Band 2 and so on.



This the only cell tower in range, it is therefore the strongest signal.
The device "selects" this tower, they handshake and
communication can occur.

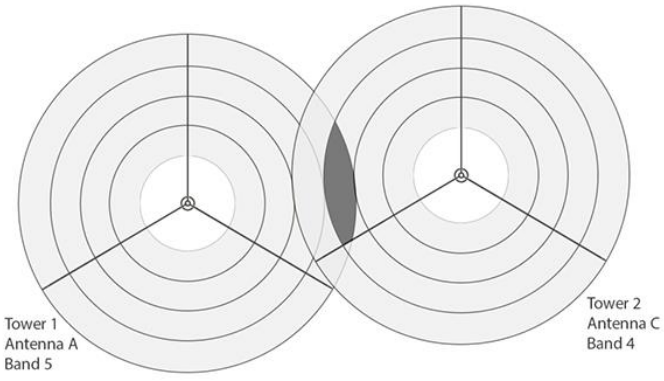
So when trying to identify it's location, the device knows both tower ID and RSSI. In the example above, the phone would use the RSSI to calculate that it's distance would be around 4 to 5 km which I have placed into Band 5.

Some cell towers do have Omni-Directional antennae; which means a single antenna has 360 degree coverage. However, it is far more common to have numerous antennae on a single tower and all antenna focused in different directions. These are configured to best suit the location. We'll cover that in more detail later. For now, let's keep things simple with 3 antennae that each cover 33% of the radius.



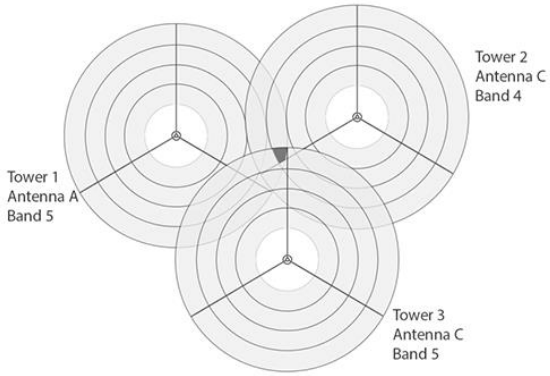
This diagram shows an Cell Tower with 3 antenna, each servicing approximately 120 degrees (Realistically, there's likely to be some overlap). This now means that we know tower, antenan and band. We can work out that the device must be in the dark grey area indicated.

In remote, sparsely populated cell tower locations, this may be as good as you get. But in locations with multiple cell towers in relatively close proximity, the device can use the tower/antenna/band information from the towers to pinpoint it's location more accurately. The secondary towers do not "handshake" with the device, it is just passively emitting a signal which is picked up and measured.



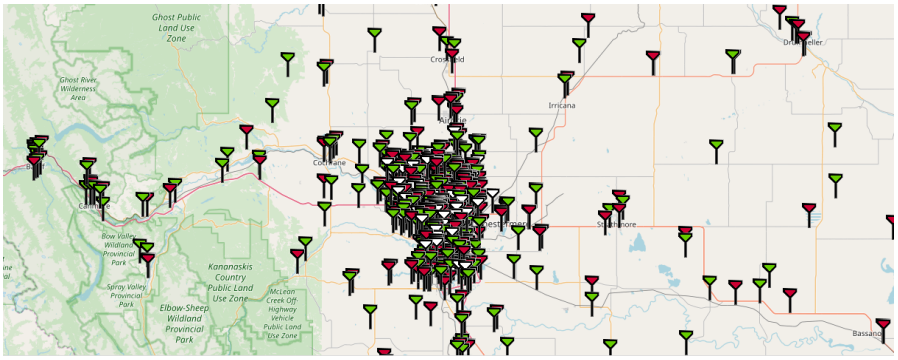
This diagram shows how by using the same information (Tower/Antenna/Band) from 2 towers, the location can be more accurately identifed. The device has to be within the grey area because this is the only area where the cell tower location overlaps.

Adding a third or forth tower into the mix increases the accuracy even more.



Finally, the location can be pinpointed fairly accurately by using more towers and looking where the values overlap.

The screenshot below (from https://www.ertyu.org/steven_nikkel/cancellsites.html) shows the cell tower locations in and around Calgary, Alberta. If you apply the above logic to this map, it's easy to see why the accuracy of Cell Site information can vary so much. Devices in locations with sparse towers simply do not have the option of using multiple towers to determine their location.

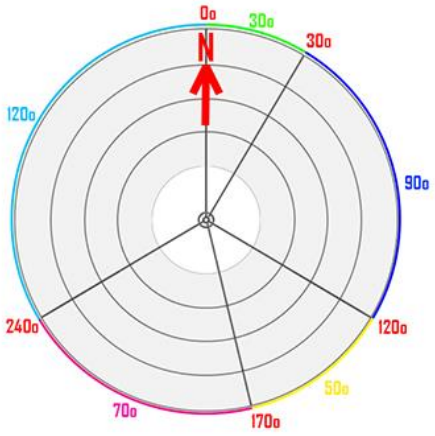


Antenna Configuration

As mentioned above, each tower can have numerous antenna, all configured to point different directions and cover different amounts of area

The configuration of the antenna is defined by AZIMUTH and BEAM WIDTH.

The AZIMUTH is simply the degrees clockwise from NORTH and The BEAM WIDTH is the degrees clockwise away from the AZIMUTH.



This image shows 5 antennae;

Colour	Azimuth	Beam width
Green	0	30
Blue	30	90
Yellow	120	50
Pink	170	70
Cyan	240	120

You can see that the area in Green starts at 0deg and travels 30deg. So the Azimuth is 0 and the Beam Width is 30.

Cell Tower Location Reliability

Geography can play a massive role in cell tower reception; with mountains, buildings, bodies of water etc. all causing reflection of the signal. This is great when it comes to getting a better signal to make calls, but is not so good when trying to work out the devices location. Signal reflection means that the device may actually be connecting to an antenna not pointed in that direction at all.

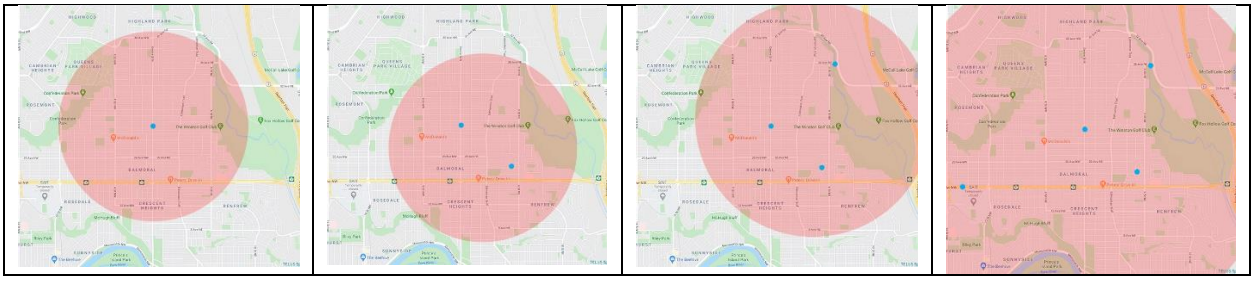
Other factors that can affect cell tower location is number of towers (as discussed above), network traffic or maintenance.

It is important to note that Cell Tower locating is NOT as accurate as many people would like to believe it is. Being connected to a particular tower does not necessarily mean that it is the closet tower.

