

# Global Grid System (H3) 를 이용한 Last Mile 분야의 공간데이터 관리와 Mapping

손진, 전한샘

Coupang

# CONTENTS

1. 배송영역 관리 시스템에서의 Global Grid System 도입의 필요성
2. 영역 데이터를 H3로 관리할 때의 어려움과 해결 방식
3. What's Next? + Q&A

Copyright © 2022 Coupang, Inc. All rights reserved. 사용된 모든 쿠팡 상표와 쿠팡 로고 및 서비스마크는 미국 및 기타 국가에 등록되어 있는 Coupang, Inc. 및/또는 그 계열사(통칭하여 "쿠팡"이라 함)의 재산입니다. 그 외 여기서 언급된 회사는 식별 목적으로만 언급된 것으로, 쿠팡은 사용된 기업명이 동 회사의 등록 상표일 수 있으며 해당 회사가 단독으로 동 상표에 대한 독점적 소유권을 가진다는 것을 인정합니다.

여기에 포함된 정보는 임직원으로서 저자 본인의 개인적 경험을 바탕으로 한 것으로 쿠팡의 견해나 의견을 나타내는 것이 아님을 밝혀 둡니다. 쿠팡은 여기에 포함된 정보의 적절성이나 공정성, 정확성, 완전성에 대해 확인하지 않았으며 그에 대해 어떠한 진술도 하지 않습니다.

# 1. 배송영역 관리 시스템에서의 Global Grid System 도입의 필요성

# 1.1 기존 배송 체계가 당면한 어려움

쿠팡은 한국에서 가장 많이 사용되는 쇼핑앱입니다. 쿠팡을 이용하는 활성고객은 1800만 명에 달하며, 이 중 절반은 와우회원입니다.

활성 고객

1800만 명

와우회원

900만 명

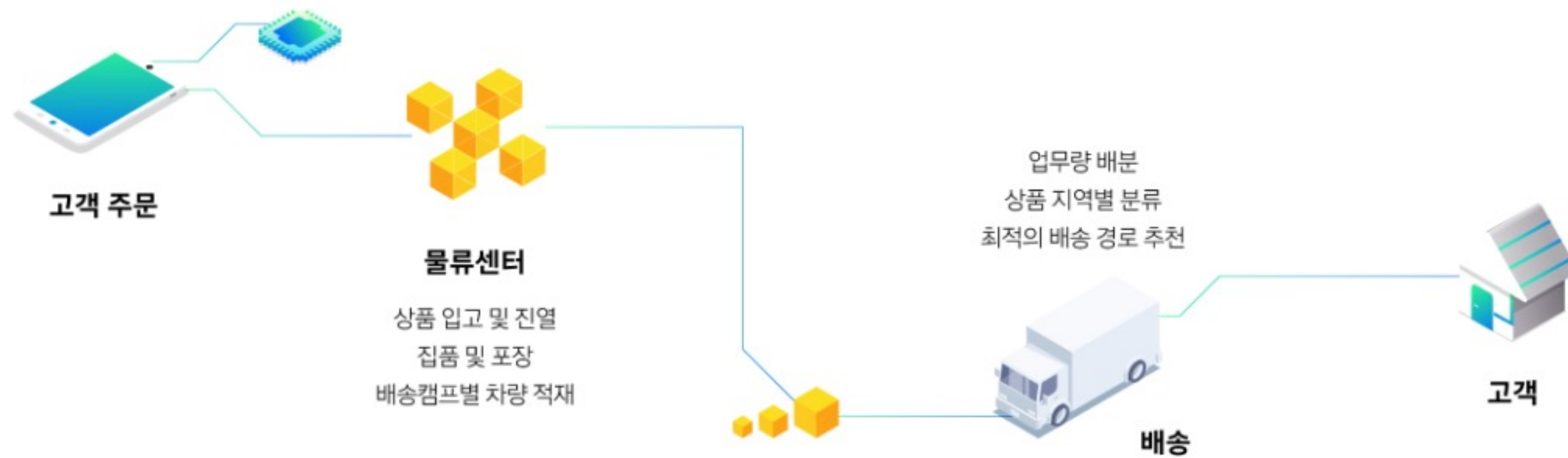


# 1.1 기존 배송 체계가 당면한 어려움

## 고객 WOW 경험

쿠팡은 수백만 종에 달하는 로켓배송 상품을 익일, 새벽, 당일에 배송하는 편리한 서비스를 제공합니다. 대한민국의 국민의 70%는 쿠팡 배송센터 10km 내 거주 중입니다.

제주도에서 배를 타고 들어가야 하는 우도 주민들까지 로켓배송 서비스를 누리고 있습니다.



# 1.1 기존 배송 체계가 당면한 어려움



아파트 헬리오시티 ☆  
9510세대 / 총 84동 / 2018.12.28 / 61.37㎡ ~ 194.2㎡

단지 정보

세대수	9510세대(기타임대 1401세대 포함, 총84개동)	저/최고층	10층/35층
사용승인일	2018년 12월 28일	총주차대수	12602대(세대당 1.32대)
용적률	285%	건폐율	19%





배송직원 1명이 배송가능한 단위로 분할 필요

# 1.1 기존 배송 체계가 당면한 어려움

## Problem Statement

우편번호/법정동 등의 영역만으론 배송영역을 효과적으로 관리하기 힘들다.

→ 배송기사가 효율적으로 배송할 수 있는 단위의 배송영역 설정이 필요하다.

## Alternatives

1. 텍스트 기반의 주소목록으로 배송영역을 관리
2. 폴리곤으로 배송영역을 직접 설정하여 관리

# 1.1 기존 배송 체계가 당면한 어려움

## 텍스트 기반의 주소목록으로 배송영역을 관리

- 영역별 주소목록 관리시에 누락이 발생하거나, 과거/최신 주소도 처리해야함

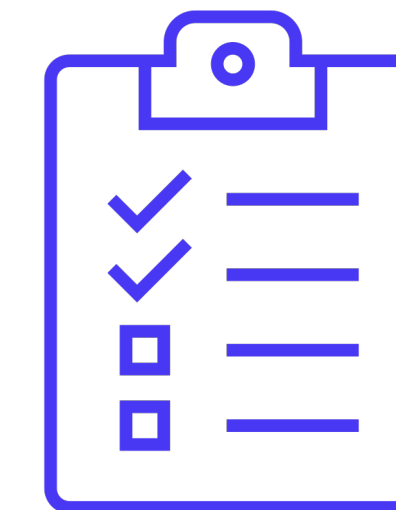
우편번호 05698

05698 분할 우편번호

이름	별칭	관리
05698-002	헬리오시티 101 ~ 106	
05698-003	헬리오시티 107 ~ 112	
05698-008	헬리오시티 501 ~ 509	
05698-009	헬리오시티 510 ~ 517	
05698-011	헬리오 201~207	
05698-012	헬리오 208~219	

05698 전체 주소

전체					
<input type="checkbox"/>	도로명	주번	분할명	건물동	관리
<input type="checkbox"/>	삼학사로2길	49	001		
<input type="checkbox"/>	송파대로	345			
<input checked="" type="checkbox"/>	송파대로	345	002	101, 102, 103, 104, 105, 106	
<input type="checkbox"/>	송파대로	345	003	107, 108, 109, 110, 111, 112	
<input type="checkbox"/>	송파대로	345	011	201, 202, 203, 204, 205, 206, 207	
<input type="checkbox"/>	송파대로	345	012	208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219	
<input type="checkbox"/>	송파대로	345	013	301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311	



주소목록 누락 관리



과거/최신 주소록 확보



# 1.1 기존 배송 체계가 당면한 어려움

## 텍스트 기반의 주소목록으로 배송영역을 관리

- 검색/정제 되지 않는 주소로의 배송이 어려움

우편번호 05698

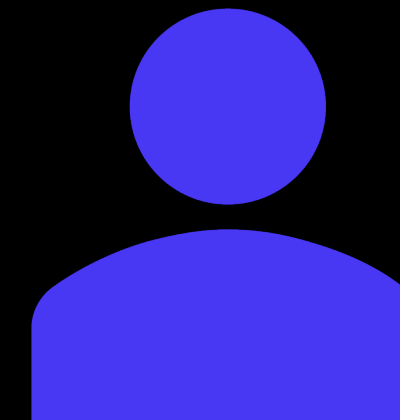
05698 분할 우편번호

이름	별칭	관리
05698-002	헬리오시티 101 ~ 106	
05698-003	헬리오시티 107 ~ 112	
05698-008	헬리오시티 501 ~ 509	
05698-009	헬리오시티 510 ~ 517	
05698-011	헬리오 201~207	
05698-012	헬리오 208~219	

05698 전체 주소

전체					
<input type="checkbox"/>	도로명	주번	분할명	건물동	관리
<input type="checkbox"/>	삼학사로2길	49	001		
<input type="checkbox"/>	송파대로	345			
<input checked="" type="checkbox"/>	송파대로	345	002	101, 102, 103, 104, 105, 106	
<input type="checkbox"/>	송파대로	345	003	107, 108, 109, 110, 111, 112	
<input type="checkbox"/>	송파대로	345	011	201, 202, 203, 204, 205, 206, 207	
<input type="checkbox"/>	송파대로	345	012	208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219	
<input type="checkbox"/>	송파대로	345	013	301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311	

