

# 30억건의 사용자 위치기록으로 전국 3000만개의 WiFi AP MAP 구축하기

구자형  
로플랫 (loplat)

loplat

# loplat

전국 단위의 실내 위치 플랫폼을 만들어,  
다양한 서비스에 기술을 제공하는 일을 하고 있습니다.

## Unsupervised locating of WiFi access points using smartphones

J Koo, [H Cha](#) - IEEE Transactions on Systems, Man, and ..., 2012 - [ieeexplore.ieee.org](http://ieeexplore.ieee.org)

WiFi positioning systems require radio maps in the form of either RF fingerprints or positions of WiFi access points (APs). In particular, knowledge of the AP positions is essential to enable a locating mechanism as well as to understand the nature of underlying WiFi ...

☆ 77 63회 인용 관련 학술자료 전체 4개의 버전

## Localizing WiFi access points using signal strength

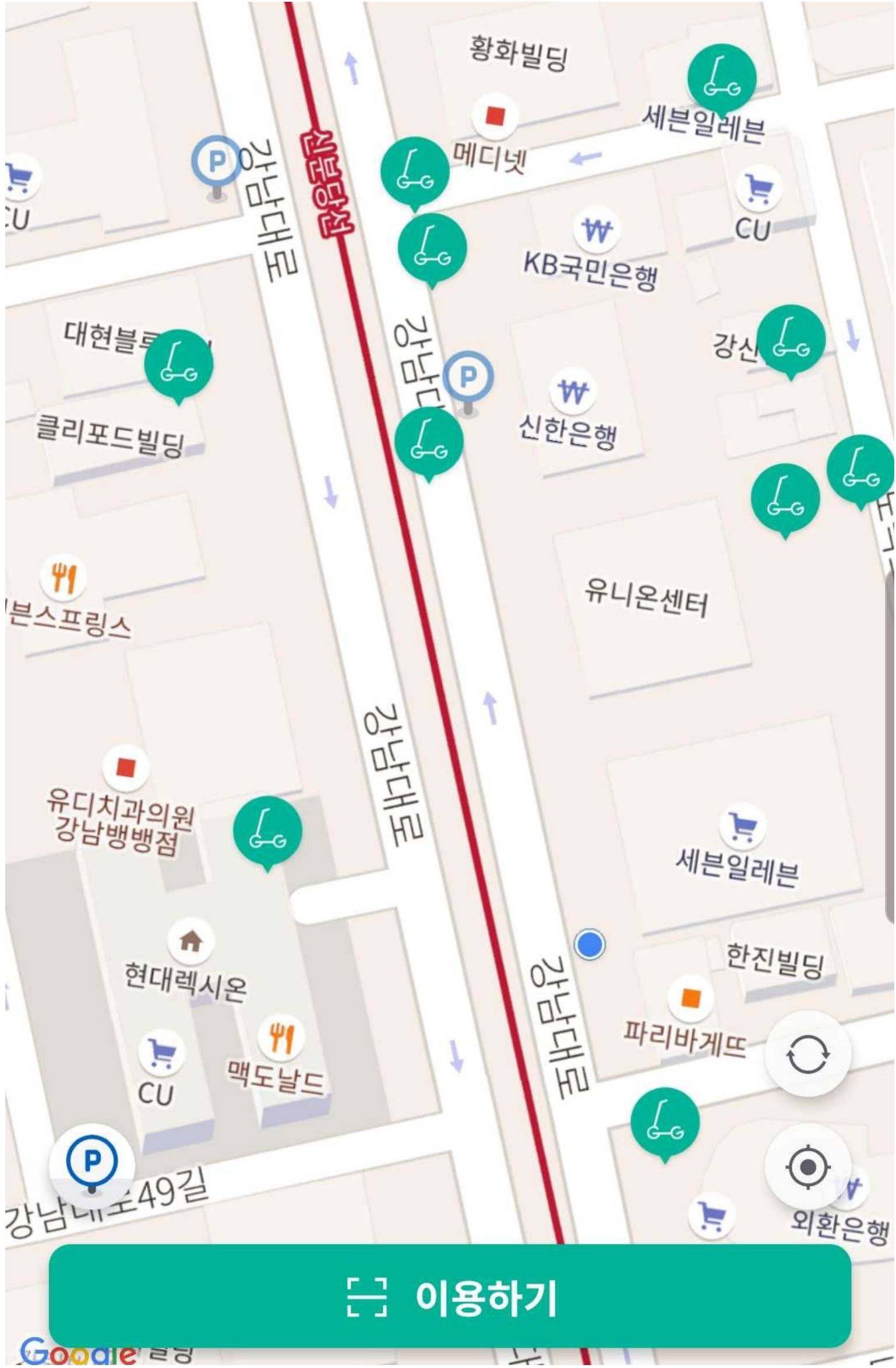
J Koo, [H Cha](#) - IEEE Communications letters, 2010 - [ieeexplore.ieee.org](http://ieeexplore.ieee.org)

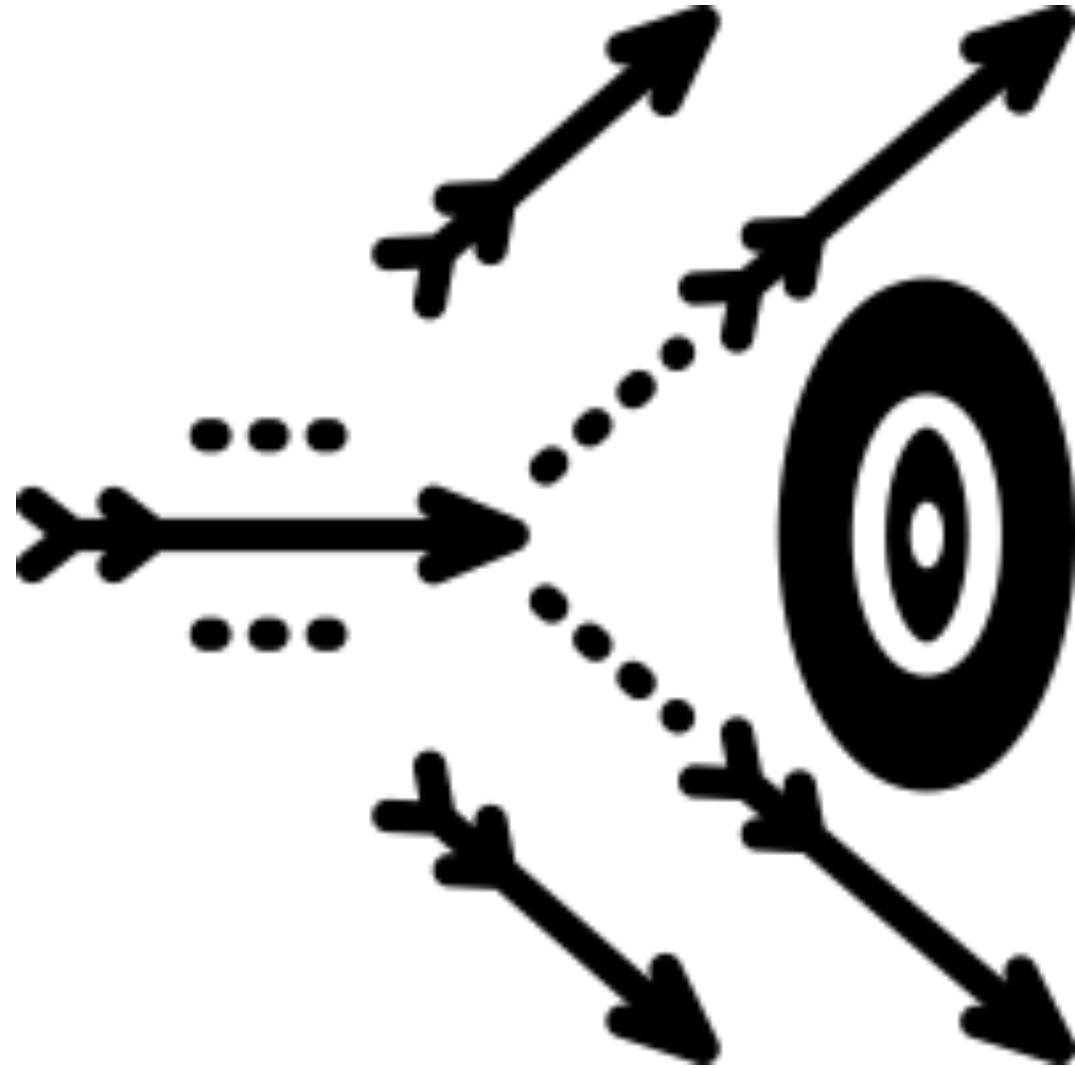
When estimating the location of WiFi access points (AP) in an anonymous environment, the required parameters of the radio propagation model are not readily obtainable. This letter investigates the use of received signal strength (RSS) for range-based AP localization when ...