It's also worth noting that cell towers have a maximum theoretical range of 31 km (~19 miles) but a more realistic range would be less than 15 km ~(10 miles).

Cell Tower Information

Cell Tower locations are usually available (at a cost) from the cell phone providers. This data will likely include the Identifier, the Physical Address of the tower, the Azimuth and Beam width and Range etc. This information is good to know but due to the reliability issues described above, should never be used as conclusive evidence of location.



A single data point results in the location of the user and a radius to show it's accuracy.	A second data point now places the radius around the two locations, thus making the exact centre of the radius an average of the two data points.	A third data point now places the radius around the three locations, thus making the exact centre of the radius an average of the three data points.	And so on.

Another source of cell tower data is www.opencellid.org; an open source database of average locations. Users basically submit their actual location information to the OpenCellID servers along with information about which tower they are connected to at the time. Then, when a user requests to know where a specific tower is, the user submitted locations are averaged.

I actually lean to preferring the OpenCellID data compared to the actual cell tower location. I like it because it's real world data. You don't need to consider obstruction and reflection etc. as the location presented has already been subjected to real obstruction and reflection. Courts however seem to prefer actual Cell Tower addresses. Presenting both, and acknowledging that neither is perfect is in my opinion, the best way to go however.

Long vs Short ID

Cell Towers are typically identified by several pieces of information (I say typically because there are sometimes differences in notation and probably different names/acroynms in different countries too)

Mobile Country Code (MCC) : Quite simply the country that the cell tower is in.

Mobile Network Code (MNC) : The owner/operator of the cell tower.

Local Area Code (LAC / TAC) : The area of the cell tower

Cell ID (CID) : The Cell Tower identifier number.

To confuse matters a little more, the tower could be indentified as LONG or SHORT notation. A LONG CID is a combination of an RNC (Radio Network Controller) and the CID.

If you have a SHORT CID that you need converting to LONG:

LONG ID = 65536 * RNC + CID

If you have a LONG CID that you need converting to SHORT :

RNC = Long CID / 65536 (integer division)
CID = Long CID mod 65536 (modulo operation)

Source : https://en.wikipedia.org/wiki/Mobile_country_code

Source : http://wiki.opencellid.org/wiki/FAQ

WiFi Crowd Sourcing

Pros : More accurate than Cell Siting / Relatively low power
Cons: Not as accurate as GPS / Won't work in remote areas

WiFi Crowd Sourcing works in a similar way to Cell Siting except with WiFi networks.

Devices are constantly scanning for WiFi networks in an attempt to find a network it recognizes (ie one that you have connected to previously) and connect to it for you. Since this scan is occurring anyway, it makes sense to make additional use of this activity.

A WiFi network has an approximate radius of around 70m outdoors but only 45m indoors (due to obstructions such as walls). Obviously, as the device is further away, the signal strength becomes weaker. As with Cell Tower connections, the measurement of the signal strength is known as RSSI (received Signal Strength Indicator). The RSSI value can be used to approximate how far away the receiver is from the emitter. The stronger the signal, the closer the receiver is to the emitter.

The diagram below illustrates a WiFi signal in Red and the device in Blue.

The fact that the device is able to see the network at all means that it must be close; within about 65m (as an approximate once you consider the signal interferences). This would still actually give a fairly wide area, encompassing at least the 4 houses visible within the red circle.

The fact that the device is able to see the network at all means that it must be close; within about 65m (as an approximate once you consider the signal interferences). This would still actually give a fairly wide area, encompassing at least the 4 houses visible within the red circle.

If RSSI is calculated, then the devices approximate distance from the source can also be calculated. In our example above, this would mean that the phone would have to be at roughly the distance it is shown, although it could still be anywhere on the circumference.

Another flaw to this would be that the exact location of every router would need to be known by the phone. If a homeowner moved to a new house or even just moved the router to a new location in the current house, the location information would be affected.

While still based on the similar idea explained above, what actually happens is that a device uses a number of the WiFi networks that it can see, all at the same time. These are not networks that the user has necessarily ever connected. Think about when you are first setting up your new device and it presents you with all of the network names it can see for you to choose from (invariably including the ever ~~funny~~ droll "*FBI SURVEILLANCE VAN*")...

This is the same street as above but with the locations and theoretical radii of all WiFi networks shown.

If we place the device over on the right, you can see that the blue radius only intersects with 4 of the networks.
These 4 networks create a unique "fingerprint" that is not going to be found anywhere

else in the world. That unique fingerprint allows the device to work out its locations with decent accuracy.

WiFi Crowd Sourcing

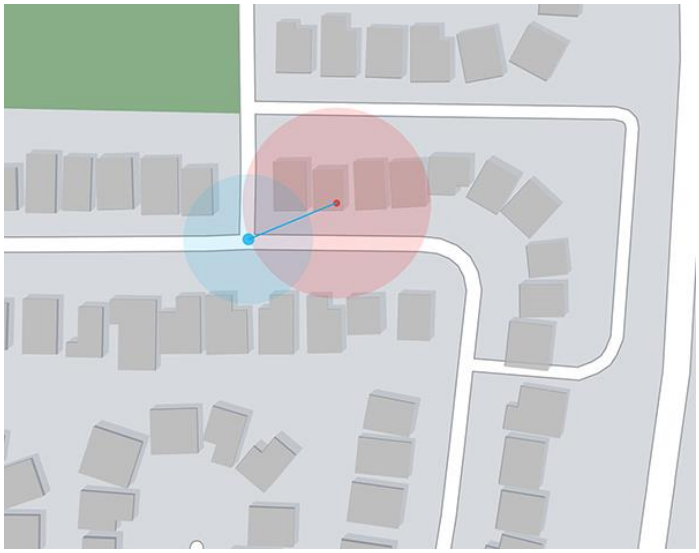
Pros : More accurate than Cell Siting / Relatively low power
Cons: Not as accurate as GPS / Won't work in remote areas

WiFi Crowd Sourcing works in a similar way to Cell Siting except with WiFi networks.

Devices are constantly scanning for WiFi networks in an attempt to find a network it recognizes (ie one that you have connected to previously) and connect to it for you. Since this scan is occurring anyway, it makes sense to make additional use of this activity.

A WiFi network has an approximate radius of around 70m outside but only 45m indoors (due to obstructions such as walls). Obviously, as the device is further away, the signal strength becomes weaker. As with Cell Tower connections, the measurement of the signal strength is known as RSSI (Received Signal Strength Indicator). The RSSI value can be used to approximate how far away the receiver is from the emitter. The stronger the signal, the closer the reciever is to the emitter.

The diagram below illustrates a WiFi signal in Red and the device in Blue.



The fact that the device is able to see the network at all means that it must be close; within about 65m (as an approximate once you consider the signal interferences). This would still actually give a fairly wide area, encompassing at least the 4 houses visible within the red circle.

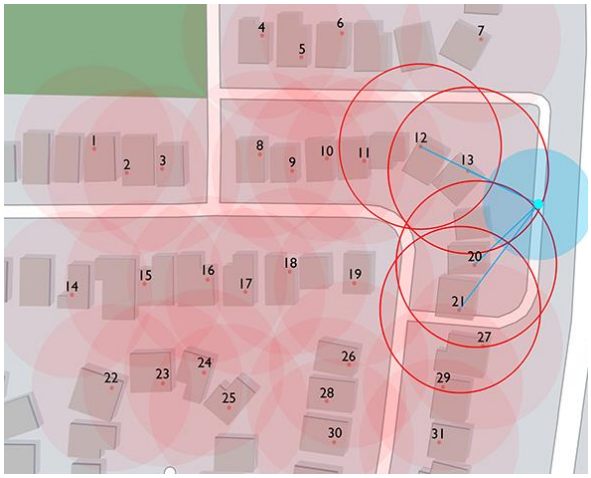
If RSSI is calculated, then the devices approximate distance from the source can also be calculated. In our example above, this would mean that the phone would have to be at roughly the distance it is shown, although it could still be anywhere on the circumference.