서울시 구파곡 구파로 11-1 1층 정문 옆  
골목 안쪽으로 쪽 들어오시면 보이는 옆문  
앞에 배송부탁드려요



# 1.1 기존 배송 체계가 당면한 어려움

## 텍스트 기반의 주소목록으로 배송영역을 관리

- 배송지역의 주소정보가 존재하지 않는 경우 영역설정이 불가능함



개별건물 정보가 없는 건물군



신축 아파트 단지



군사/보안시설



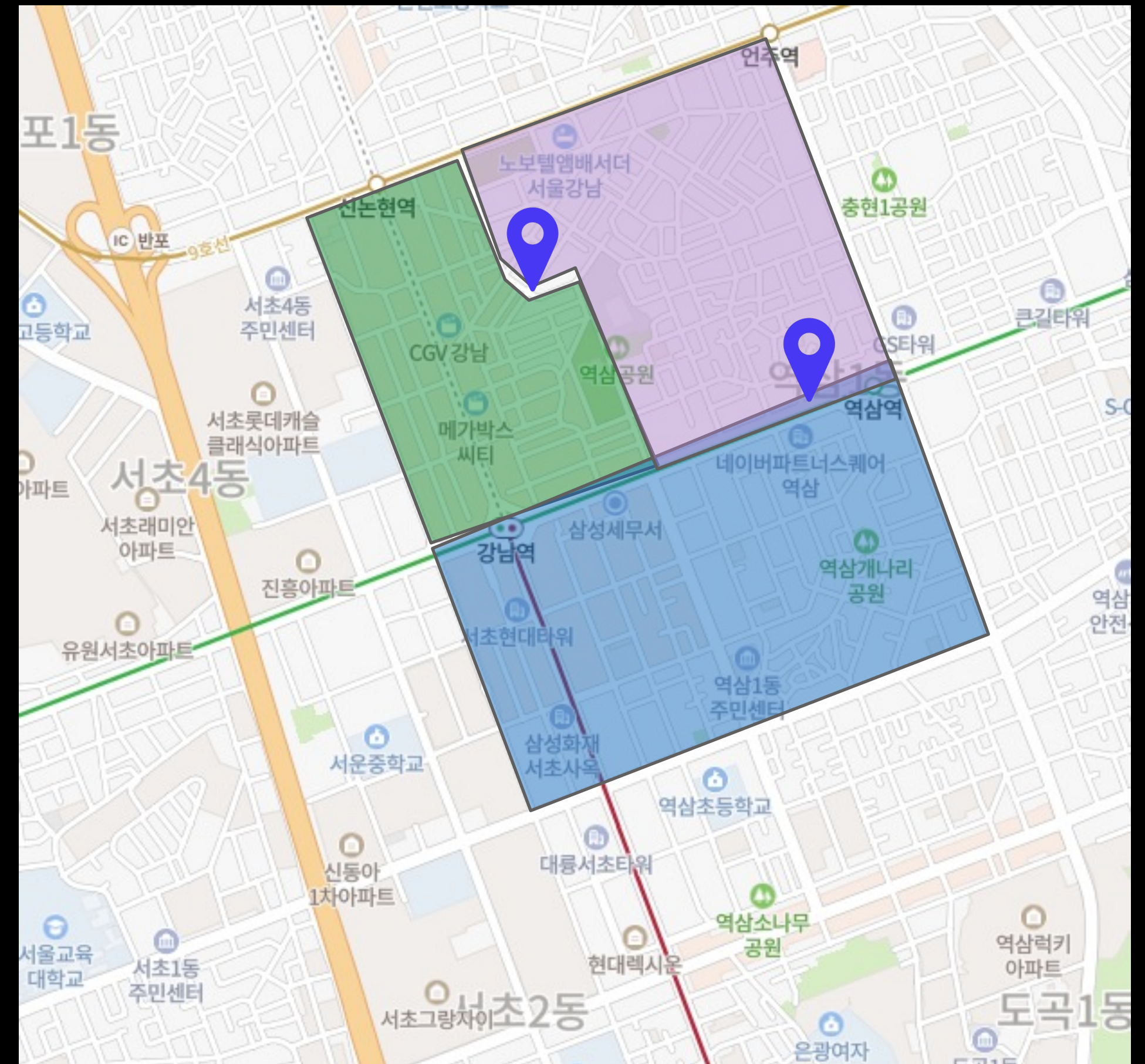
주소DB가 없는 지역

# 1.1 기존 배송 체계가 당면한 어려움

## 일반적인 폴리곤을 활용한

## 배송영역의 분할 및 관리

- 일반 사용자 입장에서 Geometry Tool을 정교하게 사용하기 쉽지 않음
- 중첩지역, 누락지역이 쉽게 발생하고 보완하기 어려움



# 1.2 Global Grid System의 필요성

## 지구 전체를 모자이크 형태로 공간을 구분하는 시스템

- 절대적인 경/위도 좌표를 특정 Grid 로 변환이 가능

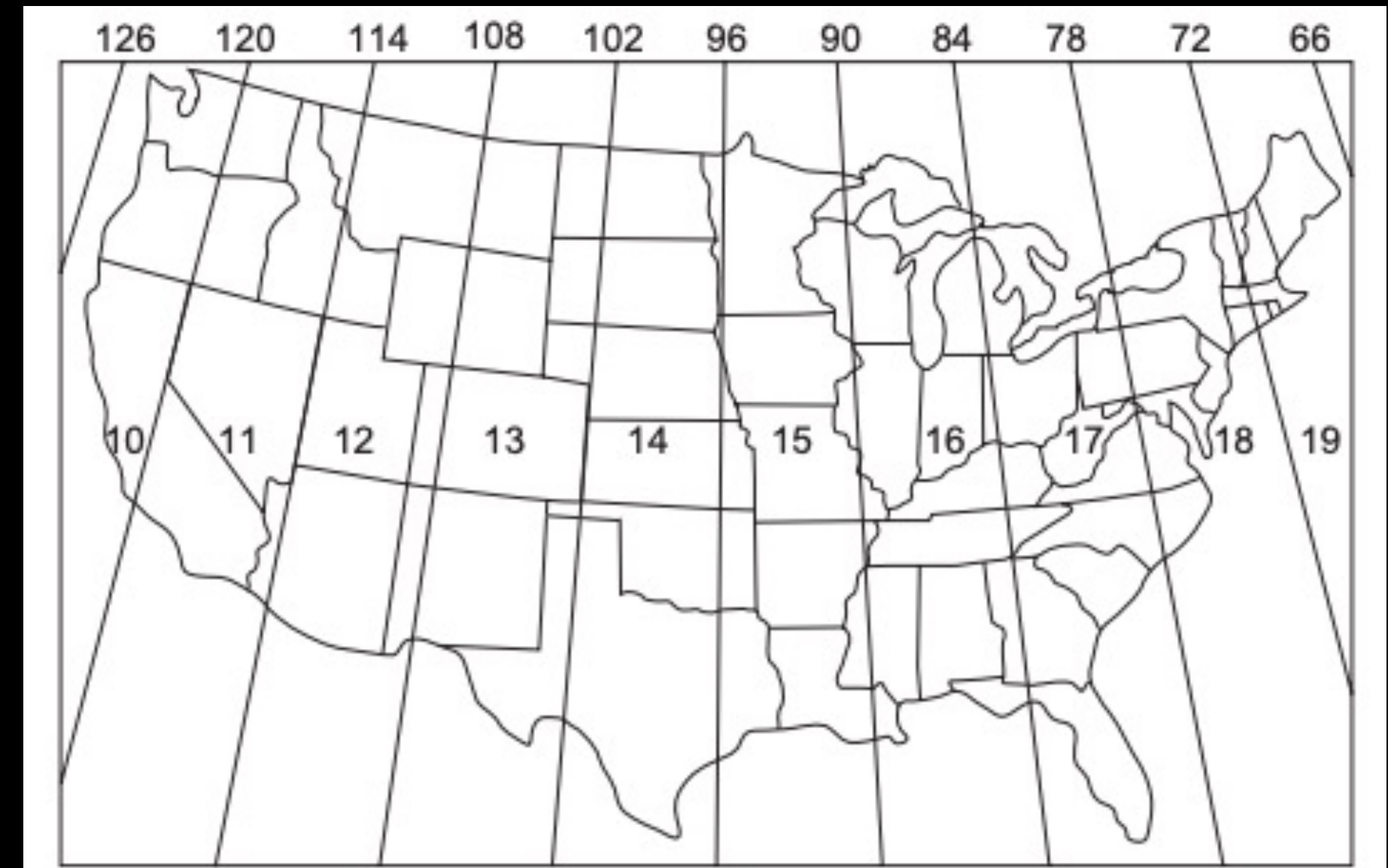
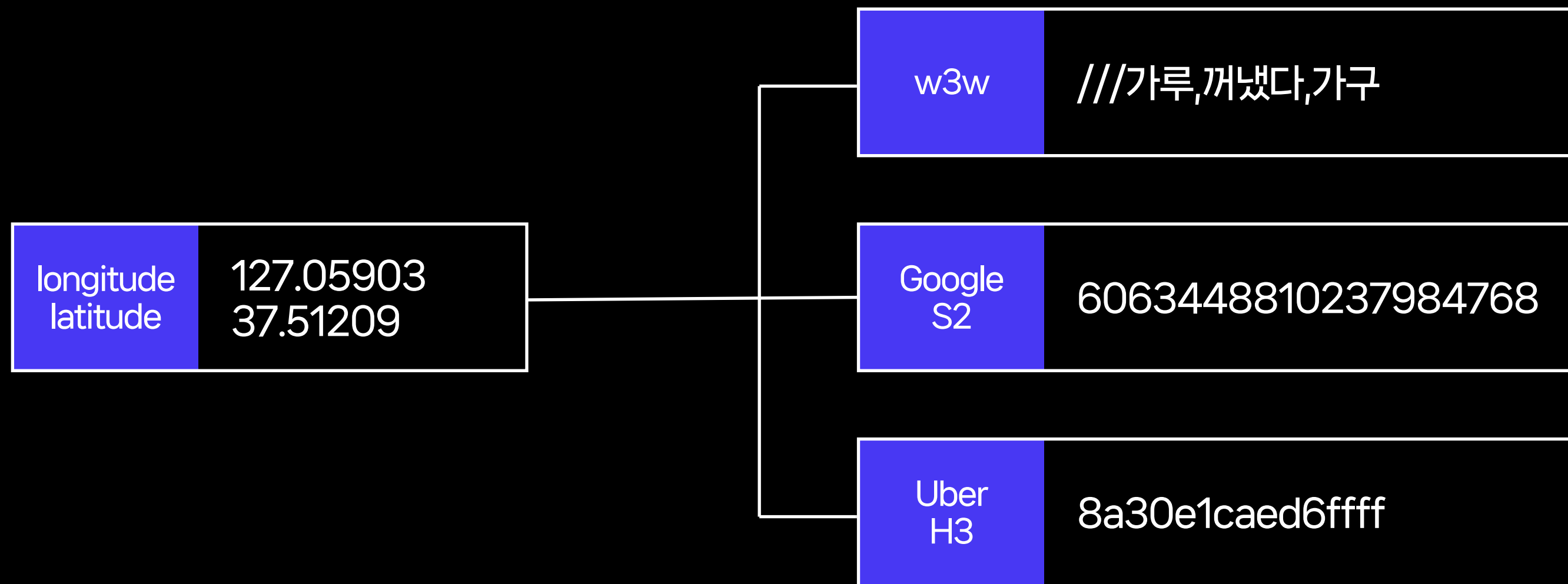
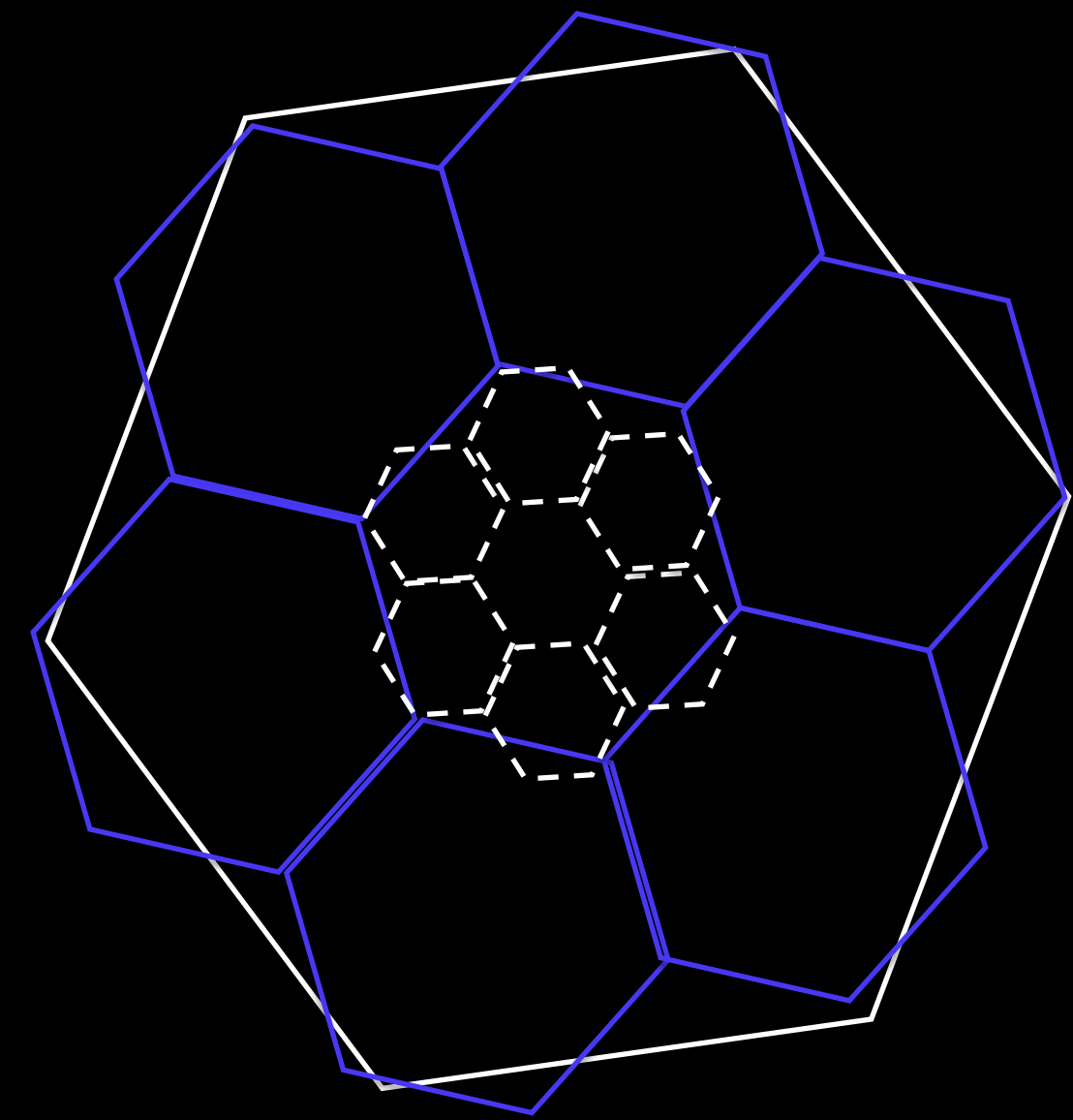
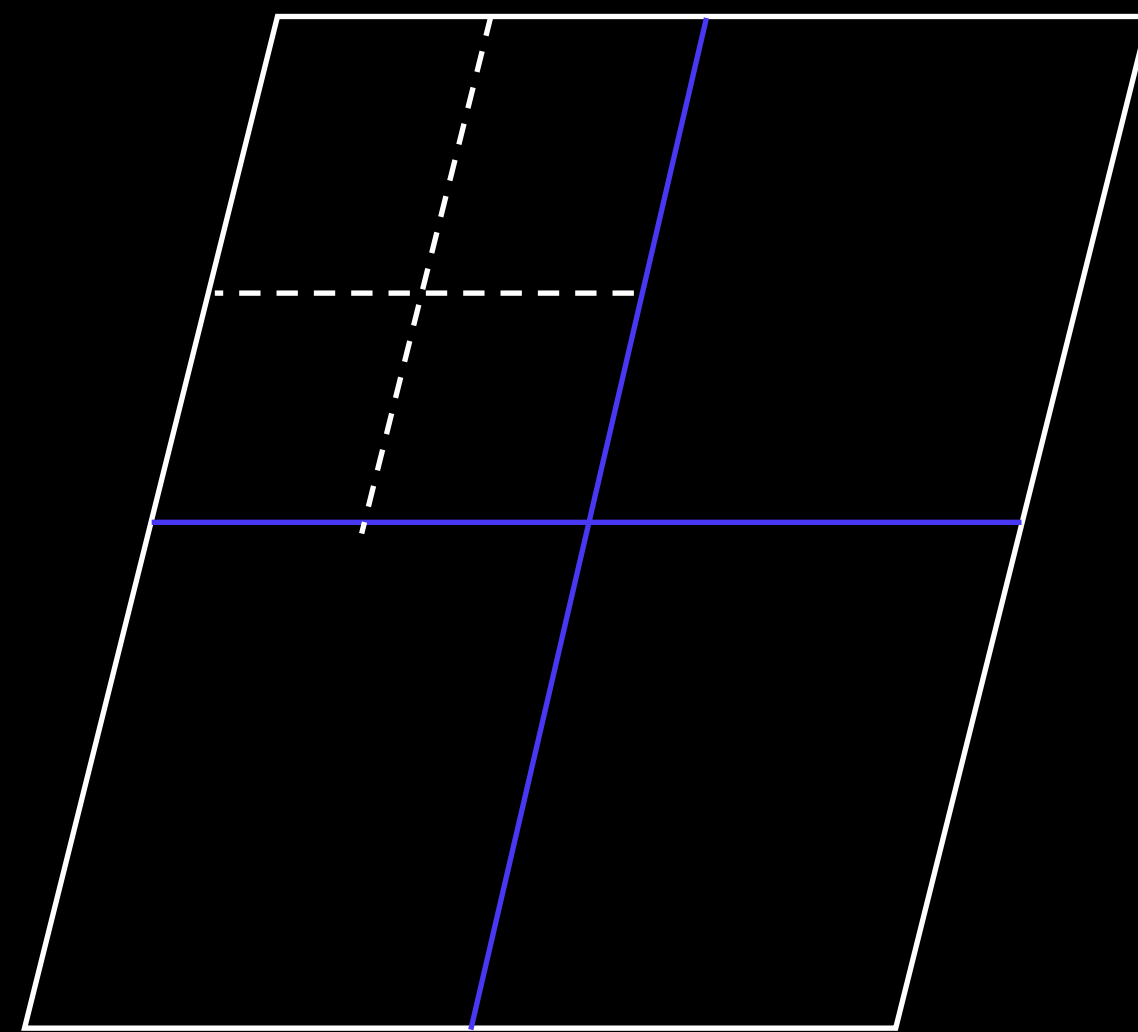
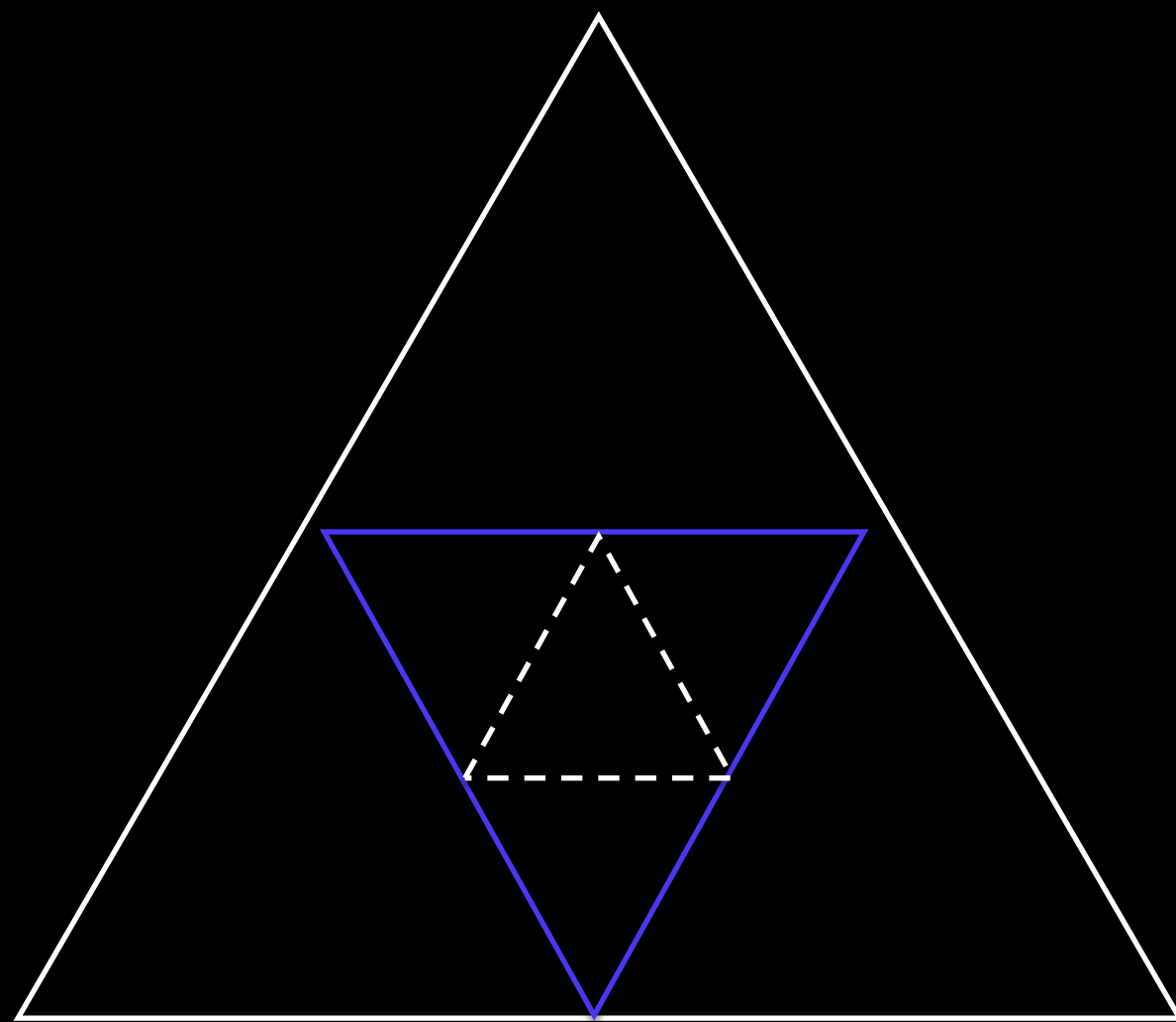