☆ 77 108회 인용 관련 학술자료 전체 5개의 버전



**DEVIEW  
2019**







GPS

+



WPS

# CONTENTS

DEVIEW  
2019

WPS 이것만은 알고 가자

Crowdsourcing 데이터로 WiFi AP 위치 찾는법

30억건의 데이터로 WiFi AP MAP 한시간만에 만들기



# 1.WPS 이것만을 알고 가자

# WPS: WiFi Positioning System

DEVIEW  
2019

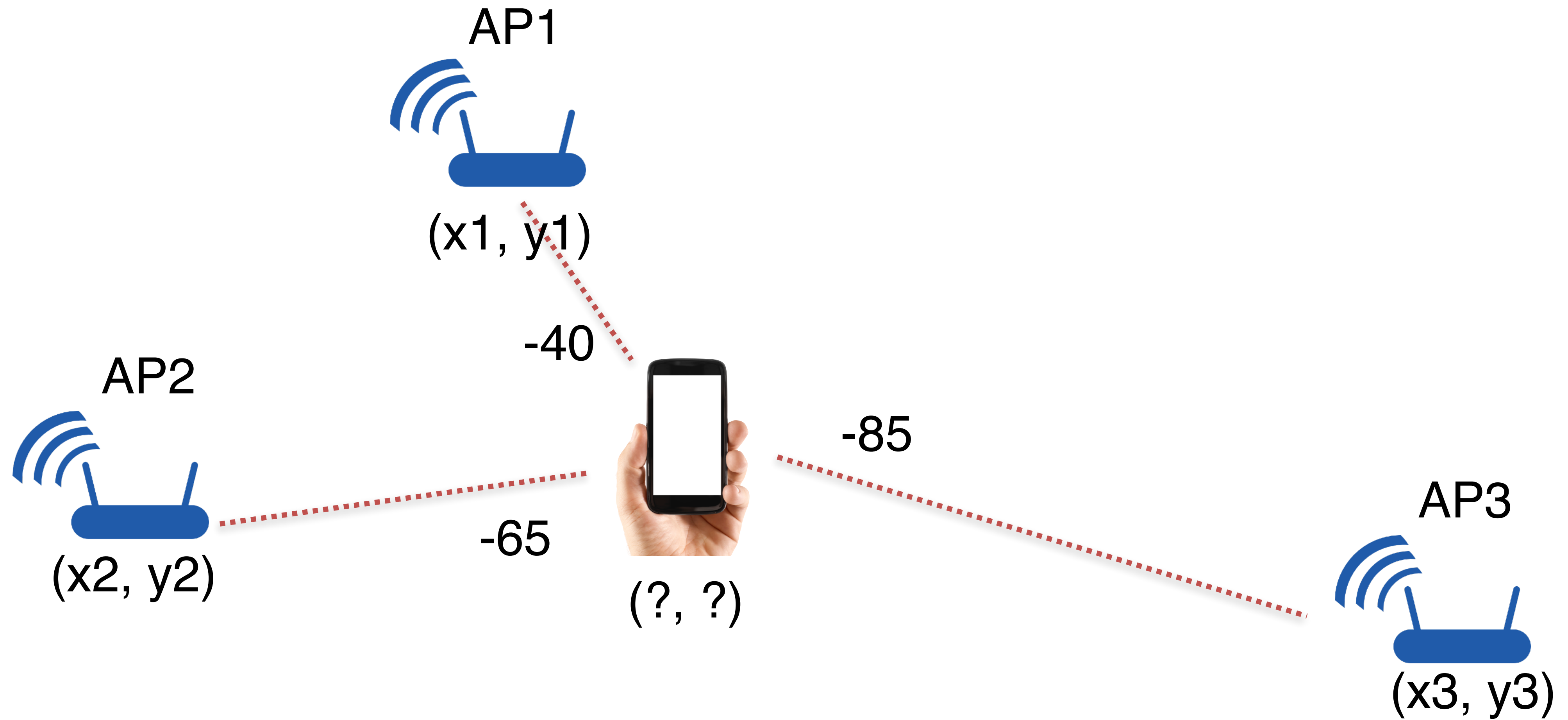
## WiFi 에 대한 기본 정보

1. WiFi 는 대부분의 경우 고유한 식별자가 있다 (bssid - mac address)
2. WiFi 신호는 거리가 멀어질수록 약해진다

그렇기 때문에, WiFi AP(Access Point)의 위치를 알고  
와이파이 신호를 수신할 수 있다면 어떠한 기기든 위치를 추정할 수 있다.