Another flaw to this would be that the exact location of every router would need to be known by the phone. If a homeowner moved home or even just moved the router to a new location in the house, the location information would be affected.

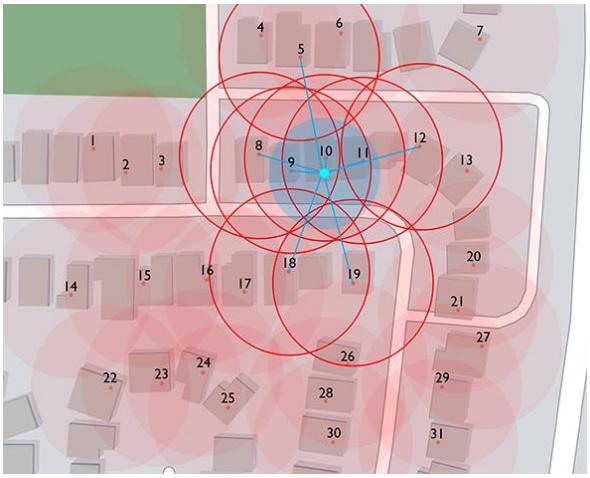
While still based on the similar idea explained above, what actually happens is that a device uses a multitude of WiFi networks that it can see, all at the same time.

This is the same street as above but with the locations and theoretical radii of all WiFi networks shown.

If we place the device over on the right, you can see that the blue radius only intersects with 4 of the networks.
These 4 networks create a unique "fingerprint" that is not going to be found anywhere else in the world. That unique fingerprint allows the device to work out its locations with decent accuracy.



On this example:
If the device moved down, it would lose sight of network 12, but gain sight of the network 27.
Moving up would lose network 21 but gain network 7.



On this example:
If the device moved left it would lose sight of network 12, but gain sight of the network 3.
Moving right would lose network 8 but gain network13.

But how does the unique fingerprint get resolved to a physical location? Wardriving basically.

Wardriving is the act of **searching for Wi-Fi wireless networks** by a person usually in a moving vehicle, using a laptop or smartphone. Software for wardriving is freely available on the internet. Warbiking, warcycling, warwalking and similar use the same approach but with other modes of transportation.

Wardriving - Wikipedia
en.wikipedia.org/wiki/Wardriving

Someone literally drove around recording their own GPS coordinates and the names/signal strengths of all visible WiFi networks at that time. This could have been done by Google or Apple while creating map data, by third party companies such as Skyhook or Wigle or by device users themselves who agree to send that kind of data back to the phone manufacturer / app developers in the form of diagnostics data. Note this paragraph on Apple's website:

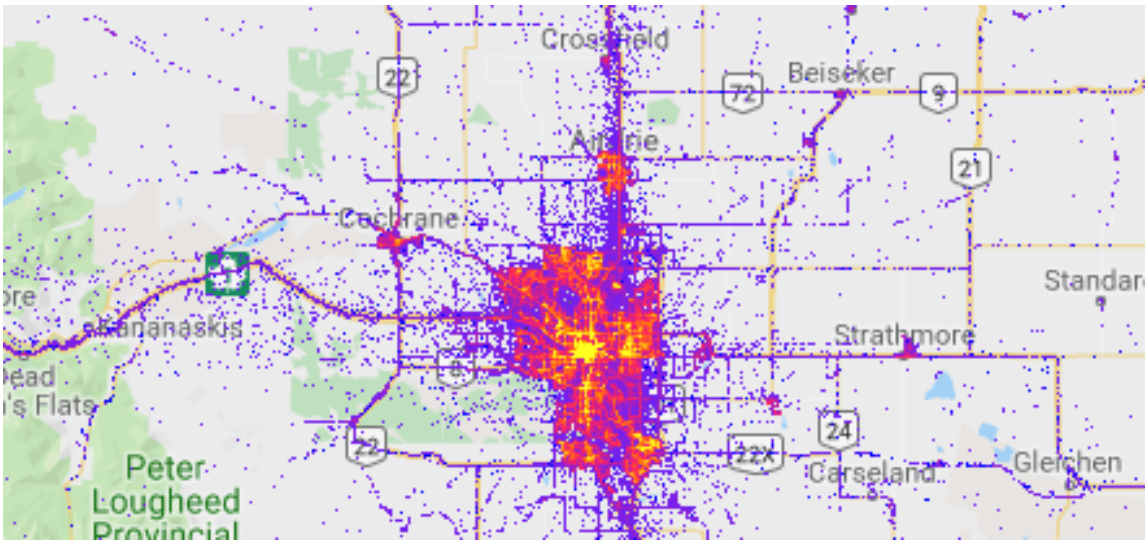
Crowd-sourced Wi-Fi and cellular Location Services

If Location Services is on, your device will periodically send the geo-tagged locations of nearby Wi-Fi hotspots and cell towers to Apple to augment Apple's crowd-sourced database of Wi-Fi hotspot and cell tower locations. If you're traveling (for example, in a car) and Location Services is on, a GPS-enabled iOS device will also periodically send GPS locations, travel speed, and barometric pressure information to Apple to be used for building up Apple's crowd-sourced road-traffic and indoor pressure databases. The crowd-sourced location data gathered by Apple is stored with encryption and doesn't personally identify you.

Remember this paragraph for later on in this post too!

However the data is obtained, it results in a *huge* database of WiFi networks and their locations; a smaller subset of which is stored locally on each device.

And it makes sense. If the device had to go online everytime the location was needed, battery life would go down and data usage would go up. By having a subset of local networks saved on the device, the location can quickly be determined without the battery/data usage overheads. Many of the tests I have done have been with a non-internet connected phone and the locations have been resolved just fine.



Services such as Wigle.net map out the WiFi networks in a searchable way. This also demonstrates the massive difference between highly populated and lesser populated areas.

Bluetooth

Bluetooth beacons can also be used in a similar way and are typically found in stores. For example, when you get near an Apple Store, you phone may pop up with a little

"Hey! Your near an Apple Store" notification. The beacon location can also be used to track your location which is referenced when you turn off Bluetooth with the message "Airdrop, Airplay, FindMy and Location Services use Bluetooth".