Figure 1. The Universal Transverse Mercator grid that covers the conterminous 48 United States comprises 10 zones—from Zone 10 on the west coast through Zone 19 in New England.

# 1.2 Global Grid System의 필요성

## Hierarchical Global Grid System

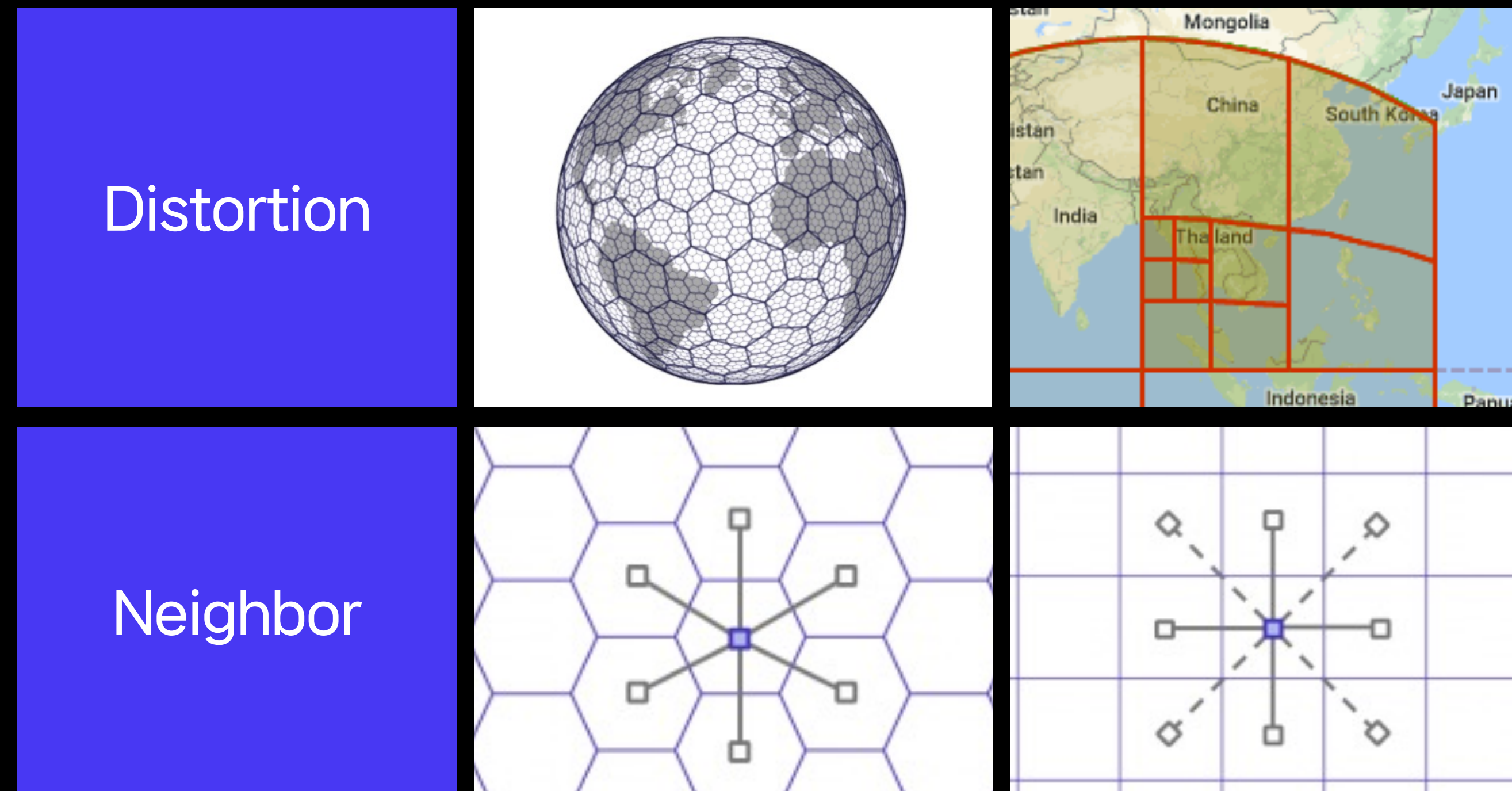
- 상위/하위 레벨의 grid가 1:N으로 mapping
- 필요한 공간해상도에 따라 자유롭게 aggregation 가능