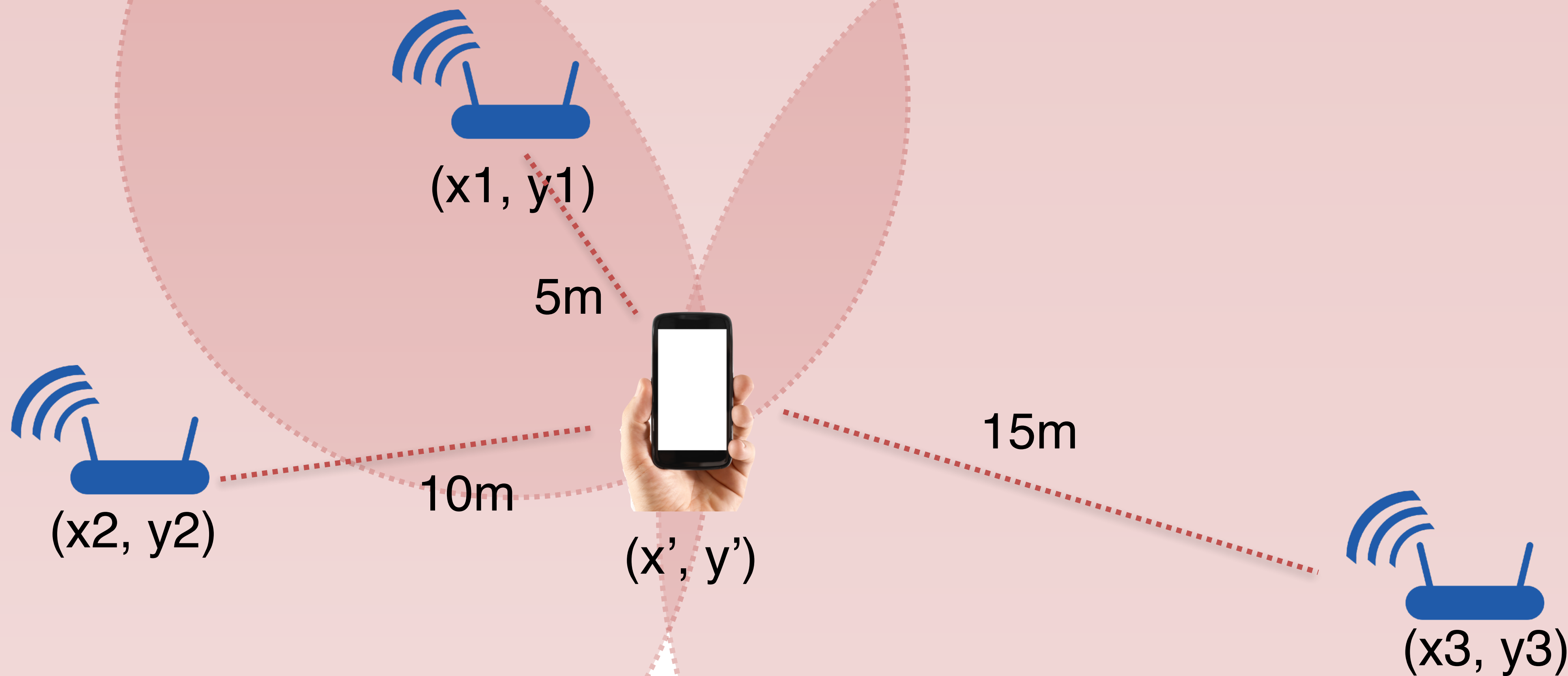
# Triangulation vs. Fingerprint

# 삼각(삼변) 측량법



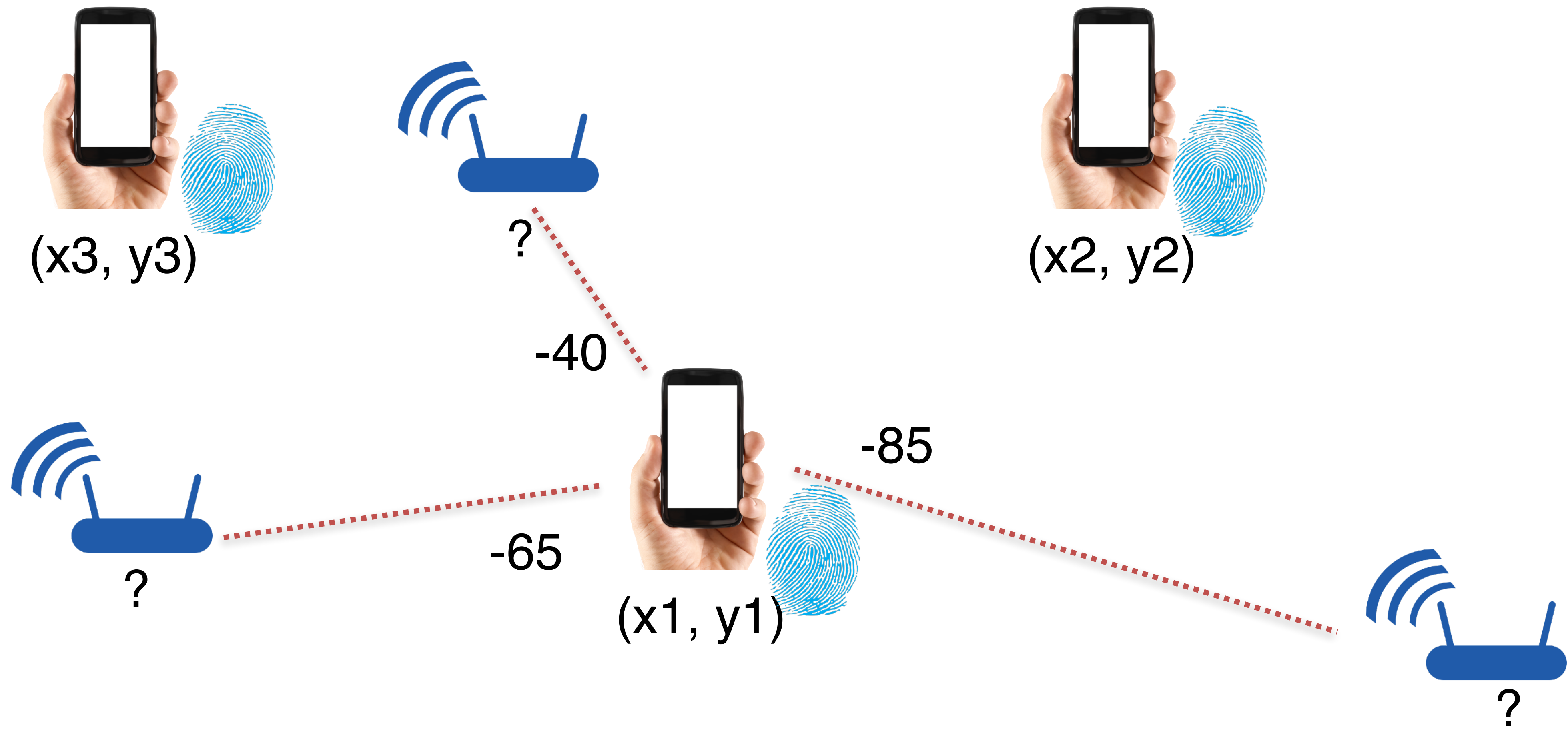
# 삼각(삼변) 측량법

DEVIEW  
2019



# WiFi Fingerprint

DEVIEW  
2019



전국 어디서든 WiFi 신호로  
위치를 찾을 수 있는 시스템을 만들자!

실내 : Fingerprint 를 촘촘하게 수집하는게 좋다.

전국 : AP 위치를 찾는게 획~얼씬 유리하다.



# WPS 구축 = WiFi AP의 위치를 찾는 문제



radio scans with ground truth

no	rss			location	
1	<b>API</b>	AP2		<b>X1</b>	<b>Y1</b>
2	<b>API</b>	AP2	AP3	<b>X2</b>	<b>Y2</b>
3	<b>API</b>	AP3		<b>X3</b>	<b>Y3</b>
4	<b>API</b>	AP3	AP4	<b>X4</b>	<b>Y4</b>
5	<b>API</b>	AP5		<b>X5</b>	<b>Y5</b>

특정 AP가 관측된 위치를  
평균내면 된다.

# 전국 스케일 시스템 구축은?

DEVIEW  
2019



# War Driving: 10 Years Ago

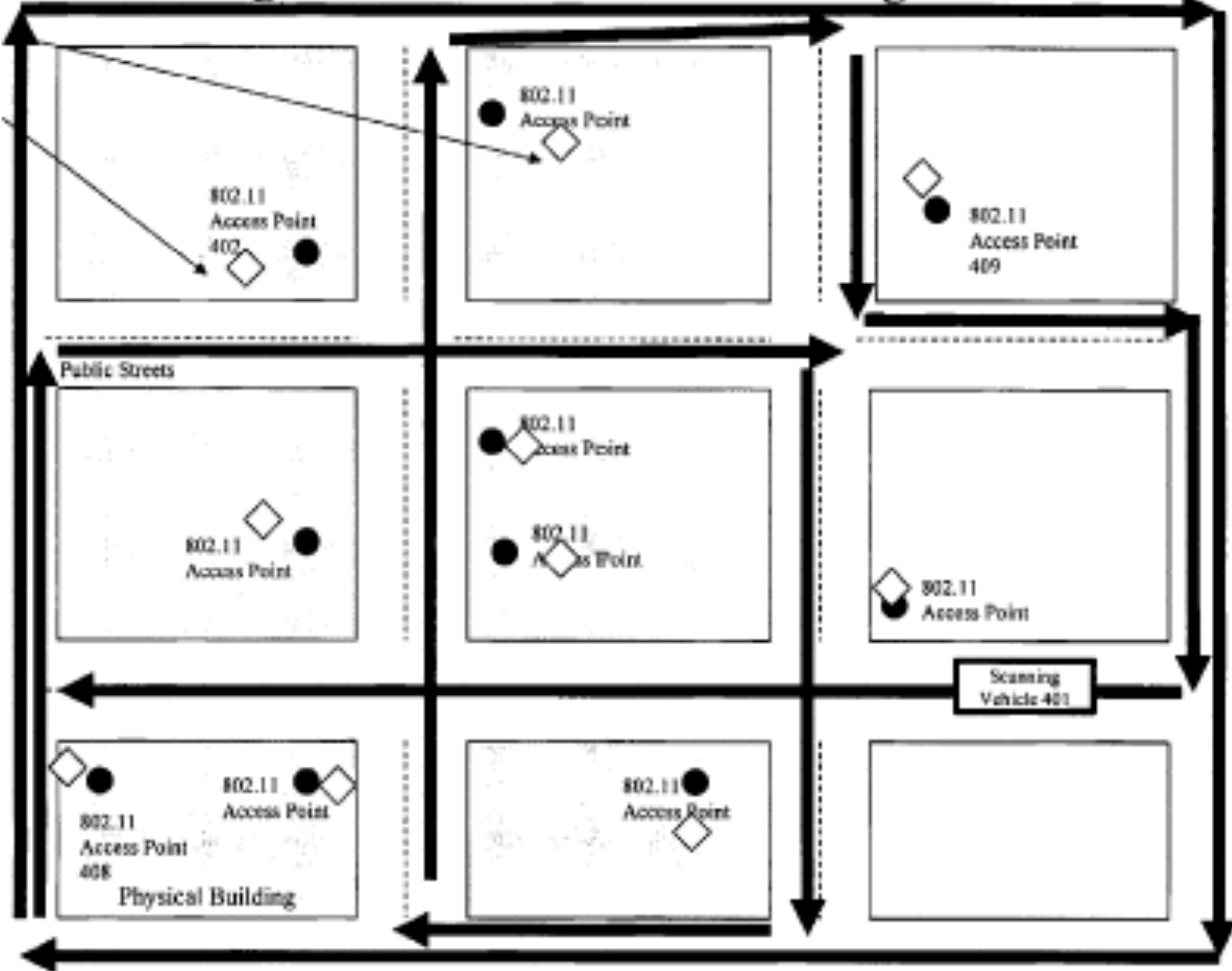
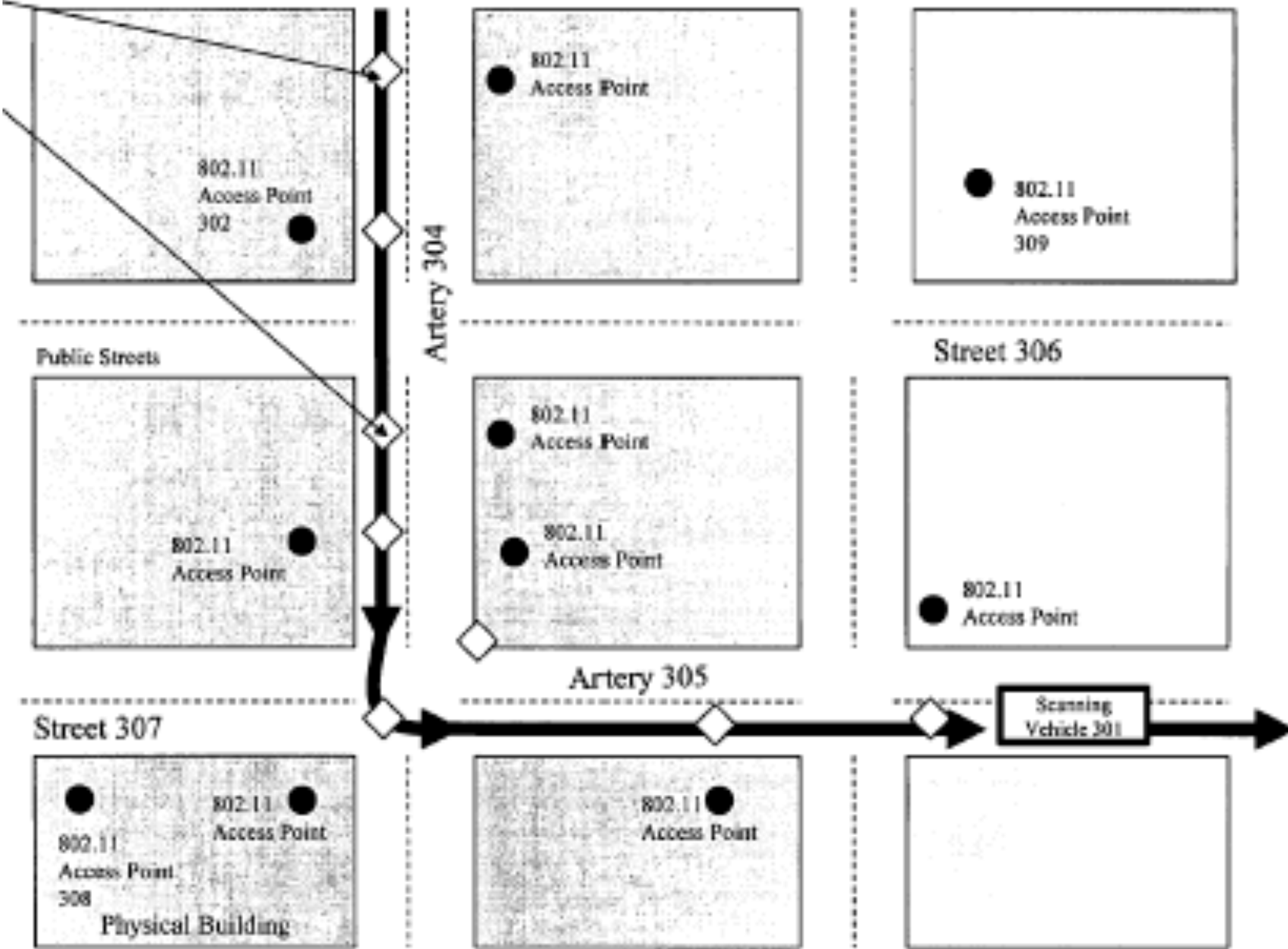
DEVIEW  
2019