Source : https://en.wikipedia.org/wiki/Wi-Fi_positioning_system

Source : https://support.apple.com/en-us/HT203033

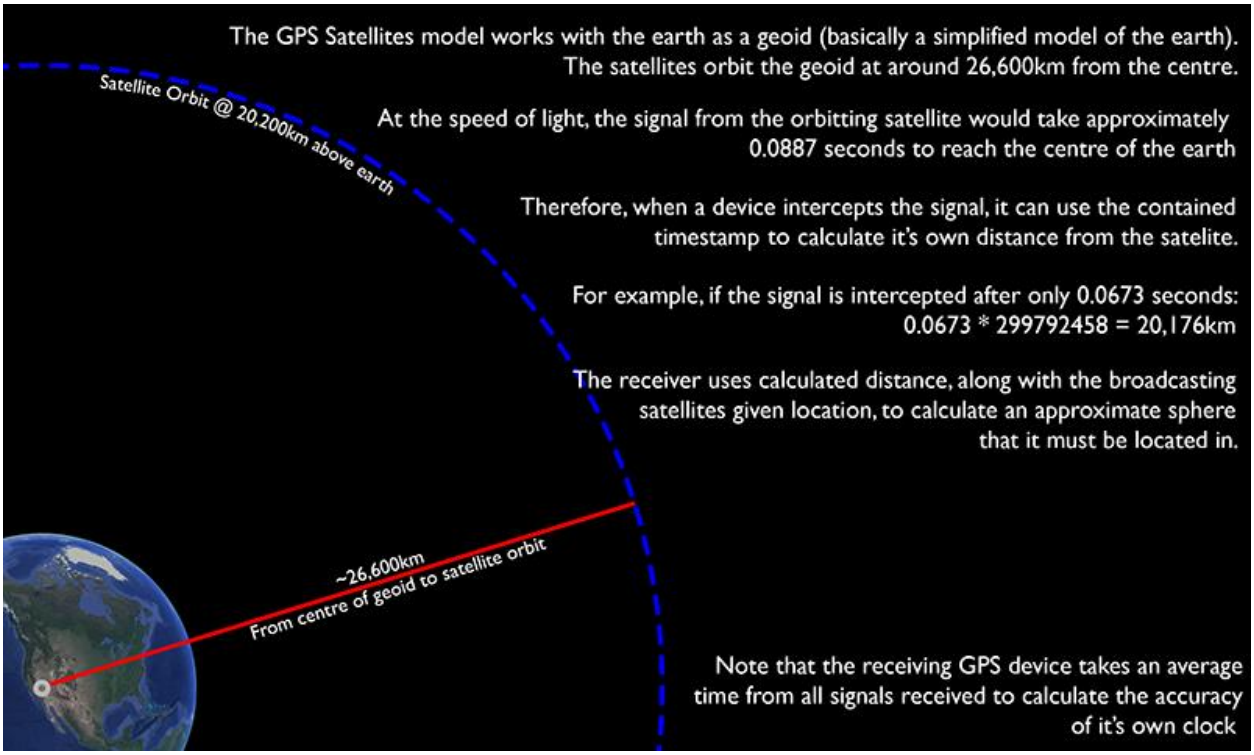
Source : https://en.wikipedia.org/wiki/Bluetooth_low_energy_beacon

GPS

Pros : Most accurate
Cons : Most power consumption / slowest / heavily affected by interference

Finally, GPS can be used to locate the device. GPS is essentially a network of over 30 satellites orbiting the earth each of which contain a synchronized atomic clock and emit a signal which includes (but is not limited to) the time of the signal, an identifier for the satellite and the satellites current location.

The signal travels through radio waves to the earth at 299,792,458 metres/second (Speed of Light) but is slowed down by factors such as atmosphere and geology. As the satellites orbit at 20,200km above the earth, that means the signal takes somewhere around 0.088728049323 seconds to reach the centre of the earth and less to reach the surface.



Notice that the receiver has worked out an approximate **sphere**, not radius. This is because satellites are trying to track in 3 dimensions, not 2 like Cell Siting and WiFi Crowd Sourcing.

The general principle is the same though as once a minimum of 3 satellite signals are received/calculated, the device has 3 intersecting spheres to identify its latitude and longitude.

The timestamp that was sent by each satellite can be used to calculate even more precise location information by comparing the infinitesimally small differences in the time the signals were received.

One huge thing that GPS can do that both Cell Siting and WiFi Sourcing cannot do, is give the devices **altitude**.

Cell Siting and WiFi Sourcing both assume you are "ground" level, so using the method of trilateration explained above is fine. (WiFi 'fakes' the altitude by having a stored list of known altitudes for "ground" level in a given location it is not the devices actual altitude).

GPS however is able to work out the devices Altitude the same way as it works out the Lat/Long. This is the benefit of working with Spheres not Circles. Although it should be noted that:
a) **A forth satellite is required to calculate Altitude**
b) Altitude is not 100% accurate. This is mainly due to the "geoid" shape used to represent the earth not being a perfect match to the earth itself.

Source : https://en.wikipedia.org/wiki/Geoid

Source : https://gulpmatrix.com/how-gps-satellite-navigation-works/

Source : https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks/

iOS Location Services

iOS comes with an API known as 'Location Services' for easily requesting the device location from the device.

Developers can specify how accurate they need the location information to be. This means that applications that just need a general area can ask for low accuracy data (a city or area for example) but applications that require more accuracy such as navigation services can request the higher accuracy; striking a balance between speed and accuracy as the app dictates.

Prior to iOS8, a "Significant" change in location meant when a device changes from one cell area to another and for the most part that was fine. But when iOS8 came out, Location Services had been overhauled to include Geo-Fencing services. It's what allows location based reminders such as "*Hey Siri, Remind me to do X when I leave work*".

Using Geo-Fences meant that it needed to be much more accurate that cell tower locations would allow and so WiFi crowd sourcing became a norm. It makes sense, because as mentioned above, WiFi networks are already being scanned anyway to find a network to connect to. Monitoring it for significant location changes is a small amount of extra programming. Basically, the device keeps track of your location and once you have moved a significant distance, it checks to see if there is a Geo-Fence setup for either the area you have entered or the area you have left and responds appropriately.

The iOS device has a large quantity of locations and WiFi networks stored locally in Cache.sqlite and threebars.sqlite (and possibly elsewhere too) so it is able to resolve locations from WiFi networks without even needing an internet connection. This list of networks appears to refresh every so often in order to keep up-to-date.

iOS Frequent and Cached Locations

Now that we've covered the basics of how devices obtain their location, I'd like to take a closer look at *when* devices get their location data.

I'm sure that most people are familiar with "Frequent Locations" but for the sake of completeness I will describe it again.

Basically, the idea is that every time a device visits a location a record is kept that the device was there from time A to time B. When the device returns to that same location again, a second record is created and therefore the tally of "total visits" to that location increases.

Eventually, when the device has returned to the same location enough times, it is considered a "**Frequent Location**". If you have an iPhone, you can check yourself what your **Frequent Locations** are by going to **Settings > Privacy > Location Services > System Services > Significant Locations** (Note the additional security requirement at this stage; this is why this data only appears on a Full FileSystem extraction).

The actual algorithms that Apple use to determine what is and what is not a frequent location, or even when to record a location visit in the cache are not documented anywhere. I have done exhaustive testing to try to help answer these questions...

Real World Testing

I have an iPhone 6 running iOS12.4.1 which I use for testing. It doesn't have a SIM card and only has a WiFi connection when I'm at home. I placed this phone in the centre console of my car as I drove around the city. The phone was never even touched for the entirety of the journeys. It's also worth noting that this was the first time I'd taken this device to any of these locations.

I extracted the phone several times a day to monitor the changes and plotted out some of my journeys using ArtEx. These are my findings.

Over the course of several days (16th June to 23rd June), with minimal usage of the device, it still cached over 7,000 location records
in *com.apple.routined/Cache.sqlite* database..

There was generally a massive difference on the daily plots depending on how much traveling had been done.