# 1.2 Global Grid System의 필요성

## Hierarchical Global Grid System

- Uber H3 vs Google S2



# 1.2 Global Grid System의 필요성

## Hierarchical Global Grid System

- H3 Resolution-level

0-15까지 16개의 단계로 Grid의 크기가 달라짐

Resolution	Area (m <sup>2</sup> )	Edge (m)	Total number of grids to cover the world
10	15,047.502	659.07807	33.8B
11	2,149.643	24.910561	237B
12	307.092	9.415526	1.66Tr
13	43.870	3.559893	11.6 Tr
14	6.267	1.348575	81.3 Tr

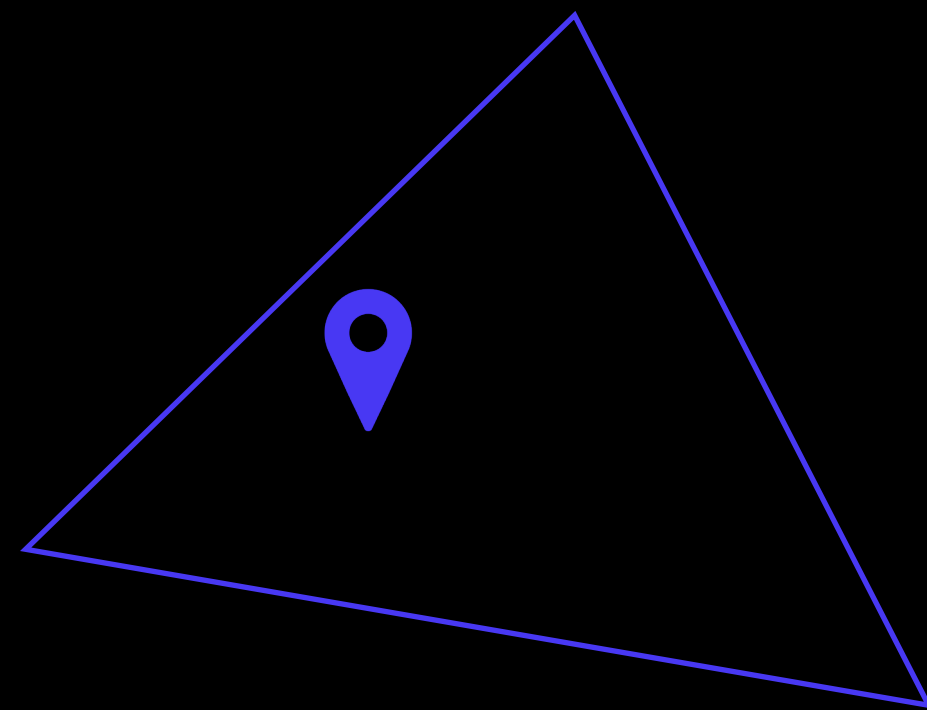


H3 grids of point (127.05903 37.51209)  
in 10, 11, 12 level resolution

# 1.2 Global Grid System의 필요성

## Geometry를 지원하지 않는 DB에서 공간데이터를 다루는 방법

- Geosearch: H3 address를 key로 index search



```
polygon = [  
  [37.11, 127.31], [37.11, 127.32], [37.12, 127.35]  
]
```



```
h3addresses = [  
  "8930e149823ffff", "8930e149827ffff",  
  "8930e149e37ffff", "8930e14983bffff",  
  "8930e14980bffff", "8930e14980ffff",  
  "8930e1498cbffff", "8930e149857ffff"]
```

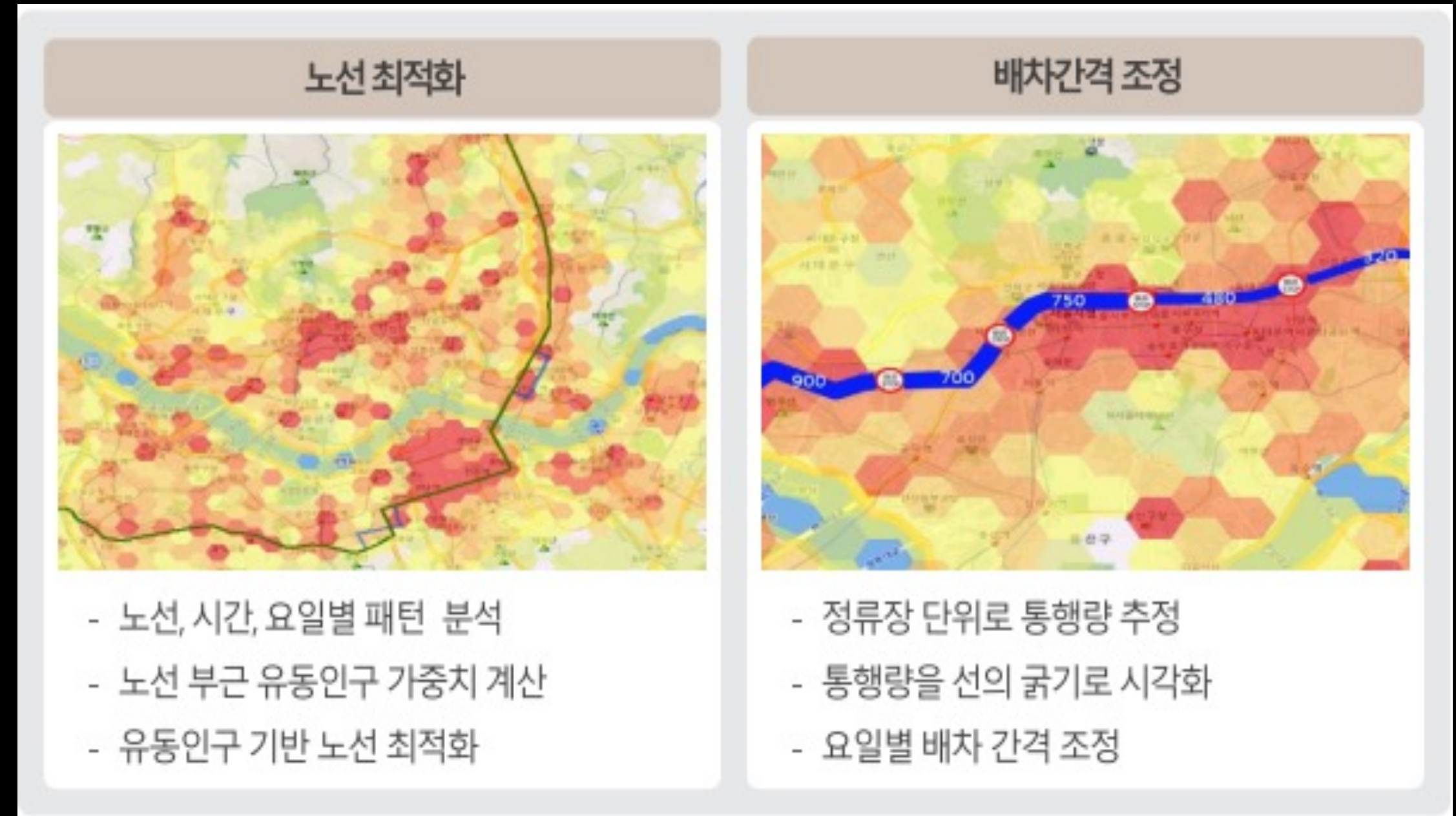


# 1.2 Global Grid System의 필요성

## Case 1 서울특별시, KT:

### 유동인구추정을 통한 심야버스수요분석

- 익명화된 휴대전화사용내역, 버스/택시 승하차 기록을 Hexagon Grid 단위로 가공
- 시계열 분석을 통해 유동인구 추정  
→ 수요분석에 따른 심야노선 의사결정지원

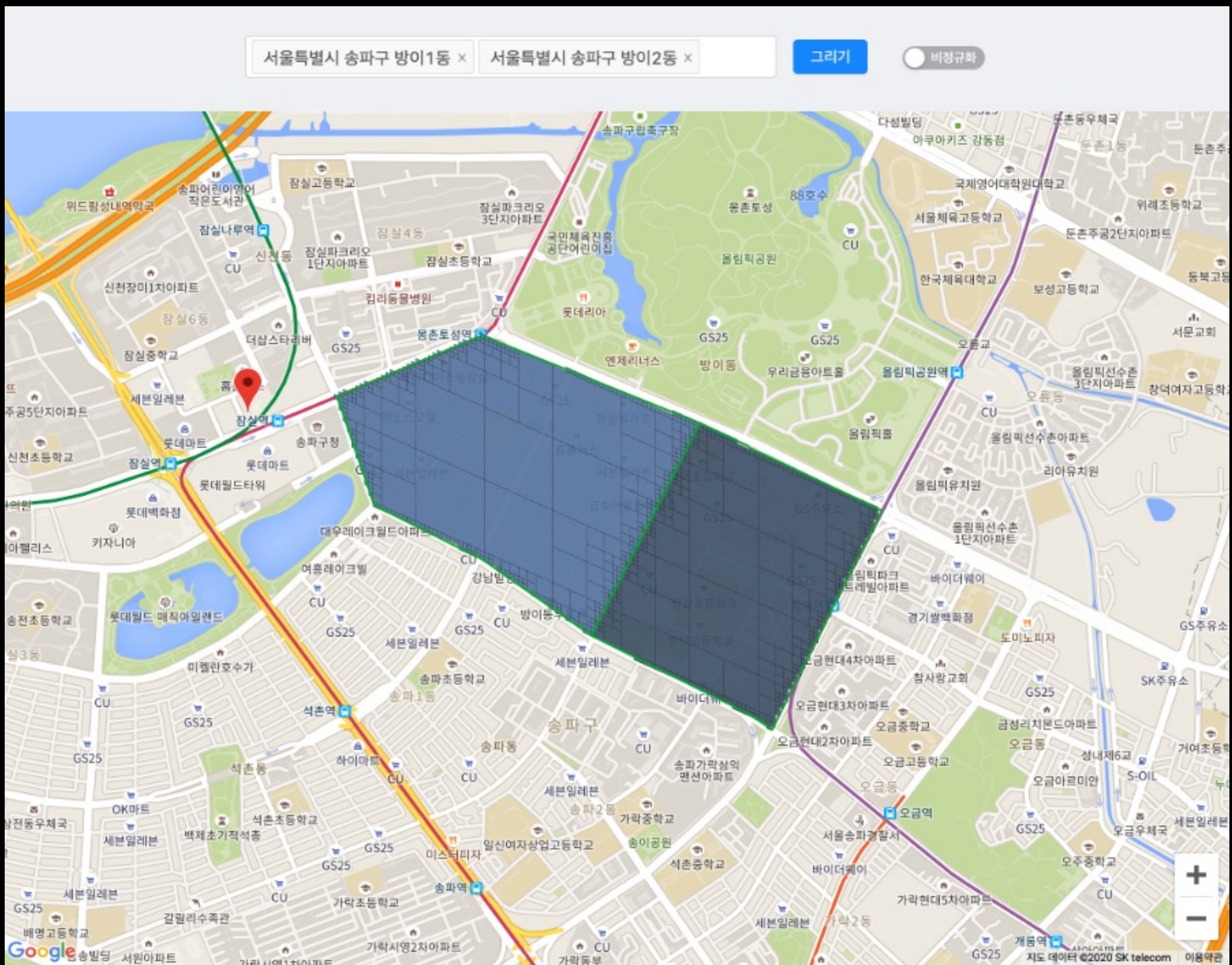


# 1.2 Global Grid System의 필요성

## Case 2 배달의 민족:

### Google S2를 이용한 배달영역관리

- 법정동을 S2 Polygon으로 Migration
- 개별 가게의 영역을 Google S2 Grid 단위로 자유롭게 설정가능



우아한형제들 기술블로그, "가게 배달지역 관리방식 개편 프로젝트"