skyhook wireless

# War Driving: Driving Technology

DEVIEW  
2019



## 2. Crowdsourcing 데이터로 WiFi AP 위치 찾는법

# 다시 생각해 보자

굳이 차로 모아야 하나?

GPS + WiFi Scan 만 있으면 된다

많이 있으면 된다



DEVIEW  
2019





36.1234, 126.9876

"Starbucks", aa:bb:cc:dd:ee:ff, -56



# 운이 좋게도 ...

## 현재 부하

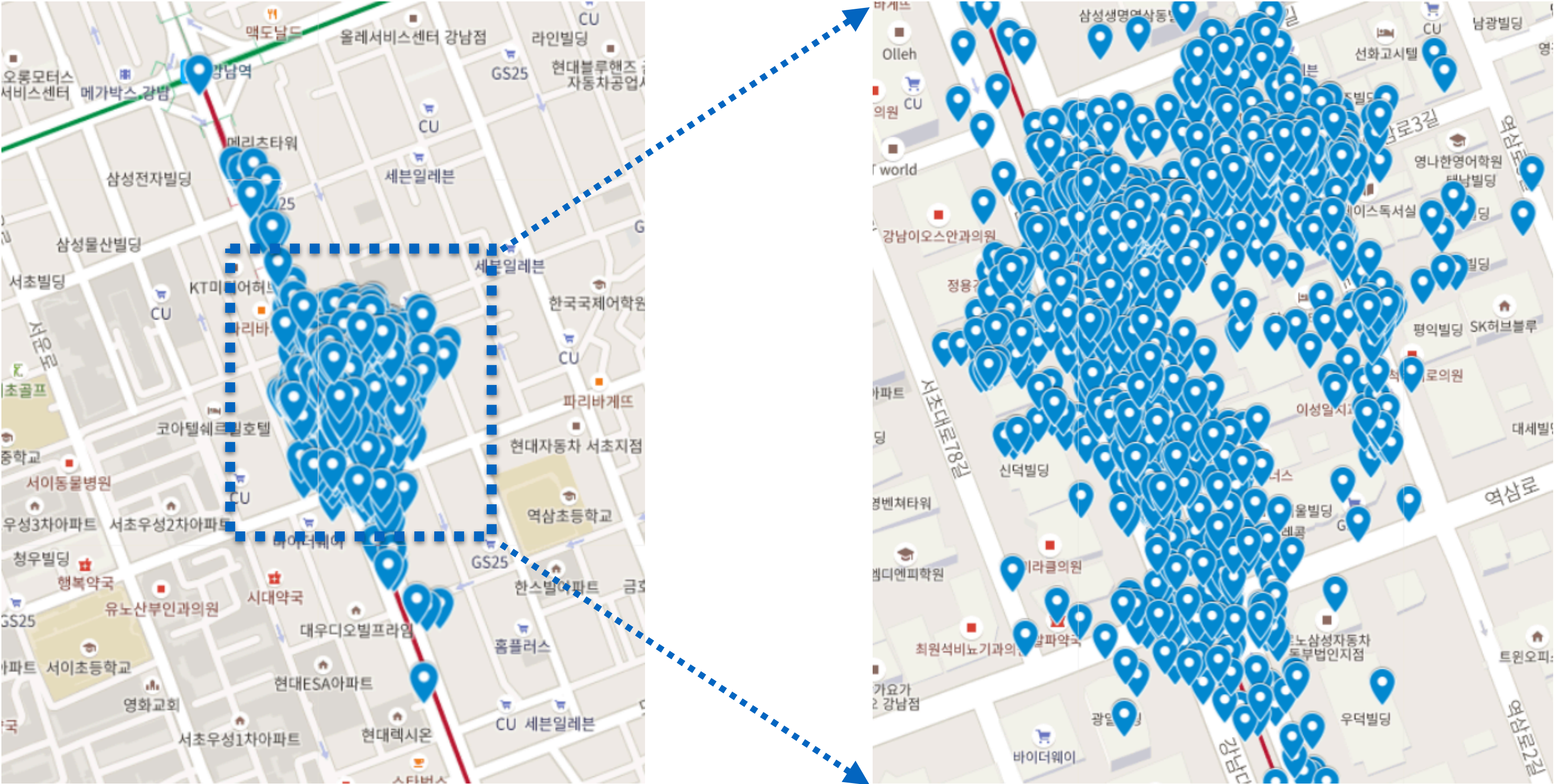
URI	요청/분 현재	요청 지난 24시간
/searchplace	894.6	56,730,494