Date	Activity	Total records for day	Average/Hour
16th June	Home > Work > Home	1231	~51
17th June	Home > Work > Home	997	~42
18th June	Home > Work	370	~15

19th June	Work > Shop > Home	684	~29
20th June	3 hour drive around city.	1645	~69
21st June	4 hour drive around city.	1232	~51
22nd June	No journey	890	~37

Notes:
Although the drive around the city on the 20th was approximately the same duration as the drive on the 21st, there was more time spent parked.
Although the 22nd was spent at home, the phone was being moved around between rooms.

So as you can see, with virtually no usage of the device, it was still recording my location on average 42 times every hour, with increased frequency when in motion.

As for what causes the location to be recorded? I'm still not entirely sure. But remember that paragraph from Apples website from earlier that I told you to remember? In particular the bit that says:

"If you're traveling (for example, in a car) and Location Services is on, a GPS-enabled iOS device will also periodically send GPS locations, travel speed, and barometric pressure information to Apple to be used for building up Apple's crowd-sourced road-traffic and indoor pressure databases."

At random(?) times, iOS will just get a GPS fix and send that data to Apple. Awesome as that should mean even better location information that with WiFi Crowd Sourcing.

The time difference between records is sporadic at best; with gaps being anywhere between a single second and 20 minutes or more.

I am of the opinion that the sporadic nature of the data is a result of several things;

a) Being in a location with no/weak WiFi networks - As the location is determined by scanning for local WiFi networks, it stands to reason that no networks means no location data.
b) Unrecognized WiFi networks - maybe the networks that are being seen by the device aren't in the devices location database?
c) As stated, it's just periodic in order to update the Apple servers.

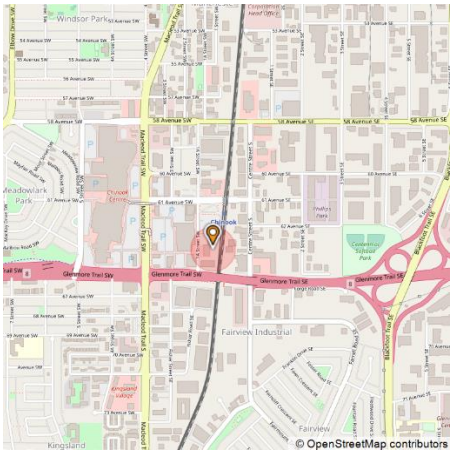
**Update : A complete oversight on my part is that as pointed out above, by test device does not have a SIM card installed. I decided to take a look at my real device and found that some of the cached locations were a 1000m+. This is clearly too large for a WiFi network so left me thinking that either the source WiFi data is inaccurate (unlikely as I would have probably observed the same low accuracy reading on my test phone) or that Cell Tower locations can be used too. I'm guess that switching between Cell

Towers causes a location to be recorded but this is a little difficult to test. It's certainly good to bear in mind though and can account for some way-out accuracies.

Accuracy

This cache database includes "Horizontal Accuracy" as a field which is a value in metres. Basically, it is an indicator of how accurate the phone believes the GPS co-ordinates to be.

ArtEx uses the Latitude and Longitude to plot the location on a map and draws an approximate radius around it using the "HorizontalAccuracy" information. The maps created by ArtEx look like this:

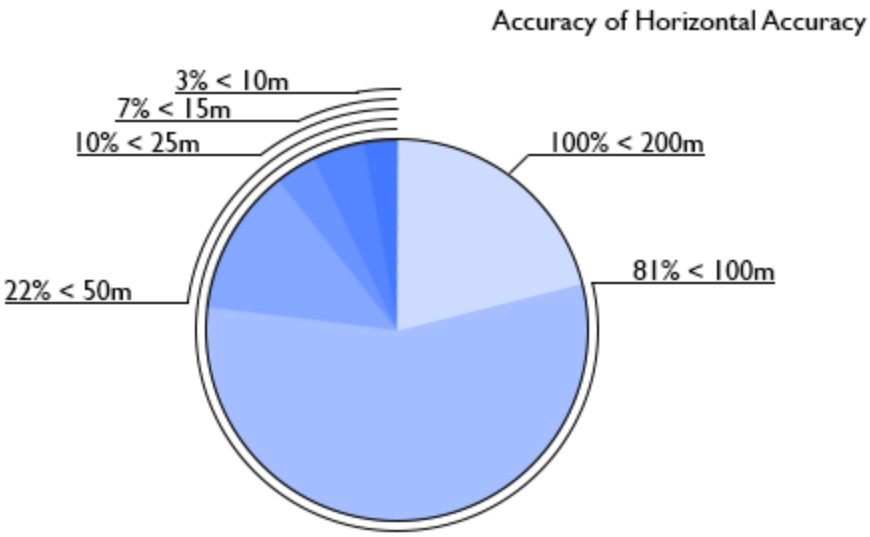


For anyone who hasn't used ArtEx, I should mention that it only draws maps when the Horizontal Accuracy is under 100m. This is simply to save on processing/memory and to make it easier to home in on the most accurate location data.

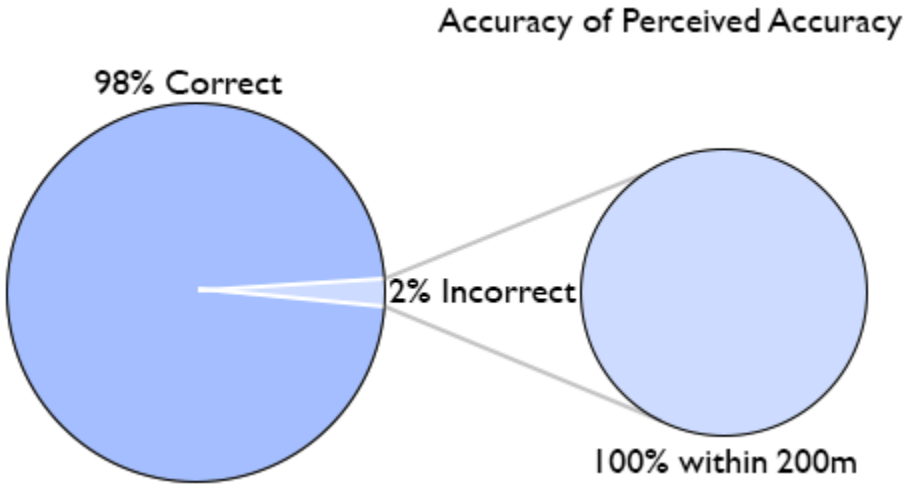
After plotting the map information for my journeys, I manually went though all the created maps and verified their accuracy, including verifying the accuracy of the locations that were not plotted to maps.

Journey 1 - June 20th 3PM to 6PM

Over the course of 3 hours, 455 Cache records were made. Every one of them reported a Horizontal Accuracy of under 200m.

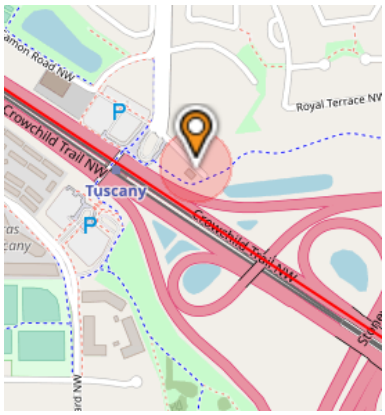


When the "Horizontal Accuracy" was compared to the actual locations I had been I found that the data was even more accurate than I had expected..