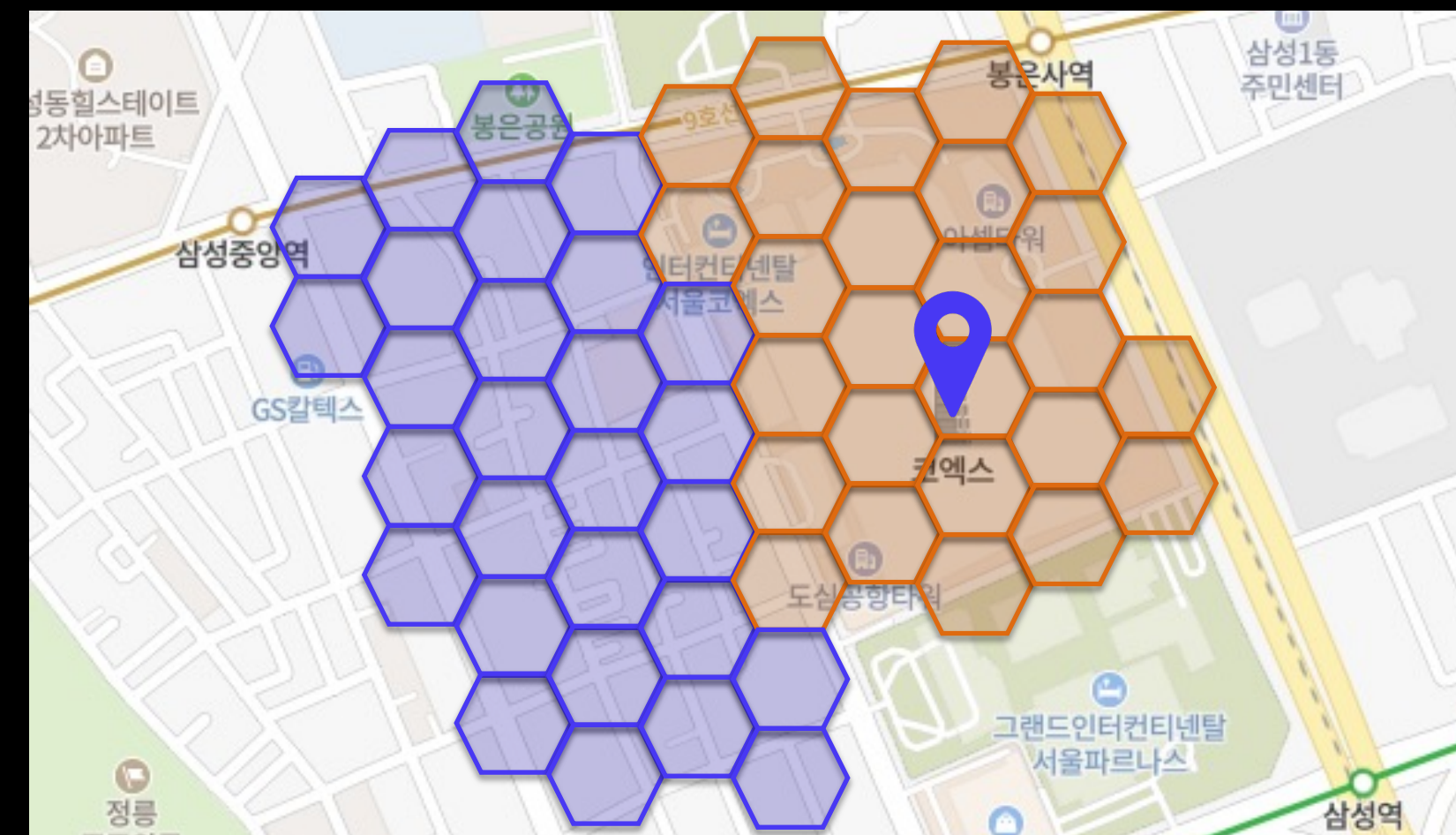


# 1.3 H3 기반의 배송영역/배송지 관리에서의 장점

## 좌표로 배송영역을 효과적으로 검색

- 텍스트 주소를 경/위도 좌표로 변환 후 저장 → 해당 좌표에 해당하는 배송영역을 조회

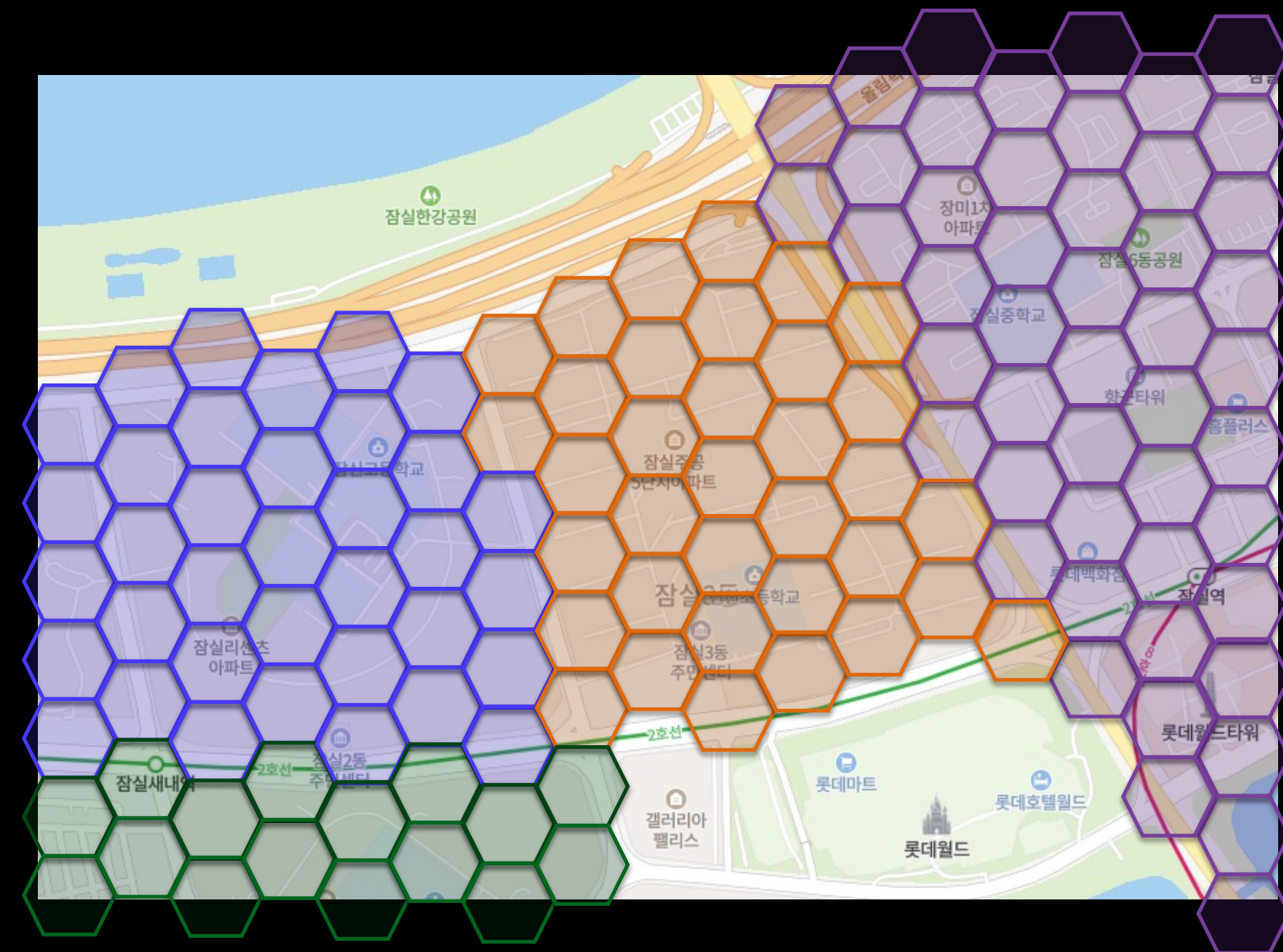
	정부 고시 주소	고객 입력 주소 위치
BEFORE	서울시 쿠팡구 쿠팡로1길 1-1	127.05903 37.51209
↓		
AFTER	서울시 쿠팡구 쿠팡대로 1	127.05903 37.51209



# 1.3 H3 기반의 배송영역/배송지 관리에서의 장점

## MECE (Mutually Exclusive Collectively Exhaustive)

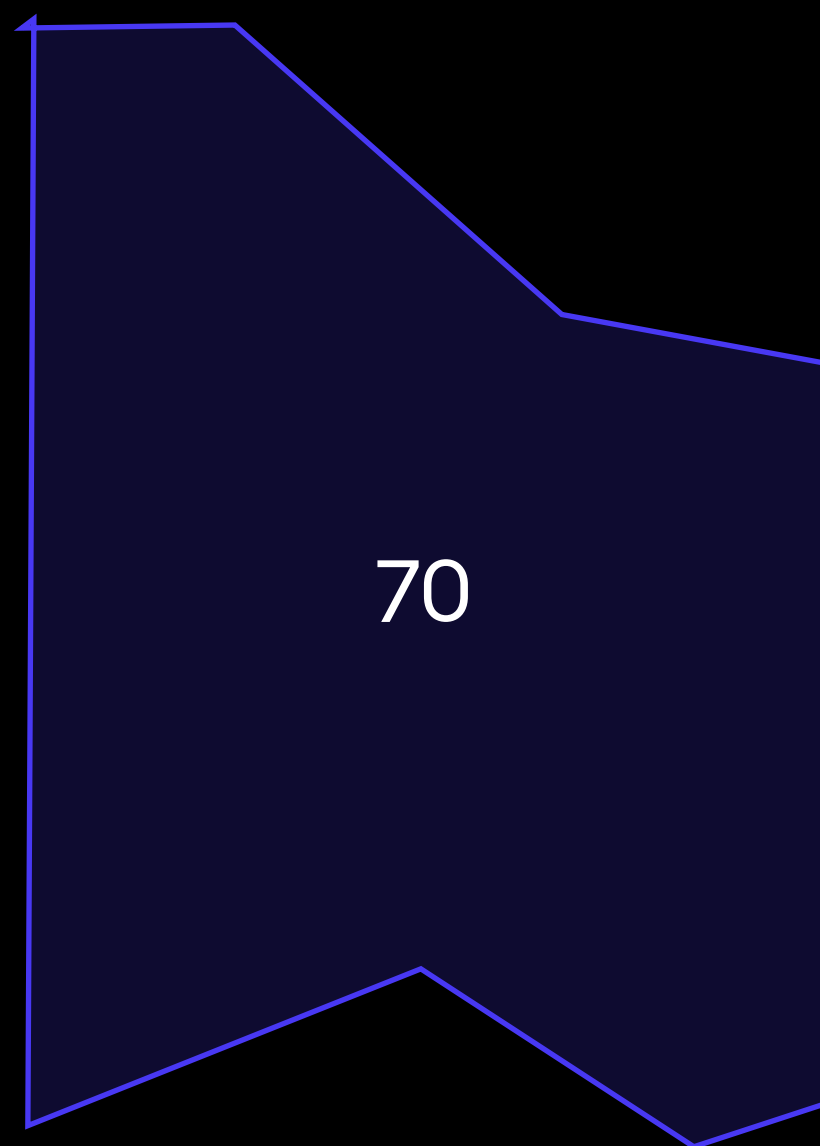
- 전 세계의 모든 지역을 격자 단위로 관리할 수 있음
- 누락/중첩 지역을 명시적으로 빠르게 확인하고 처리가 가능