36.1234, 126.9876

'coex',aa:bb:cc:dd:ee:ff, -56

# 하나의 AP 에 대해 관측데이터가 엄청많다

DEVIEW  
2019



# Centroid



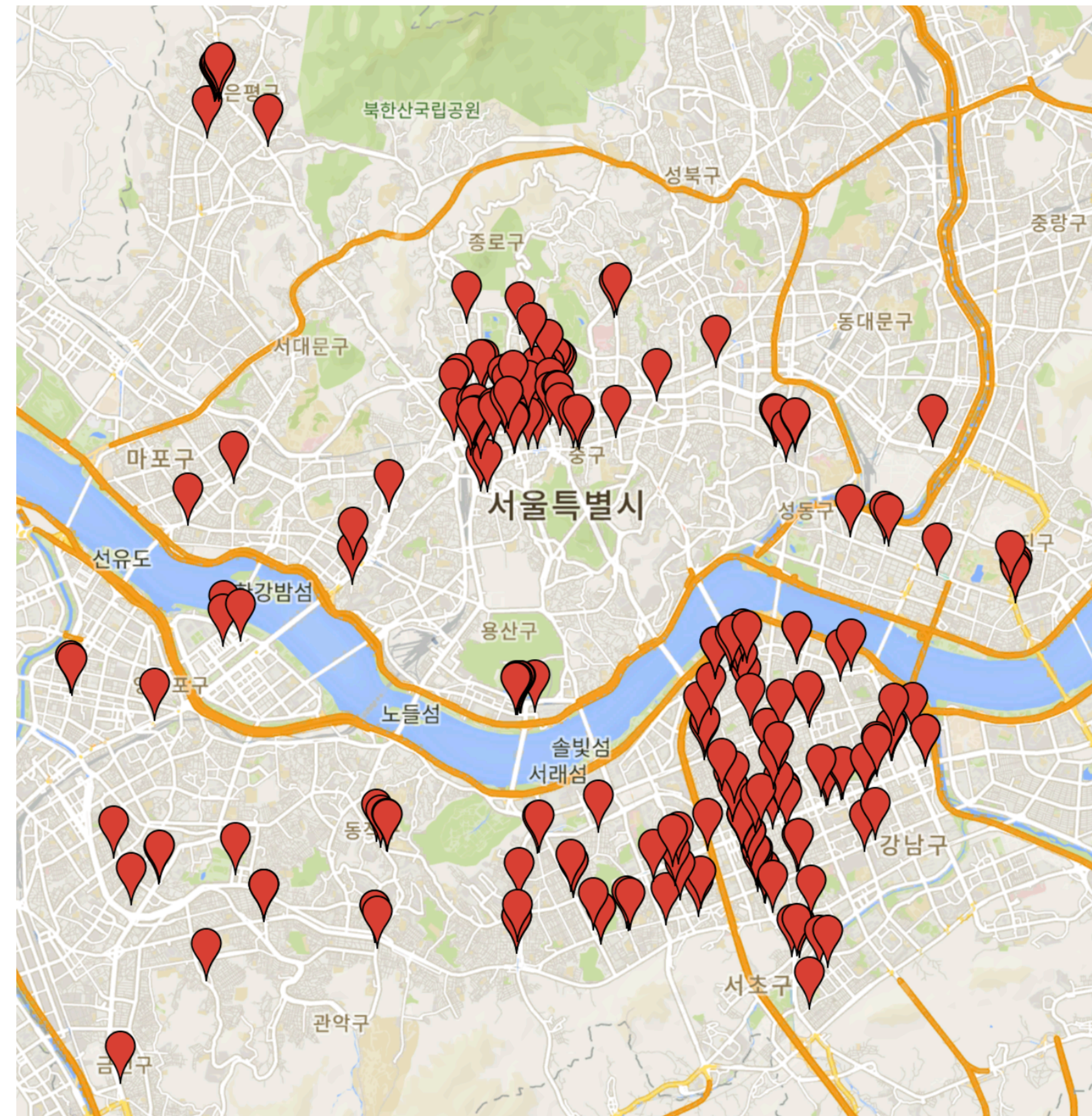
Centroid

$$\left[ \frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3} \right]$$



# But, AP 관측이 예상만큼 아름답지 않다.

DEVIEW  
2019



# WiFi 흥행 시대 ..., 그만큼 다양한 상황

DEVIEW  
2019



버스



지하철



핫스팟



이사

그외, AP 식별자 중복, GPS 오차,  
정보지연 등 OS 이슈, +모르는 이유들

# 이유가 많다. 그래서

DEVIEW  
2019

여러 상황을 한번에 기계적으로 다룰 수 있는  
단순한 방식이 필요하다.

# 클러스터링

유효한 AP 가 있다면,  
실제 위치 근처에서 대부분 관측이 될 것이다.

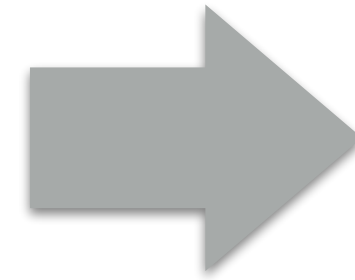


# Clustering

DEVIEW  
2019

AP#1

위도,	경도,	신호세기
(37.1212,	126.1234,	-50)
(37.1213,	126.1233,	-40)
(37.1223,	126.1243,	-90)
...		
...		
(37.1211,	126.1233,	-67)



거리기준 clustering

클러스터 포함 거리기준 = 200 미터

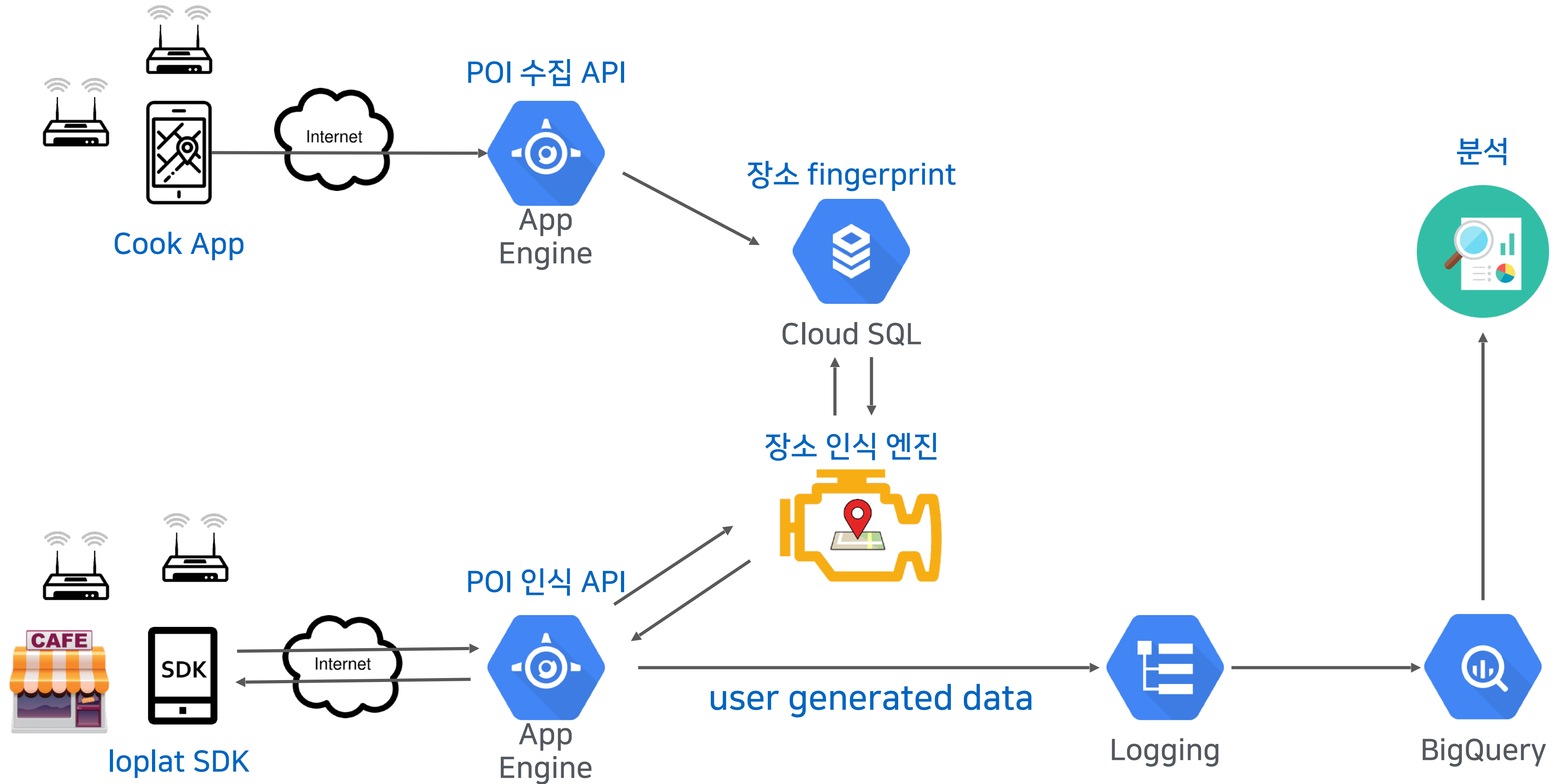
```
for 모든 관측
  for 생성된 클러스터
    if 관측 in 클러스터
      update 클러스터
    else
      새 클러스터 추가
```

즉, 관측데이터로부터 만든 클러스터 중  
지배적인 클러스터 (90%) 가 있고,  
최근에 지속적으로 관측이 되었다면  
바로 그 클러스터에서 AP 의 위치를 결정하면 된다.