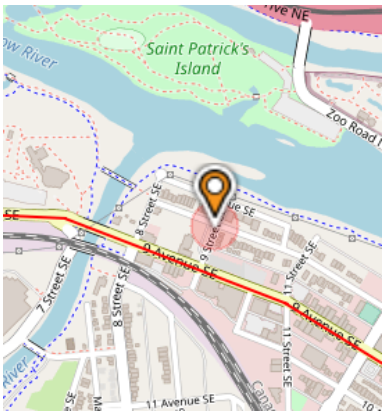


Of the 455 records, 445 were within the radius specified. The 10 that were incorrect would have been correct if the Horizontal Accuracy hadn't been quite so confident. For example:

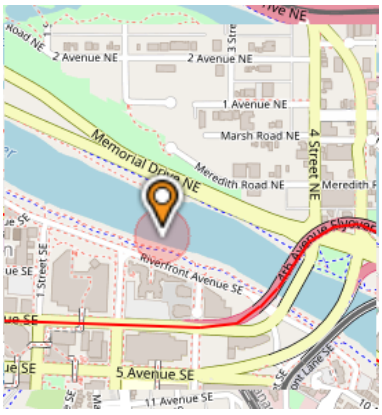
Red line indicates actual path of travel



This is so close that the discrepancy could be down to the accuracy of the radius that ArtEx drew.



This one is a little further afield, the distance between the marker and the red line is around 95m.

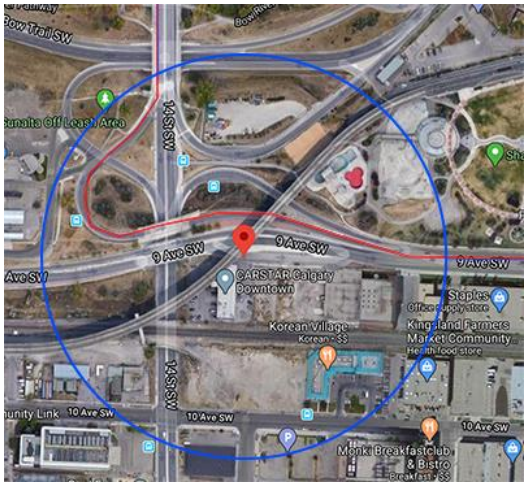


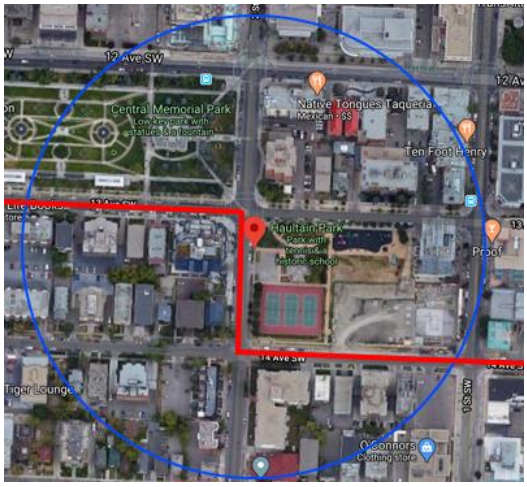
This one was the furthest afield, with the distance between the marker and the red line being around 190m.

So although the locations were within 200m of my actual location, the Horizontal Accuracy field was incorrect.

I also manually plotted the locations that fell outside of the 100m accuracy and for which ArtEx didn't map for me.

Red line indicates actual path of travel. Blue circle indicates Horizontal Accuracy Radius

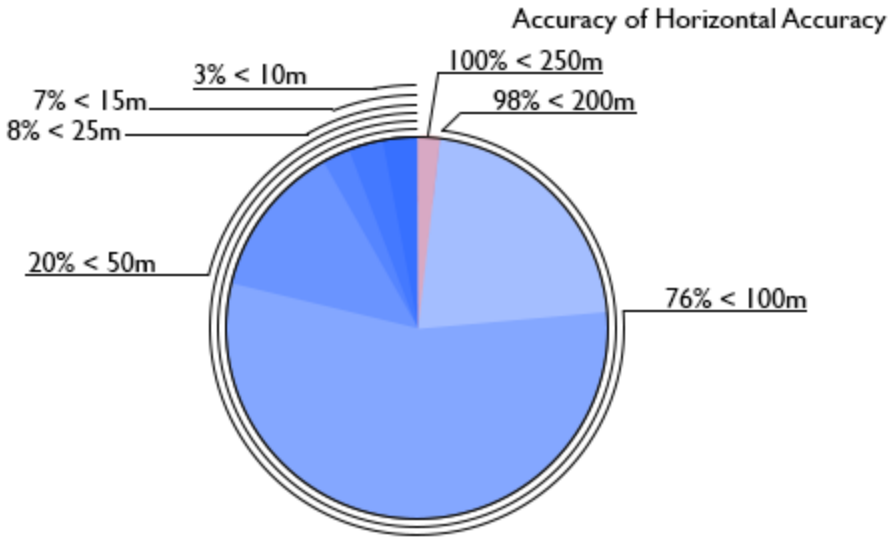




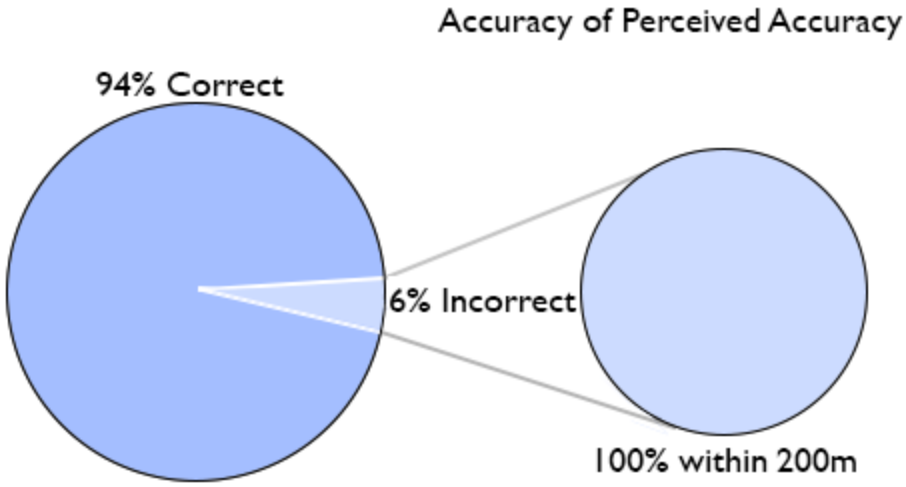
Even the "low accuracy" points were still incredibly close to my actual path.

Journey 2 - June 21st 11AM to 3PM

Over the course of 4 hours, 578 Cache records were made. Every one of them reported Horizontal Accuracy of under 250m.

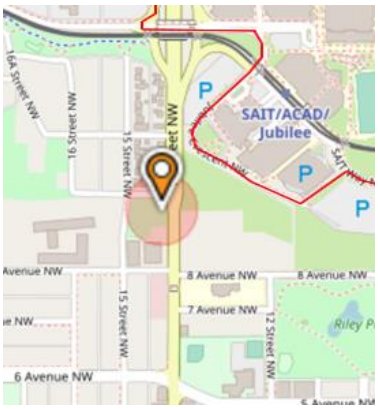


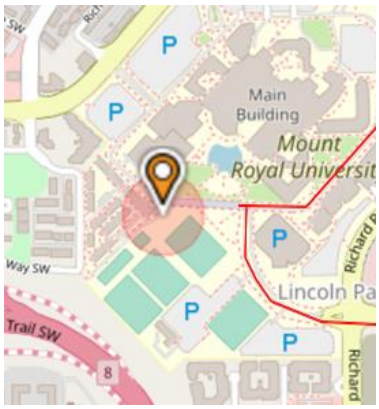
When the "Horizontal Accuracy" was compared to the actual locations I found it was relatively similar to the day before but not quite as good.