# 1.3 H3 기반의 배송영역/배송지 관리에서의 장점

## Grid 기반의 통계 데이터 분석

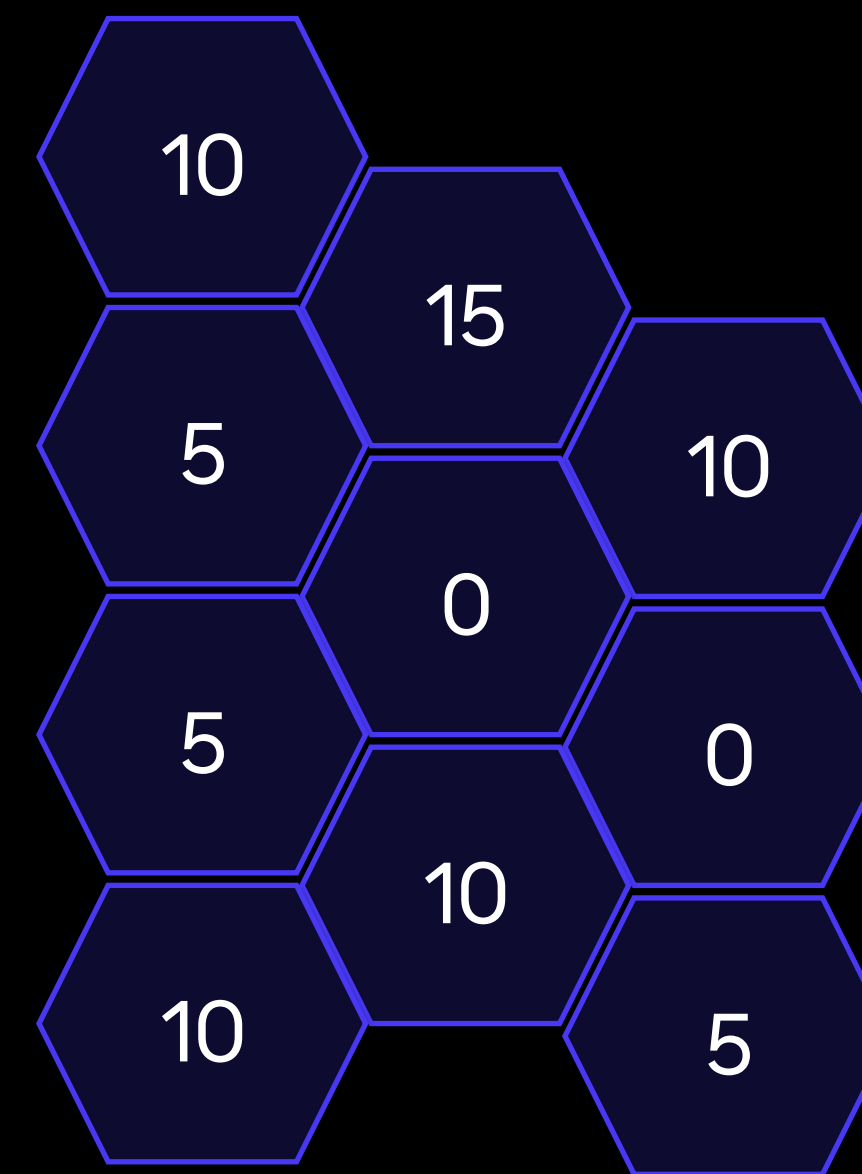
- Grid 단위로 데이터가 적재된 경우 영역이 바뀌더라도 통계분석이 가능



BEFORE BLUE : 70



AFTER BLUE : 120  
ORANGE : 80



BEFORE BLUE : 70



AFTER BLUE : 120  
ORANGE : 80

## 2. 영역 데이터를 H3로 관리할 때의 어려움과 해결방식

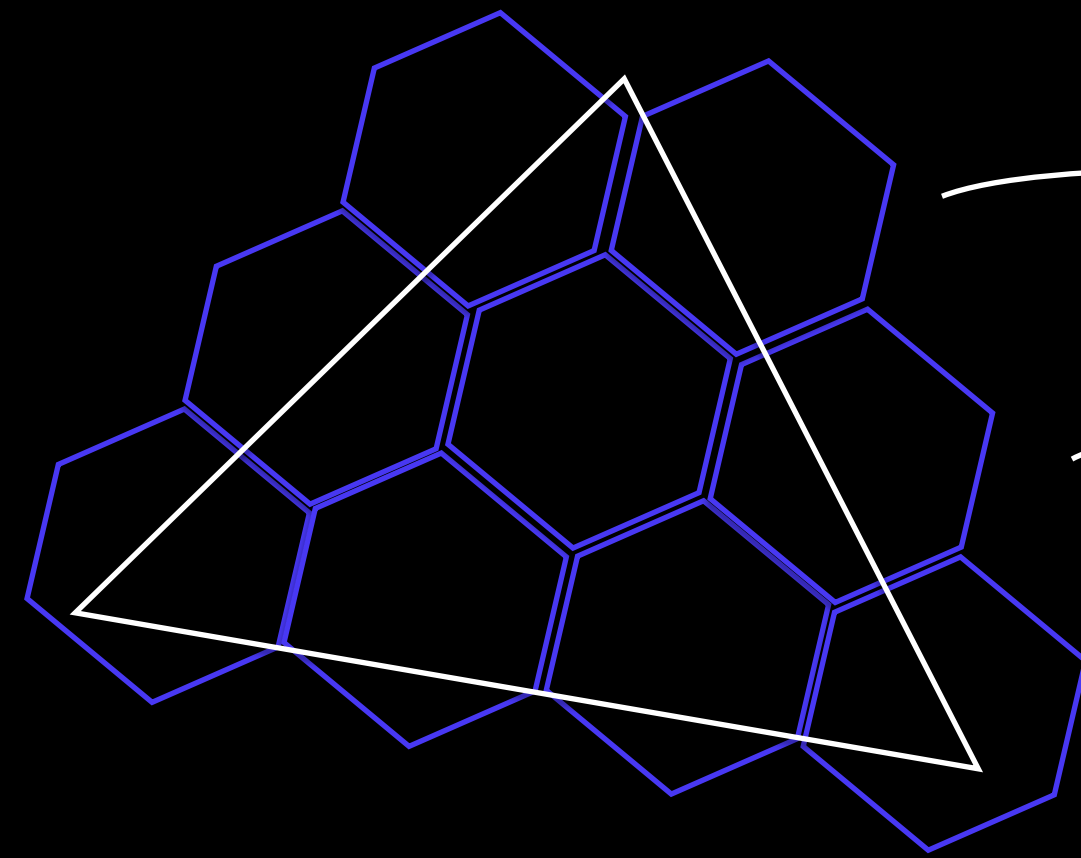
**#1 RDBMS에 Grid를 Row로 저장**



# 2.1 H3 – RDBMS 모델링

## How to

- 12-level Resolution의 하나의 격자를 RDBMS의 row로 1:1 대응



h3Index	group
8abddfda <sup>f</sup>	Group 1
8abddfda <sup>g</sup>	Group 1
8abddfda <sup>a</sup>	Group 2

# 2.1 H3 – RDBMS 모델링

## Problem

- 12-level resolution 은 각 건물을 정교하게 구분하여 표현하지 못함



Resolution	Area (m <sup>2</sup> )	Edge (m)
12	307.092	9.415526
13	43.870	3.559893
14	6.267	1.348575

# 2.1 H3 – RDBMS 모델링

## Problem

- 정교한 표현을 위해 14-level resolution을 도입 시 예상되는 row count

Resolution	Estimations of h3address counts		
	Korea	Taiwan	Japan
11	72M	24M	144M
12	432M	169M	864M
13	3.1B	1.1B	6.2B
14	21.7B	8.3B	43.4B