# 3. 30억건의 데이터로 WiFi AP MAP 한시간만에 만들기

# Our Service Architecture

DEVIEW  
2019



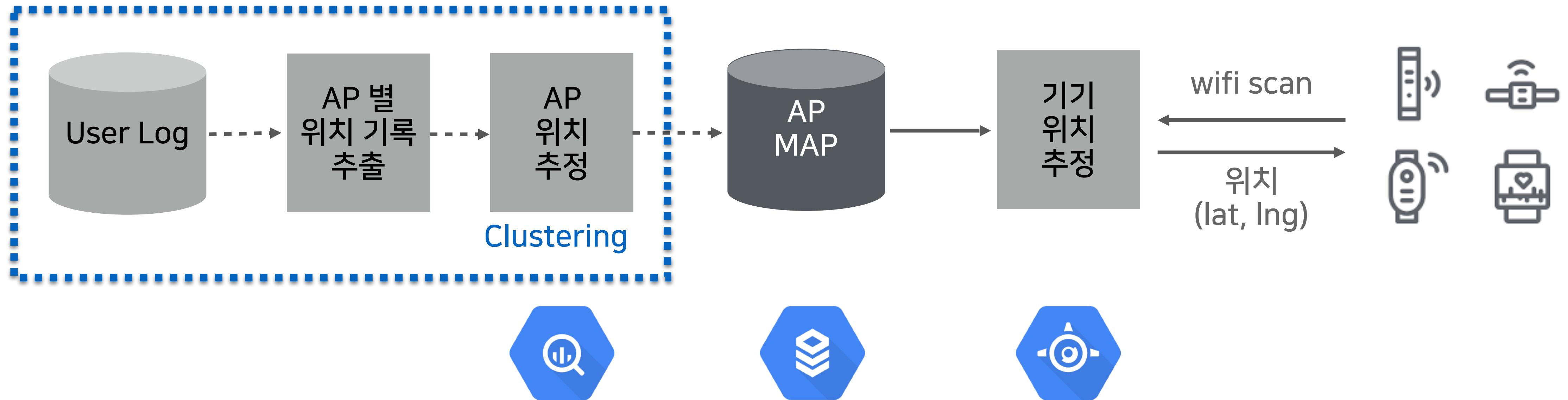


19. 9. 1. ~ 19. 9. 30.

BigQuery	<b>Analysis</b>	사용량	298	tebibyte	\$1,489
BigQuery	<b>Active Storage</b>	사용량	37,728	gibibyte	\$779
BigQuery	<b>Long Term Storage</b>	사용량	69,738	gibibyte	\$720
BigQuery	<b>Streaming Insert</b>	사용량	7,848,355	mebibyte	\$392

# WPS Building System

DEVIEW  
2019



# Building WiFi AP Map in Bigquery

DEVIEW  
2019

1 단계: AP 별 관측 기록 모으기

2 단계: Clustering 을 통한 AP 위치 추정

# 단계1. AP 별 관측 기록 모으기

DEVIEW  
2019

get\_bssid\_location\_list\_2

편집기 숨기기 전체 화면

```
1 CREATE TEMPORARY FUNCTION
2   parseVisit(q STRING, k STRING)
3   RETURNS STRING
4   LANGUAGE js AS """
5     return parseVisit(q, k);
6   """
7 OPTIONS (
8   library = 'gs://mutely/bigquery_udf_libs/loplat_bq_udf_lib1.js'
9 );
```

실행 쿼리 저장 보기 저장 쿼리 예약 더보기

실행 시 이 쿼리가 15.2TB를 처리합니다. ✓



# 단계1. AP 별 관측 기록 모으기 (BigQuery)

DEVIEW  
2019

```
SELECT bssL.bssid, ts local, bssL.rss, lat, lng
FROM
  (SELECT * FROM
    (SELECT
      ts_local
      ,parseBsslist(msg) as bssList
      ,parseVisit(msg, 'lat') as lat, parseVisit(msg, 'lng') as lng
    FROM
      (SELECT timestamp, msg
        FROM `request_log_*`, UNNEST(protoPayload.line) AS line
        WHERE _TABLE_SUFFIX BETWEEN '20170910' AND '20180110'
      )
    )
  ), UNNEST(bssList) as bssL
WHERE bssL.rss > -75
GROUP BY bssL.bssid
```

# 단계1. AP 별 관측 기록 모으기 (결과)

DEVIEW  
2019

Row	bssid	ts, rss, lat, lng, time_lag, accuracy, provider(net, gps, lpp)
1	00:23:aa:83:cb:f1	71117,-54,37.4773216,127.1244652,42,19,net,71121,-62,37.4771148,127.1249
2	90:9f:33:80:5e:d4	70915,-79,37.5079379,127.033314,7,20,net,71120,-68,37.5080559,127.03339
3	00:1d:33:03:da:bc	70902,-67,37.4931193,126.7881793,1,64,net,70901,-70,37.4932977,126.7869
4	02:23:aa:6a:11:39	71103,-75,37.8304458,126.8328597,2,30,net,71203,-72,37.8302186,126.8330

테이블 ID	banded-totality-629:p_johnkoo.bssid_list_1906_1908
테이블 크기	666.18GB
열 개수	42,783,810

## 2 단계: Clustering 을 통한 위치 추정

```
OPTIONS ( library = `gs:/.../bigquery_udf_libs/loplat_bq_udf_lib.js` );
```

```
CREATE TEMPORARY FUNCTION  
  clusterLocations (s STRING)  
  RETURNS STRUCT<lat FLOAT64, lng FLOAT64>  
  LANGUAGE js AS """  
  return clusterLocations(s); """;
```

```
SELECT bssid, cluster.lat as lat, cluster.lng as lng  
FROM  
  (SELECT bssid, clusterLocations(locations) as cluster  
  FROM bssid list 3months  
  )
```

# 2 단계: Clustering 을 통한 위치 추정 (코드)

DEVIEW  
2019

`clusterLocations(locations): udf written in java script`

`– pseudo code –`

`clusters = []`

`CLUSTER_RADIUS = 200 meters`

`// location: ts, rss, lat, lng, ...`

`for(location in locations)`

`for(cluster in clusters)`

`if(dist(center of cluster, location) > CLUSTER_RADIUS)`

`insert new cluster into clusters`

`else:`

`adjust center of cluster with location data`

`find the best cluster in clusters`

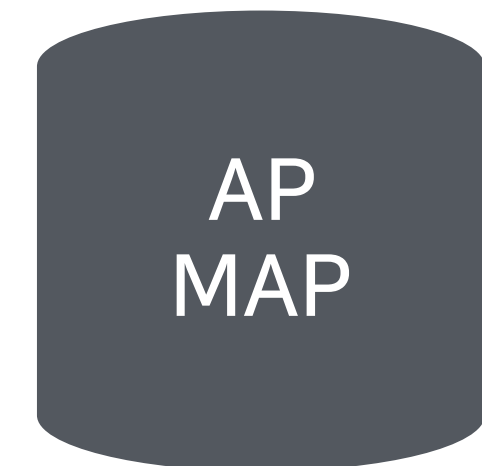
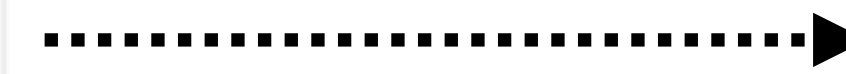
`→ size is dominant and recently observed`

# 2 단계: Clustering 을 통한 위치 추정 (결과)

스키마   세부정보   미리보기

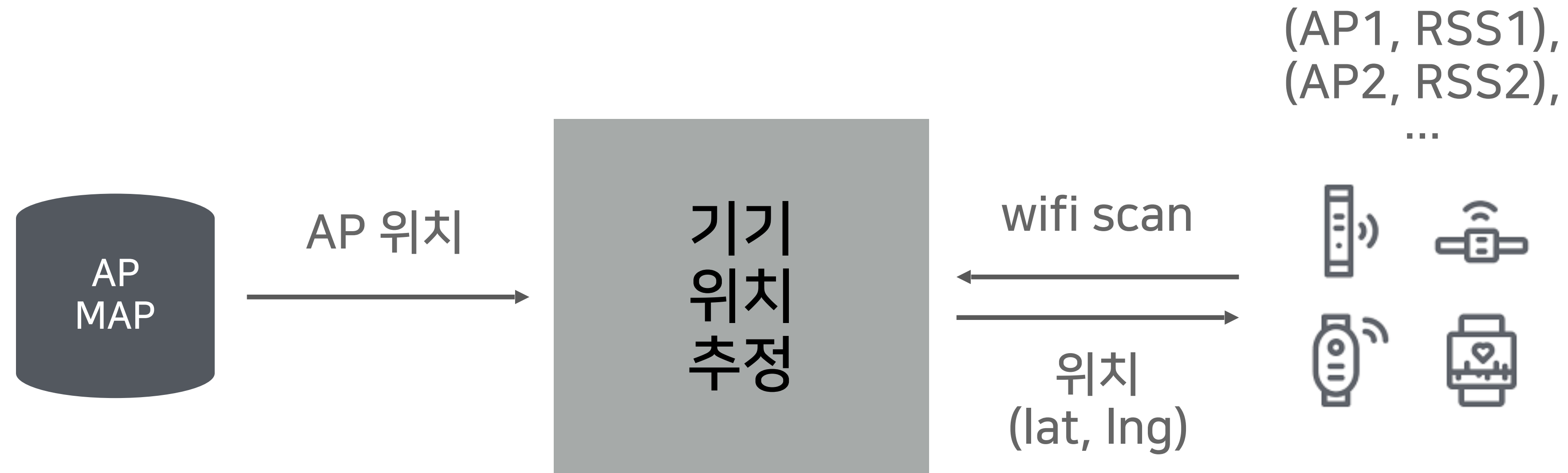
행	bssid	lat	lng	tss	tse	floor
1	ec:e1:a9:34:7e:e2	36.323517	127.419779	90601	90830	0
2	12:23:aa:b6:ac:f6	37.474506	126.701587	90601	90825	0
3	88:3c:1c:d2:15:b3	37.521673	127.047334	90621	90831	0
4	00:23:aa:d1:12:42	37.479408	126.913378	90719	90829	0

to SQL DB



# WPS: 기기 위치 추정

DEVIEW  
2019



AP MAP 으로 부터 위치정보를 가져오고,  
신호 세기를 참고하여 위치 계산  
(Tri-lateration or Weighted Centroid)