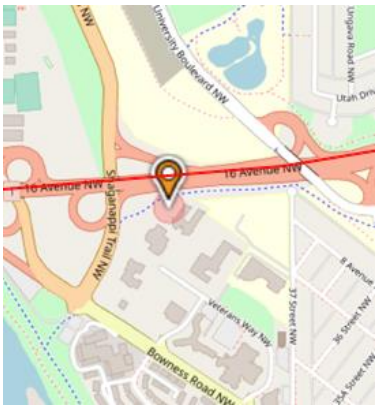


Of the 578 records, 546 were within the radius specified. The 32 that were incorrect would have been correct if the Horizontal Accuracy hadn't been quite so confident. For example:

Red line indicates actual path of travel







In every case on this day, the incorrect records were still pretty close.

The upgrade to "Frequent Location"

So now we have an idea of how often locations are saved in the Cache database and how accurate they are. But what flips the status from simply a location that was visited to one that is designated a "Frequent Location"?

Let's start by taking a look at the tables found in Local.sqlite and Cloud.sqlite (Both very similar content but as the names suggest, one is local on the device and one is designed to be sent to the cloud).

The three tables I'm interested in right now are:

ZRTLEARNEDLOCATIONOFINTERESTMO
ZRTLEARNEDLOCATIONOFINTERESTTRANSITIONMO
ZRTLEARNEDLOCATIONOFINTERESTVISITMO

Firstly, ZRTLEARNEDLOCATIONOFINTERESTMO is a list of the device's Frequent Locations. Of note, it includes ZLOCATIONLATITUDE, ZLOCATIONLONGITUDE, ZDATAPOINTCOUNT, ZPLACECREATIONDATE and ZPLACEEXPIRATIONDATE but it does not include any information about when they were actually visited.

I noticed that of the 6 locations, all had the same expiry date and the creation dates were not unique either.

ZPLACECREATIONDATE (Converted)	ZPLACEEXPIRATIONDATE (Converted)
2020-06-10 11:22:42 (-06:00)	2020-08-10 18:43:12 (-06:00)
2020-06-10 11:22:43 (-06:00)	2020-08-10 18:43:12 (-06:00)
2020-06-15 07:38:13 (-06:00)	2020-08-10 18:43:12 (-06:00)
2020-06-22 18:36:37 (-06:00)	2020-08-10 18:43:12 (-06:00)
2020-06-22 18:36:39 (-06:00)	2020-08-10 18:43:12 (-06:00)
2020-06-22 18:36:42 (-06:00)	2020-08-10 18:43:12 (-06:00)

You can see that records 1 and 2 share a Creation Date, as do records 4, 5 and 6 (well, within a few seconds anyway).

This means that the Frequent Location record cannot possibly be made at the time a location is revisited and must happen afterwards, during a single process that works out all visits and creates the appropriate records in this table.

I typically found that this process ran every 2 to 4 days but don't quote me on that. There really wasn't any discernable pattern. I expected maybe every X days or every X records but neither seemed to be right.

ZRTLEARNEDLOCATIONOFINTERESTTRANSITIONMO relates to the movement between locations; You may have noticed on your own phone when reading Frequent Locations it often says something to the effect of "*Arrived via a 10 min. drive*". This table is how it knows that you drove and how long it took.

The **ZLOCATIONOFINTEREST** field relates to the **Z_PK** on the **ZRTLEARNEDLOCATIONOFINTERESTMO** table.

Again we have Creation and Expiry Dates, but now we also have **ZSTARTDATE** and **ZSTOPDATE** which relate to the journey to the frequent location. This table also has a field **ZPREDOMINANTMOTIONACTIVITYTYPE** which is the mode of travel. 4 appears to be car whereas 1 appears to be walking.

ZRTLEARNEDLOCATIONOFINTERESTMO is a log of the time actually spent at the Frequent Locations.

Again you can see that a lot of records share the same creation date, suggesting that a block process creates these records at the same time.

ZCREATIONDATE (Converted)	ZENTRYDATE (Converted)
2020-06-12 16:14:32 (-06:00)	2020-06-10 16:02:19 (-06:00)
2020-06-15 07:38:10 (-06:00)	2020-06-14 16:11:19 (-06:00)
2020-06-22 18:36:31 (-06:00)	2020-06-21 12:58:28 (-06:00)
2020-06-17 06:59:56 (-06:00)	2020-06-16 05:04:38 (-06:00)
2020-06-22 18:36:31 (-06:00)	2020-06-19 16:15:54 (-06:00)
2020-06-18 07:45:05 (-06:00)	2020-06-17 16:04:23 (-06:00)
2020-06-22 18:36:31 (-06:00)	2020-06-20 18:14:43 (-06:00)
2020-06-17 06:59:56 (-06:00)	2020-06-16 15:57:06 (-06:00)
2020-06-15 07:38:10 (-06:00)	2020-06-12 15:55:16 (-06:00)
2020-06-15 07:38:10 (-06:00)	2020-06-13 11:57:05 (-06:00)
2020-06-10 11:22:41 (-06:00)	2020-06-09 15:58:31 (-06:00)
2020-06-22 18:36:31 (-06:00)	2020-06-18 05:05:19 (-06:00)
2020-06-10 11:22:41 (-06:00)	2020-06-09 09:15:02 (-06:00)
2020-06-22 18:36:31 (-06:00)	2020-06-19 15:33:18 (-06:00)
2020-06-18 07:45:05 (-06:00)	2020-06-17 05:00:36 (-06:00)
2020-06-22 18:36:31 (-06:00)	2020-06-20 15:39:32 (-06:00)
2020-06-16 05:17:26 (-06:00)	2020-06-14 17:24:46 (-06:00)
2020-06-11 09:54:29 (-06:00)	2020-06-10 05:05:04 (-06:00)

You can see that there are:
3 records at 2020/06/10 at 07:38:10 (Green)
3 records at 2020/06/10 at 11:22:41 (Yellow)
2 records at 2020/06/17 at 06:59:56 (Pink)
2 records at 2020/06/18 at 07:45:05 (Orange)
6 records at 2020/06/22 at 18:36:31 (Red)

You will also notice that the creation dates are not in date order (It's ordered by the **Z_PK** field) and the CreationDates are also incredibly similar to the PlaceCreationDates of the **ZRTLEARNEDLOCATIONOFINTERESTMO** table.

So we can assume that there is a process that runs every few days that determines which locations have been visited enough to upgrade them to "Frequent Locations". The presentation below goes into a little more detail about how this works.

Interestingly, in the *Cloud.sqlite* database is a table called **ZRTADDRESSMO** which is the resolved address for each of the Frequent Location records. This includes Locality, City, Road, Area of Interest etc.. BUt this table is not found in *Local.sqlite*.

Note for testing : The records are created once a condition has ended. For example, the Frequent Location Visit record is only recorded in the database once I leave the location. Don't do test extractions expecting to see when you arrived at home/office if you are still there. Likewise if you were imaging a device at the scene, you will not get the time the device arrived there.

Now we've seen how the cached records are distilled into Frequent Locations, we can see that it actually doesn't matter how many times you visit a location. **What matters is how many records there are to show you in that location**. Records are created somewhat randomly (at least that may be how they appear), and if you're there long enough, there may be enough records during a single visit to qualify as a Frequent Location.