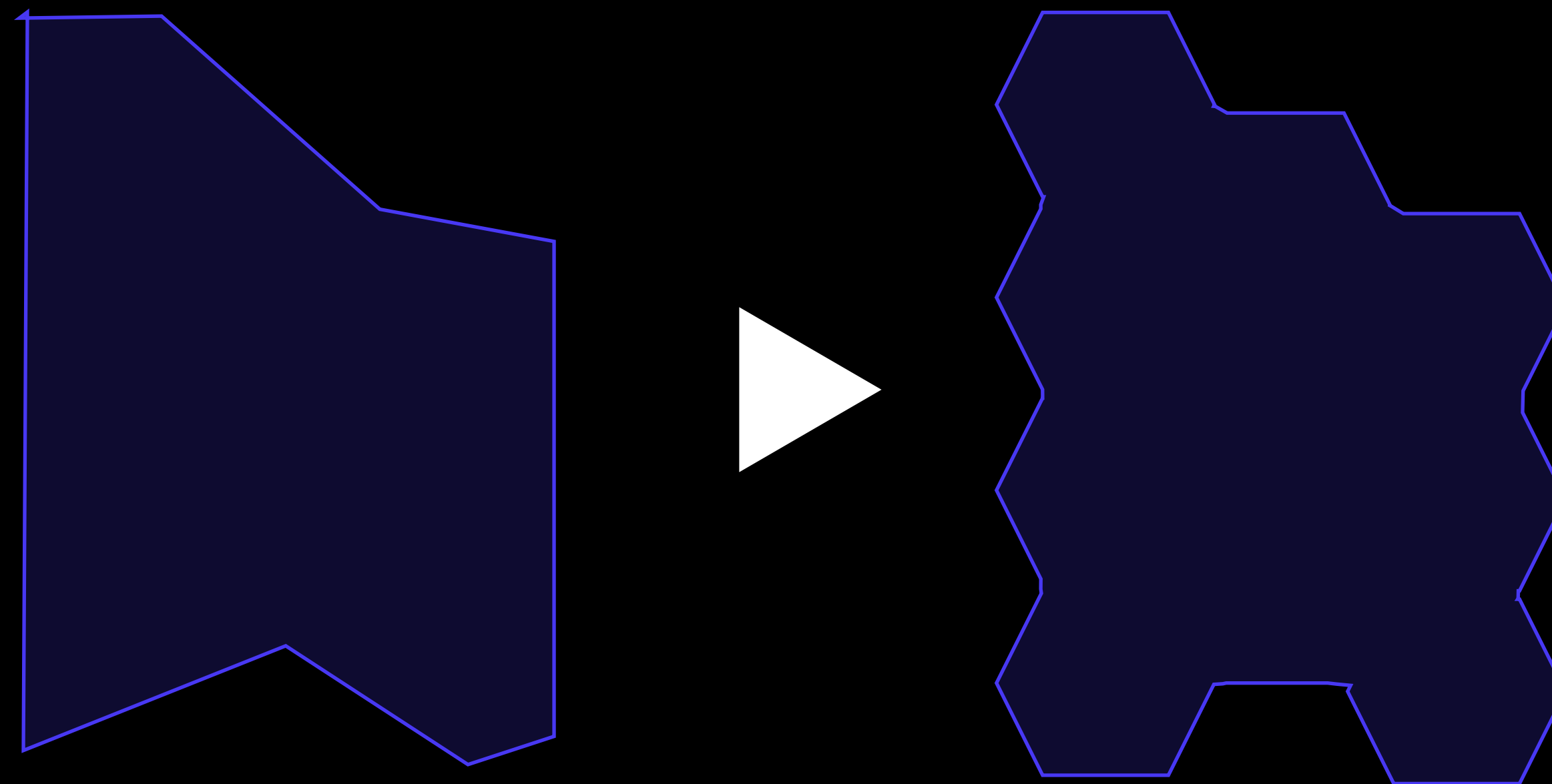
# #2 Grid영역을 다시 Polygon으로 저장

## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## How to

- Polygon -> 격자 모양의 polygon

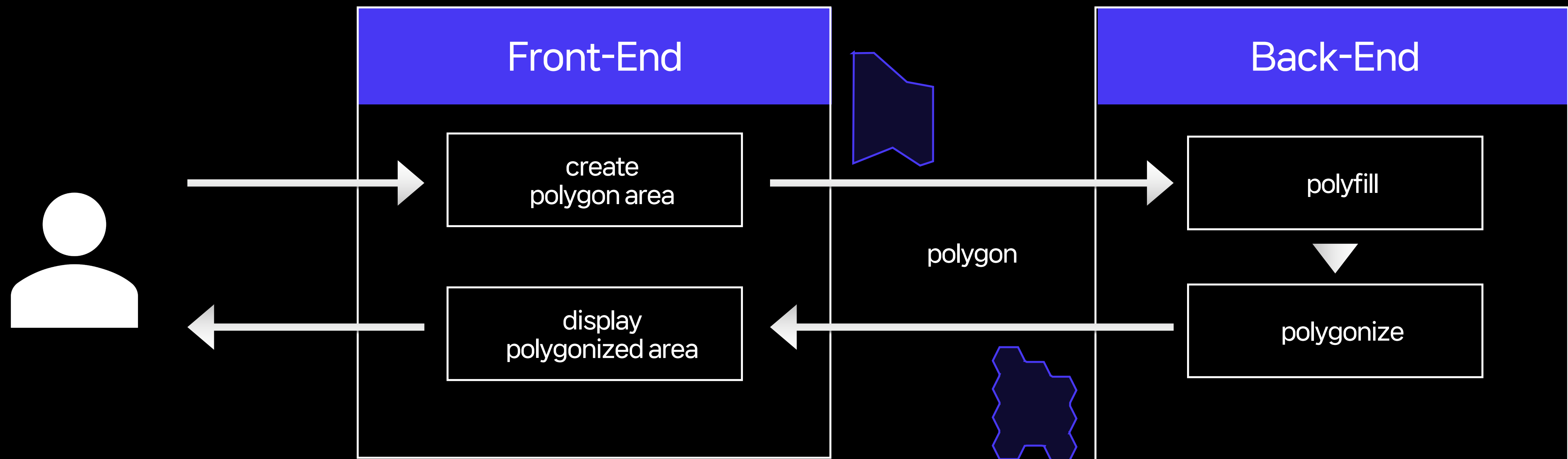


## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## How to

- 서버와 클라이언트의 통신은 polygon 으로
- 서버에서 polygon을 격자 모양의 polygon으로 변환

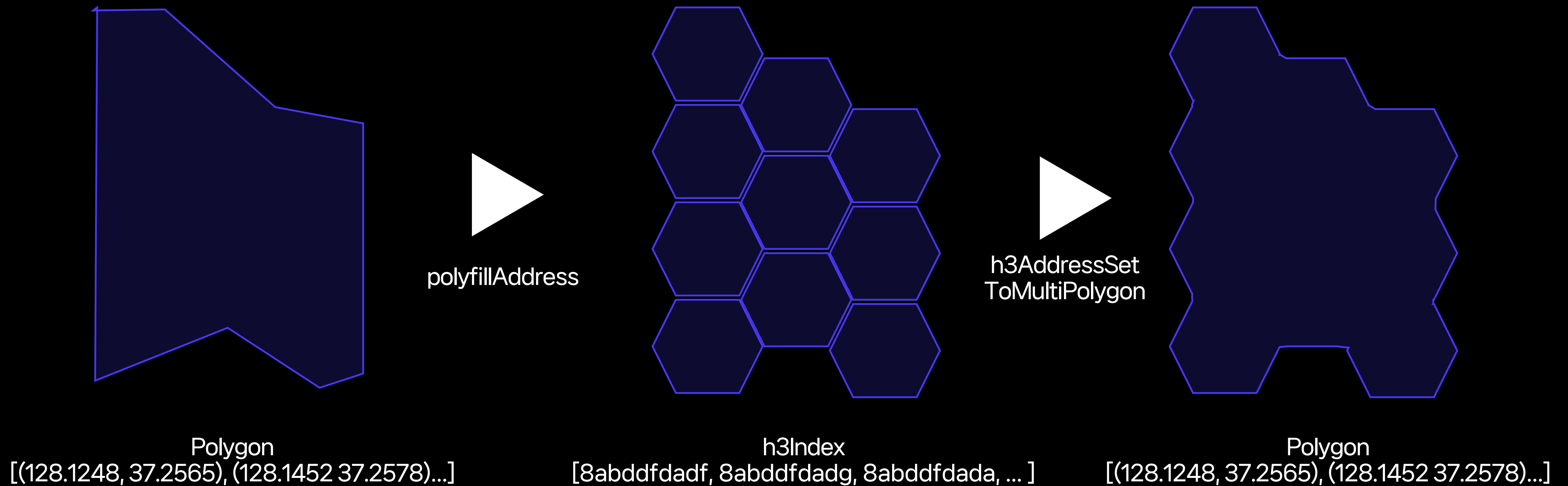


## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## How to

- H3 native API: `polyfillAddress` → `h3AddressSetToMultiPolygon`



# 2.2 고해상도 영역을 표현하면서

## Index scalability 유지하기 [Polygonize]

### Polygon 상호 변환에 대한 어려움

- 기존 Uber API 사용시 일반적인 사용이 어려운 수준으로 처리 속도가 느림

	Hexagon Counts	지연시간	내부 배열 생성
H3 Native API	12,000,000 140 MB	2시간 이상	98,000,000 Heap Memory 오류 발생



우편번호 28065  
넓이: 5,175,970 m<sup>2</sup>



## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## Polygon 상호 변환

폴리곤 분할

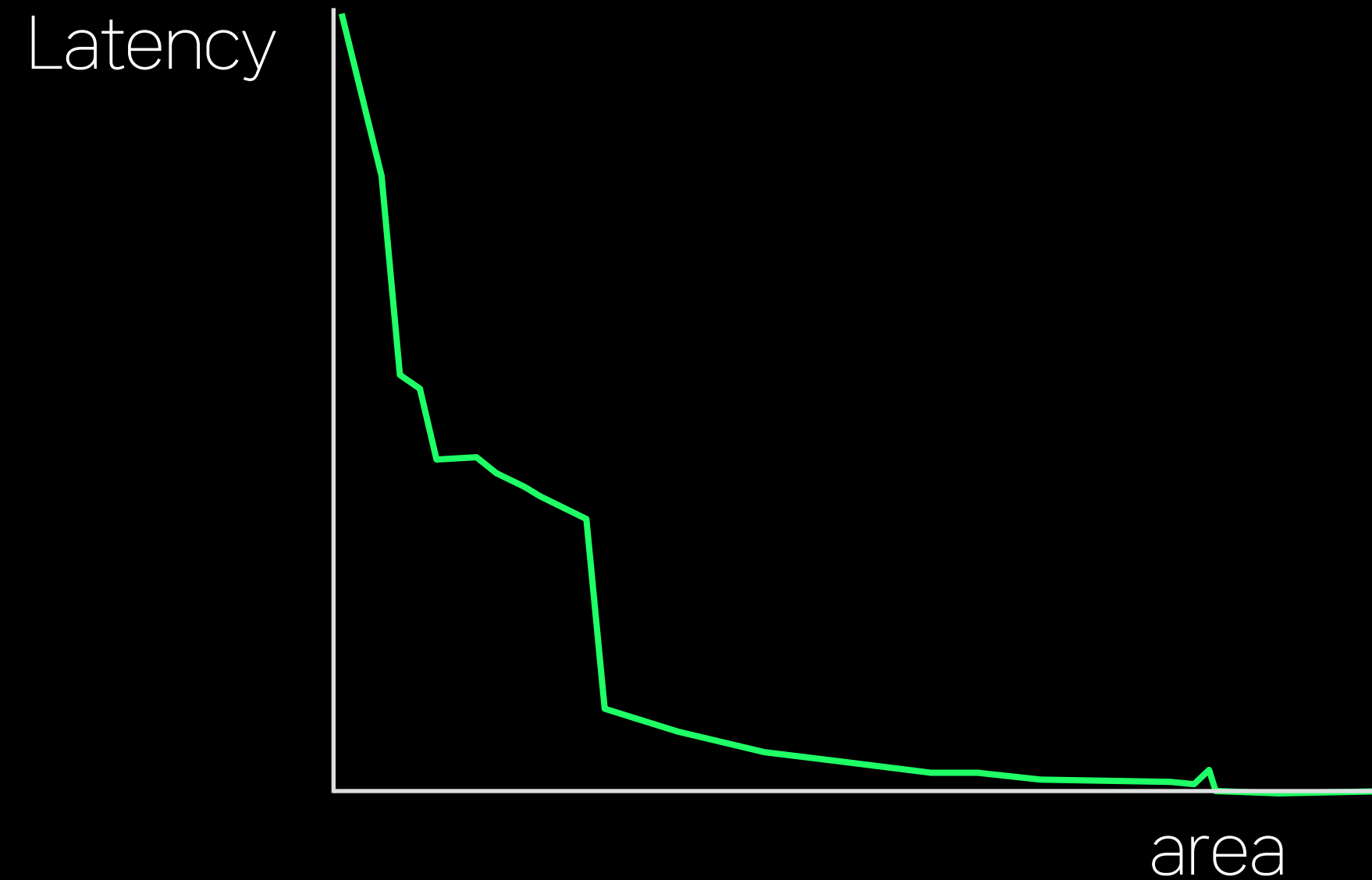
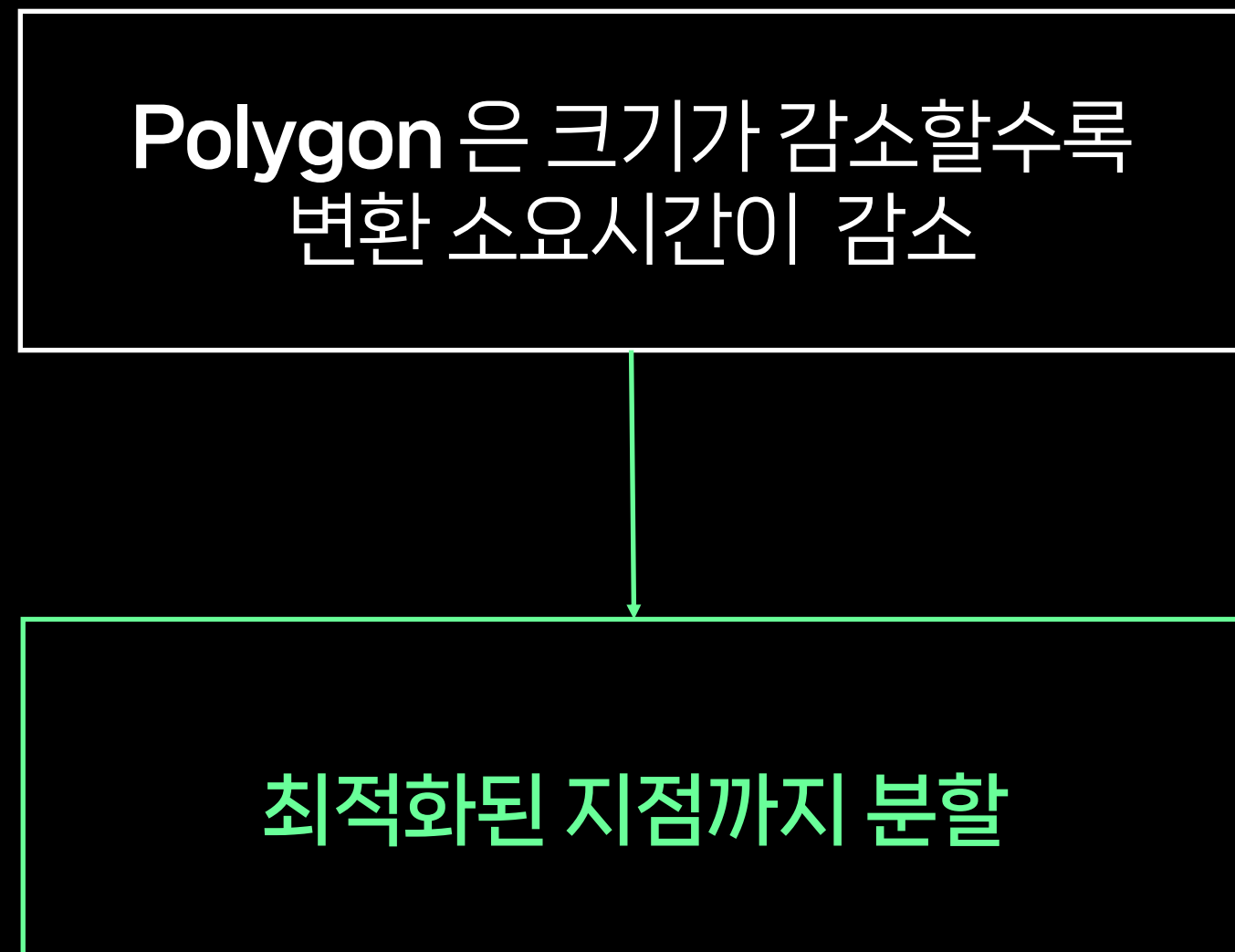
폴리곤 분해