# WPS: 기기 위치 추정

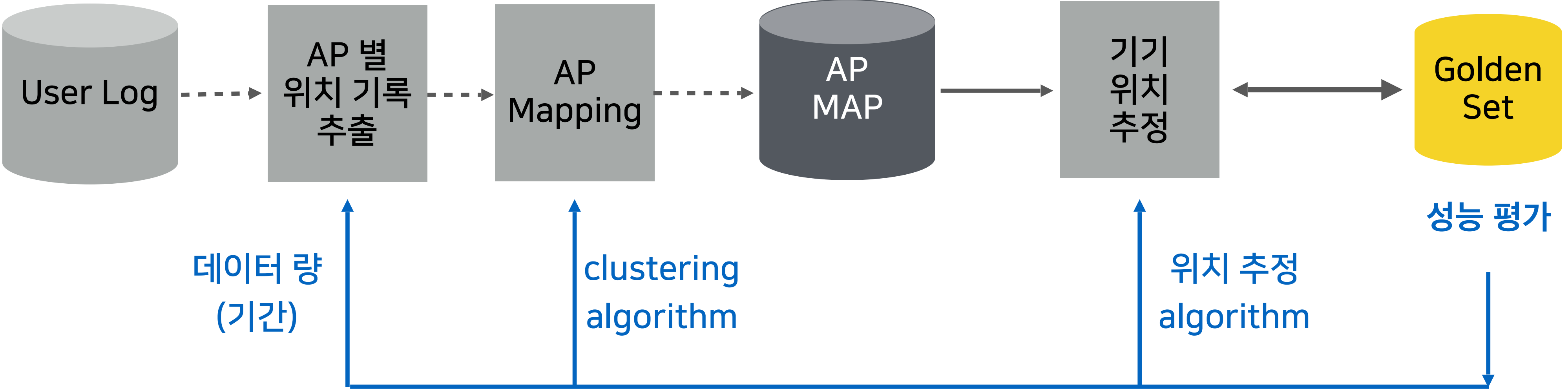
WiFi AP Map 이 완벽했으면 좋겠다.

하지만, 잘못 맵핑된 것이 있을 수도 있다.

그리고, AP Map 을 만들고 난 후에 이사를 간 AP 도 있다.

다시한번 거리기준으로 Clustering 을 하여  
dominant 한 cluster 의 AP 만 사용하여 (outlier 제거)  
위치 추정 정확도를 높인다.

# Optimization with Golden Set





# 그래서 얼마나 정확하냐?

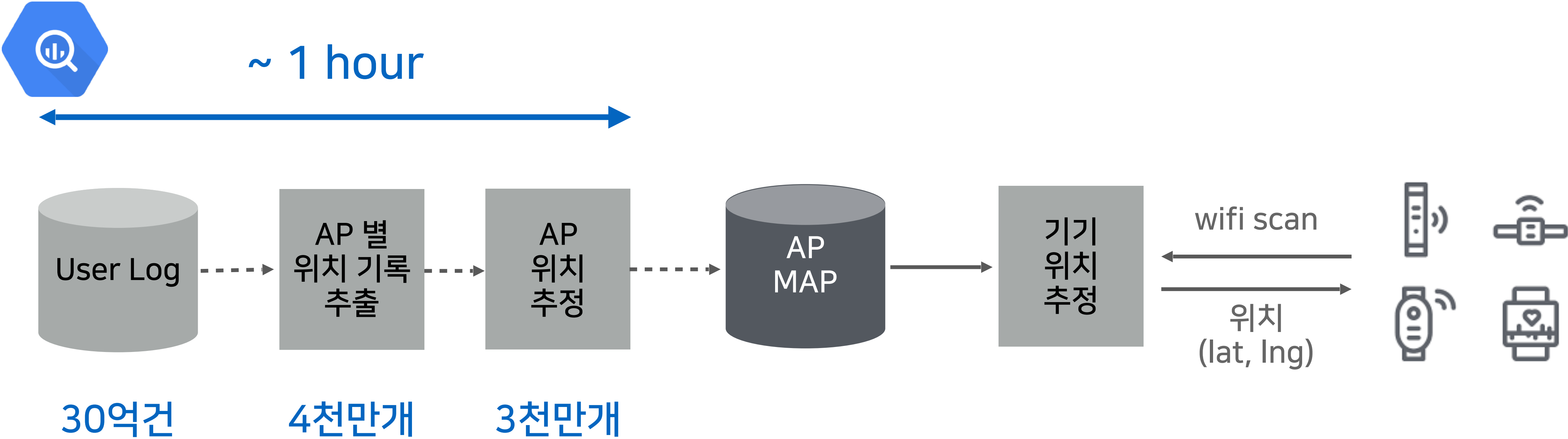
golden set : 19272 개

인식 못함: 31개

mean error: 18.25

median error: 16.28

# WPS Building System



<https://developers.loplat.com/wps/>

Thank You

# 시간이 된다면... 생각해볼 만한 문제

DEVIEW  
2019

2D -> 3D

기압계

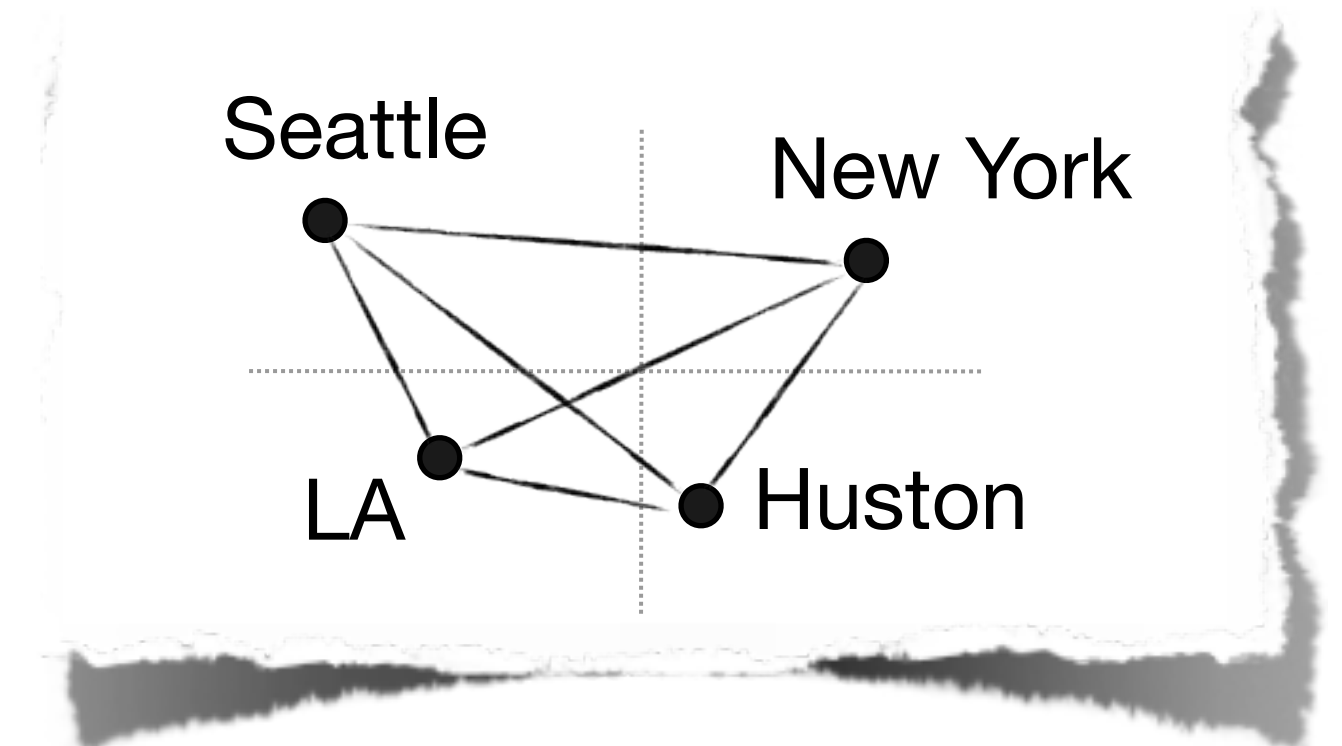
## MDS (Multi-dimensional Scaling) 기법 활용