In my testing, I had a Frequent Location listed within just 38 DataPoints; a location I only visited once but spend an hour at walking around.

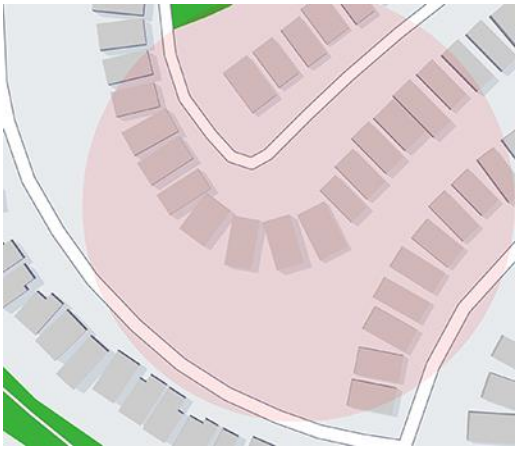
We should also define *what a Frequent Location* is in terms of area.

Obviously, to be useful in knowing when I'm at "Home" or "Work" or wherever, it's no good to have a specific Lat/Long coordinate. For example, I want my phone to know I'm at home regardless if I'm pulling onto my driveway or at the far reaches of my backyard. For my Work address, the area may be even larger. So there must be some tolerance.

My testing has shown that once i am around 80m from my home address, my phone registers that I've now left that Frequent Location. I tested this by slowly moving away from the house and stopping frequently for a set amount of time. Once I extracted the phone, I knew how far away from my house I was at the time the "Exit" event occurred.

As this has been consistent at around 80m, I don't believe this is the result of a coincidence that my phone decided to create a record at that time and I instead believe that this is directly related to my phone leaving the WiFi range of my home network triggering a location check.

Of course, 80m is a fairly massive area...

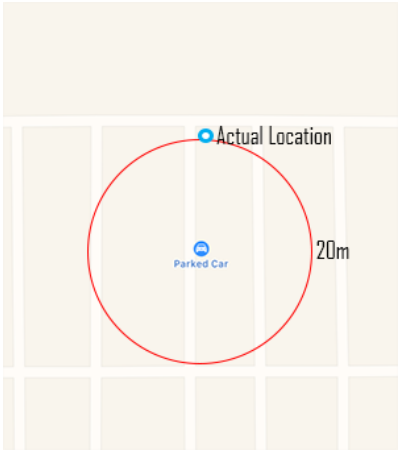


The good thing about this is that just being close to a frequent location is probably enough to keep the record of it (instead of losing it as a purged cache record). The downside is that if you need highly accurate data, it may not be good enough.

Vehicle Events

Another location artifact of note is the vehicle event locations stored in ***~/private/var/mobile/Library/Caches/com.apple.routined/local.sqlite***. This appears to be somewhat related to the bluetooth connection information found in KnowledgeC.

The idea is that when I'm in my car, my phone is connected via Bluetooth. Once I stop driving and turn the car off, the bluetooth connection ends and a vehicle event of "Parked Car" is created.



In practice, I've found it to be cleverer than it sounds.

I conducted several tests with different variables:

While in motion, Bluetooth turned off on the phone: No vehicle event was recorded.

While in motion, Bluetooth turned off on the car stereo: No vehicle event was recorded.

Drive to location A, turn engine off and leave the phone static in the car for 25 minutes: No vehicle event was recorded.

Drive to location A, turn engine off and walk away from the car with the phone: vehicle event created before I was even 100m away.

I have tested these scenarios multiple times, with a few variables such as with/without network connectivity etc. and it appears pretty consistent that it is the combination of

Bluetooth Disconnection followed by walking that causes the phone to realise you have gotten out of the car and therefore create the Parked Car event.

This was important for a case I was working where I was able to correlate the bluetooth connection/disconnection events of a device to frequent location events.

The thought process was that when the bluetooth connects, it was because the engine started and a journey (ie. a Exit Frequent Location in this case) was anticipated.

Entering a Frequent Location (ie arriving home) would precede a Bluetooth Disconnection as the user arrived home and exited the car.

The issue was that there were bluetooth disconnection and reconnection event with no associated Frequent Location information. This suggested the user parked the car briefly but there was absolutely no Vehicle Park events for the day in question.

My determination was that the phone never left the car, hence a park event was never recorded. The time of the bluetooth disconnection/connection was extremely relevant and showed that *something* occurred at that time. This hypothesis was later supported by additional evidence found soon after..

Update : As pointed out by Josh Hickman (@josh_hickman1 / thebinaryhick.blog), CarPlay also appears to create Vehicle Events. Alas, my car does not have CarPlay so I am unable to test. And the last time I drove a CarPlay vehicle was well over the 2 months that the vehicle events appear to be stored for.

Looking at the Extraction of Josh's phone that was made available in May, there are zero bluetooth events related to a vehicle and 82 vehicle events. Although closer inspection shows that there were numerous duplicates and only 5 unique events.

WiFi Access Points

Also in the Cache.sqlite database is a table called **ZWIFIACCESSPOINTMO** which contains thousands of records, each with RSSI, Date and MAC address (Amongst other things).

Before you get all excited at finding this treasre trove of location data, you should notice that the dates are very... same-y. It appears that this is (one of) the tables used by the device to assist with determining locations and is not a list of where the device was.

The ZFINGERPRINT field appears to relate to the ZRTFINGERPRINTMO table but that's as far as I've looked into it. It can't be used to determine the devices location at any given time so I've not investigated any further at this time.

netusage.db

Finally, it would be remiss of me to write an article about iOS Cell Siting and Location Services without mentioning netusage.db.

netusage.db can be found in ~/private/var/networkd/netusage.db and contains a table called ZNETWORKATTACHMENT.

This table keeps track of all cell towers connected to; but it only keeps the FIRST and LAST DATE the connection is made.

For example; My phone connects to tower A for the first time. A record is created with the Tower ID and today's date as both the FIRST and LAST connected dates.

When I connect to that tower again the day after, the existing record is updated with a new LAST connected date.

It's not great, but I used it (in absence of much other location data) to prove that a device had left the city and headed west, contradicting the suspect's story. Without going into too much detail, the device had numerous cell tower connection records that all showed a FIRST connection date of the date of the offence and I used their order of creation in the database to work out a rough direction of travel.

Of course, this was only really helpful because the distances involved were way too much to be the result of network handoff. The truly beautiful thing was that the cell tower connections pretty much matched the connections of my victim's device; for whom we also had Cached Location data which confirmed the location data obtained from the cell tower connections.

Wrapping Up

Thank you for sticking with this until the end. I realise it's a big topic but hopefully my research has answered some questions you may have had or will make life easier for you in the future.

Location Services is something that the majority of people know exists, but there are so many questions remaining.

Overall, I'd say it is very reliable with only a few anomalous records which still aren't bad, just not perfect. (Just remember to take into account the Horizontal Accuracy information).

You'll need to access a FULL filesystem extraction (passcode must be known) in order to get the location data and the cache is typically wiped after 7 days so bear that in mind too.

My rule of thumb is that if the phone says it was there, it was. Or it was at least close. BUT, if it doesn't say it was there it **does not mean that it wasn't there**. It just means that a record wasn't created.

Remember, you can download my iOS Usage Visualization tool "**ArtEx**" for FREE from the 'Software' section of my site which maps out Location data from Cache, Frequent Locations, Cell Towers, Media, Apple Pay and Vehicle Events.