	Seattle	LA	Houston	New York
Seattle	0	959	1891	2408
LA		0	1374	2451
Houston			0	1374
New York				0

- $p_i = (p_{ix}, p_{iy})$  : 각 노드의 위치
- $f(p_{ij})$  :  $p_i$  와  $p_j$  사이의 거리
- $d_{ij}$  : dissimilarity

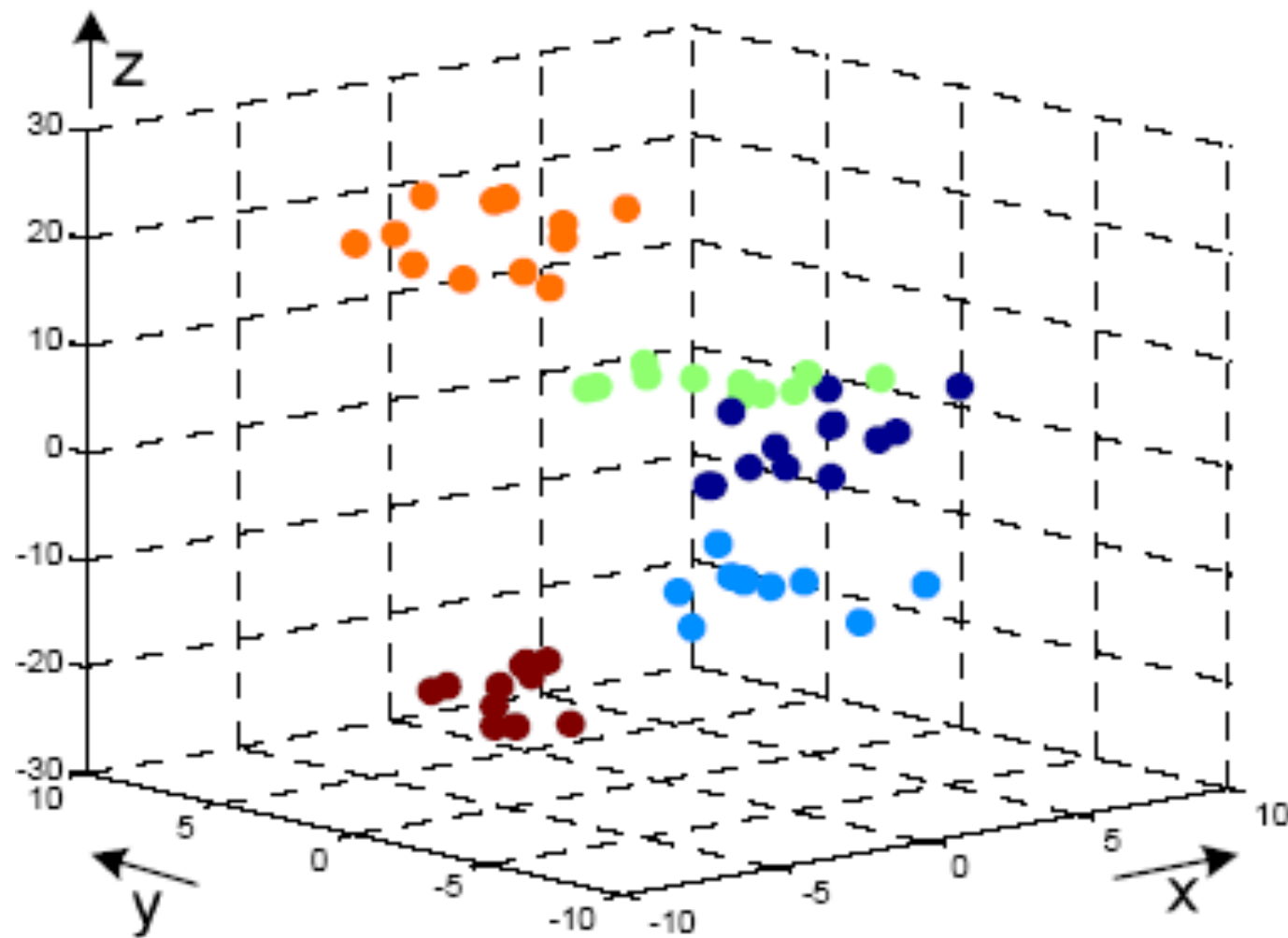
then, find all  $p_i$  to minimize

$$\sqrt{\frac{\sum_i \sum_j [f(p_{ij}) - d_{ij}]^2}{scale\_factor}}$$



## MDS (Multi-dimensional Scaling) 기법 활용

- $p_i = (p_{ix}, p_{iy}, p_{iz})$  :  $AP_i$  의 위치
- $d_{ij}$  : wifi scan 에서 보여지는 AP 들의 관계를 이용하여 추정



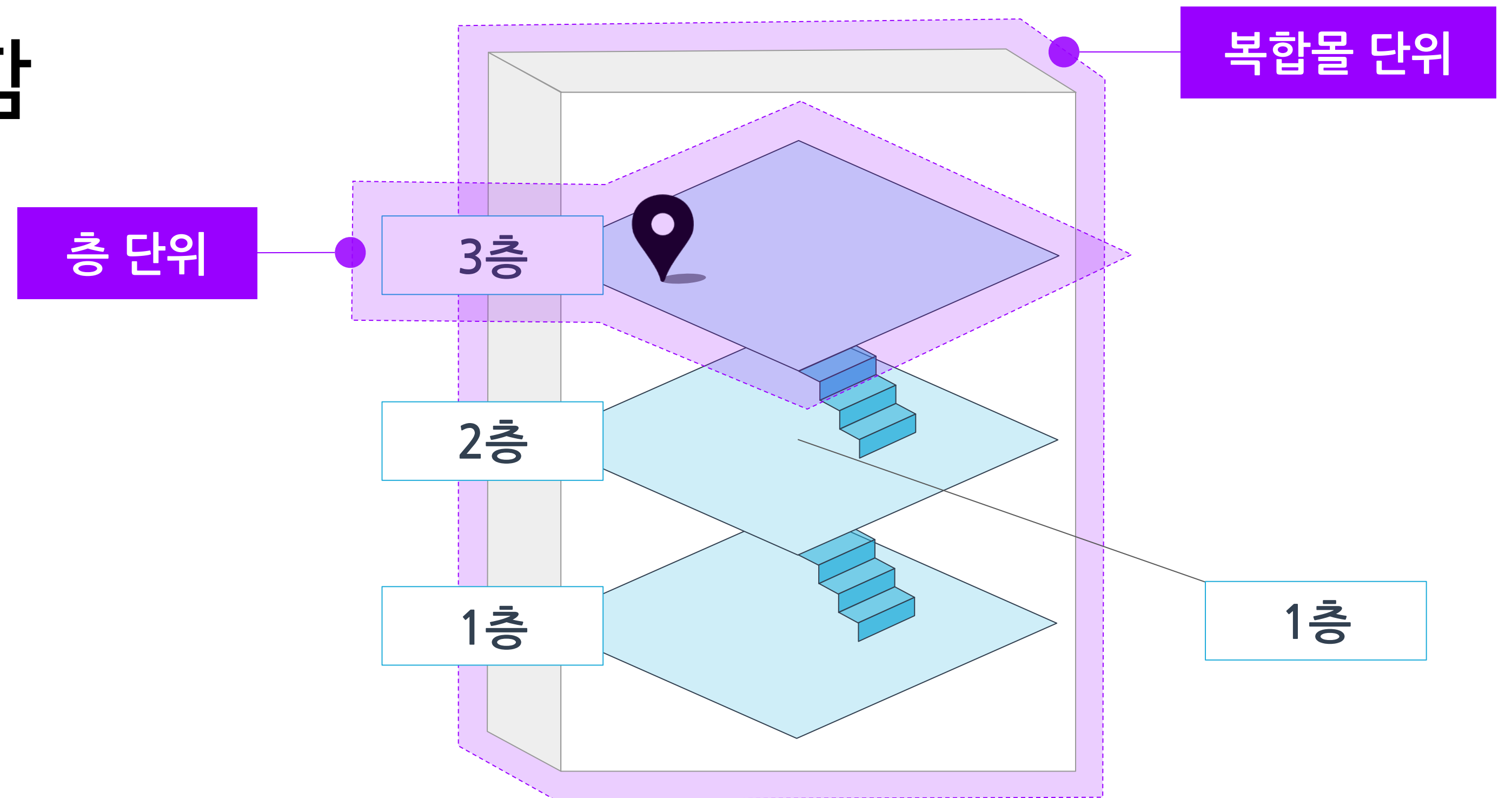
단일 건물에서는 OK

건물 구별을 못하는 상황에서는  
쉽지 않은 방법

# 2.5D

노가테크의 기술력으로 쌓은 정보를 활용

유저로그에 인식된 층 정보 포함



ex) 백화점, 대형마트, 쇼핑몰 등



Thank You