헥사곤 폴리곤  
변환 최적화

## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

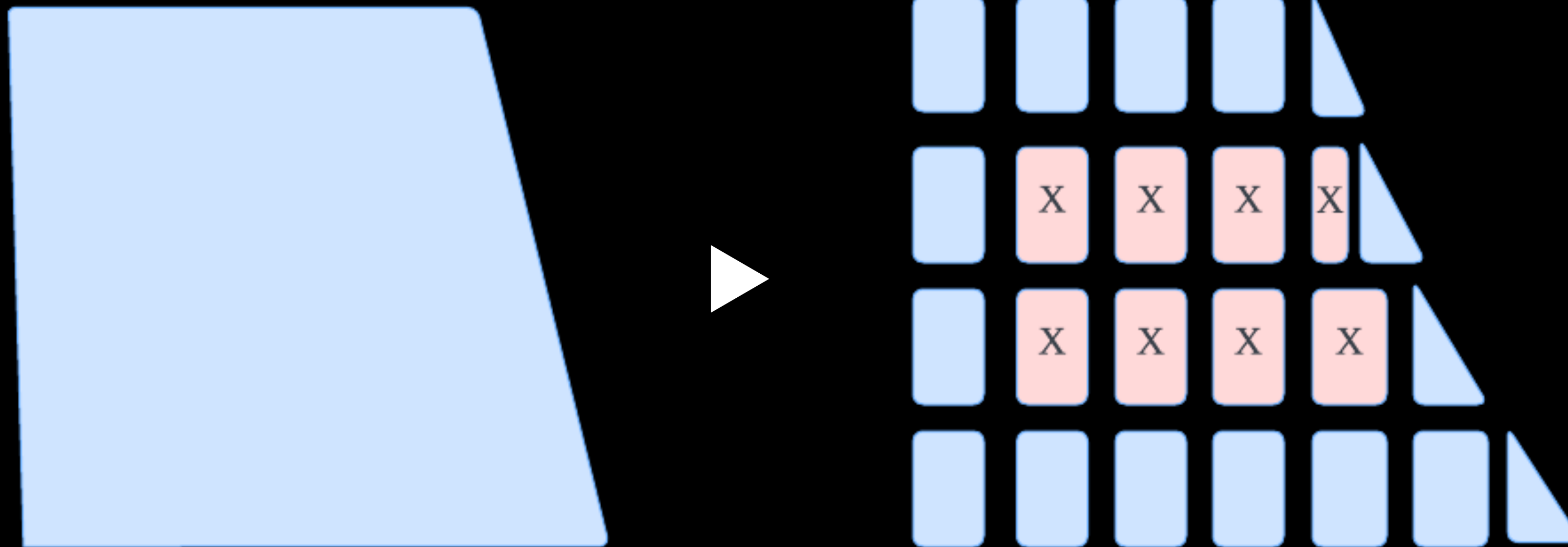
## Polygon 분할



## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## Polygon 분할



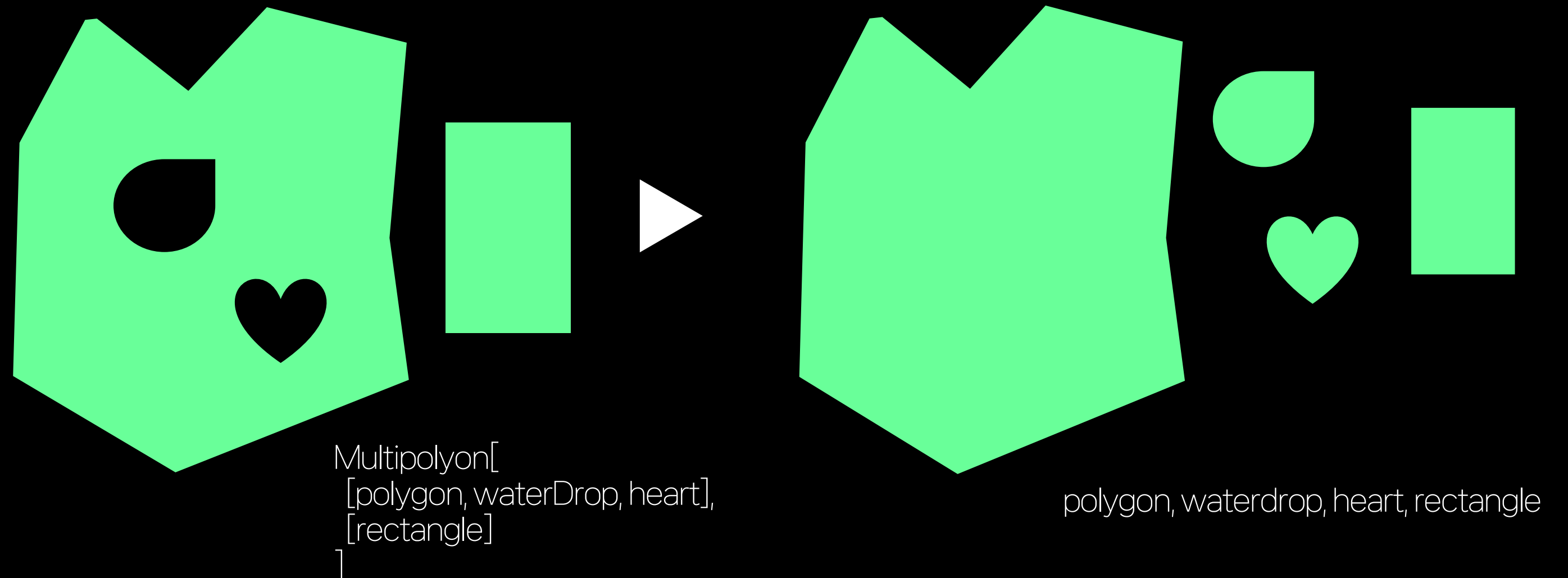
## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## Polygon 분해

헥사곤 폴리곤으로 변환 시  
하나의 폴리곤에 대해 최적화

MultiPolygon의 Polygon, Hole을  
개별 폴리곤으로 분해/변환/병합



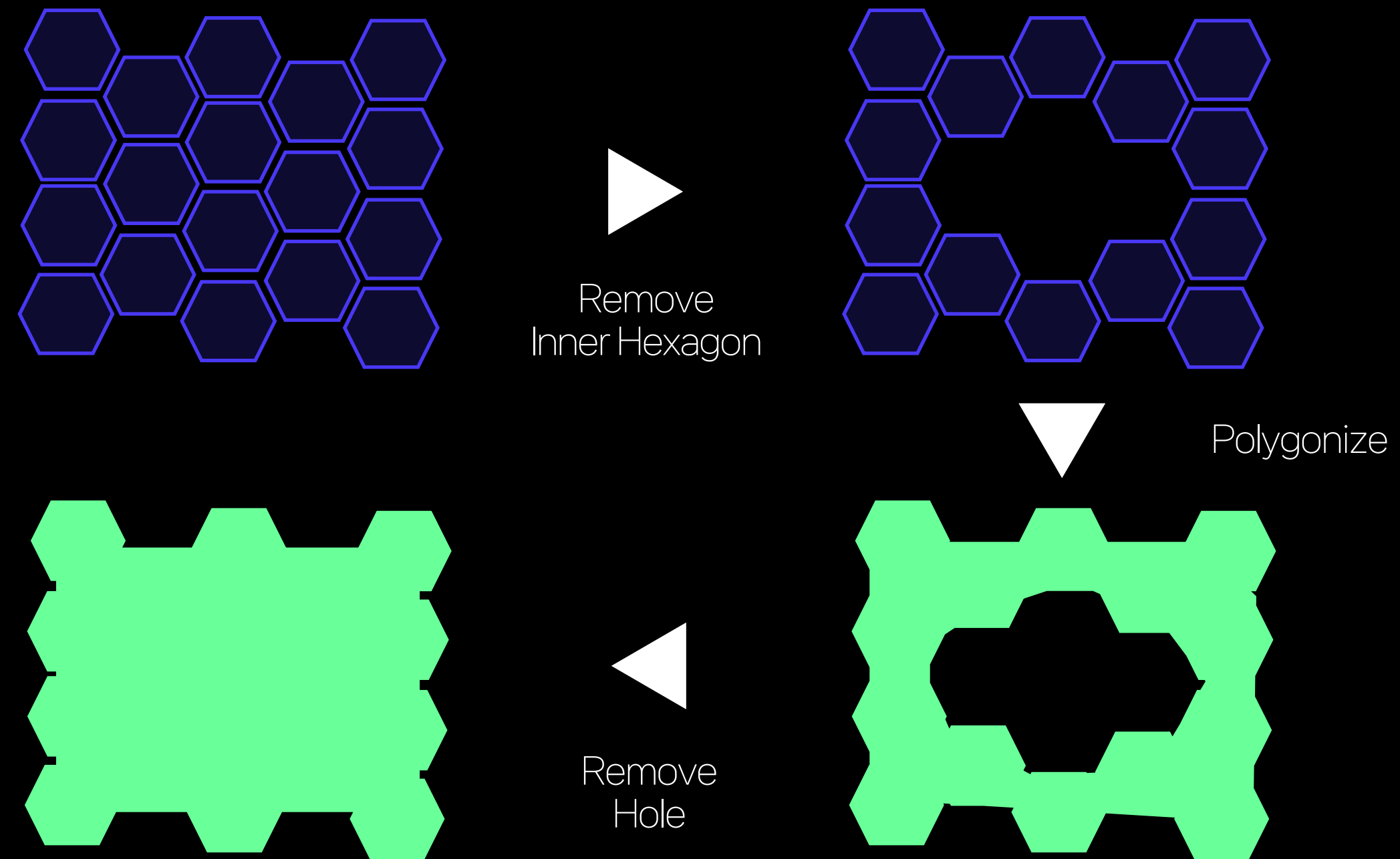
## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## 격자 모양 Polygon 변환 최적화

동일면적이라도  
격자수에 따라 속도/메모리 증가

격자를 최소화하여 변환



# 2.2 고해상도 영역을 표현하면서

## Index scalability 유지하기 [Polygonize]

### Polygon 상호 변환에 대한 어려움

- 개선 후

	Hexagon Counts	지연시간	내부 배열 생성
H3 Native API	12,000,000 140 MB	2시간 이상	98,000,000 Heap Memory 오류 발생
개선 Version	131,365 1.5 mb	20 초	메모리 이슈가 생기지 않도록 분할 처리



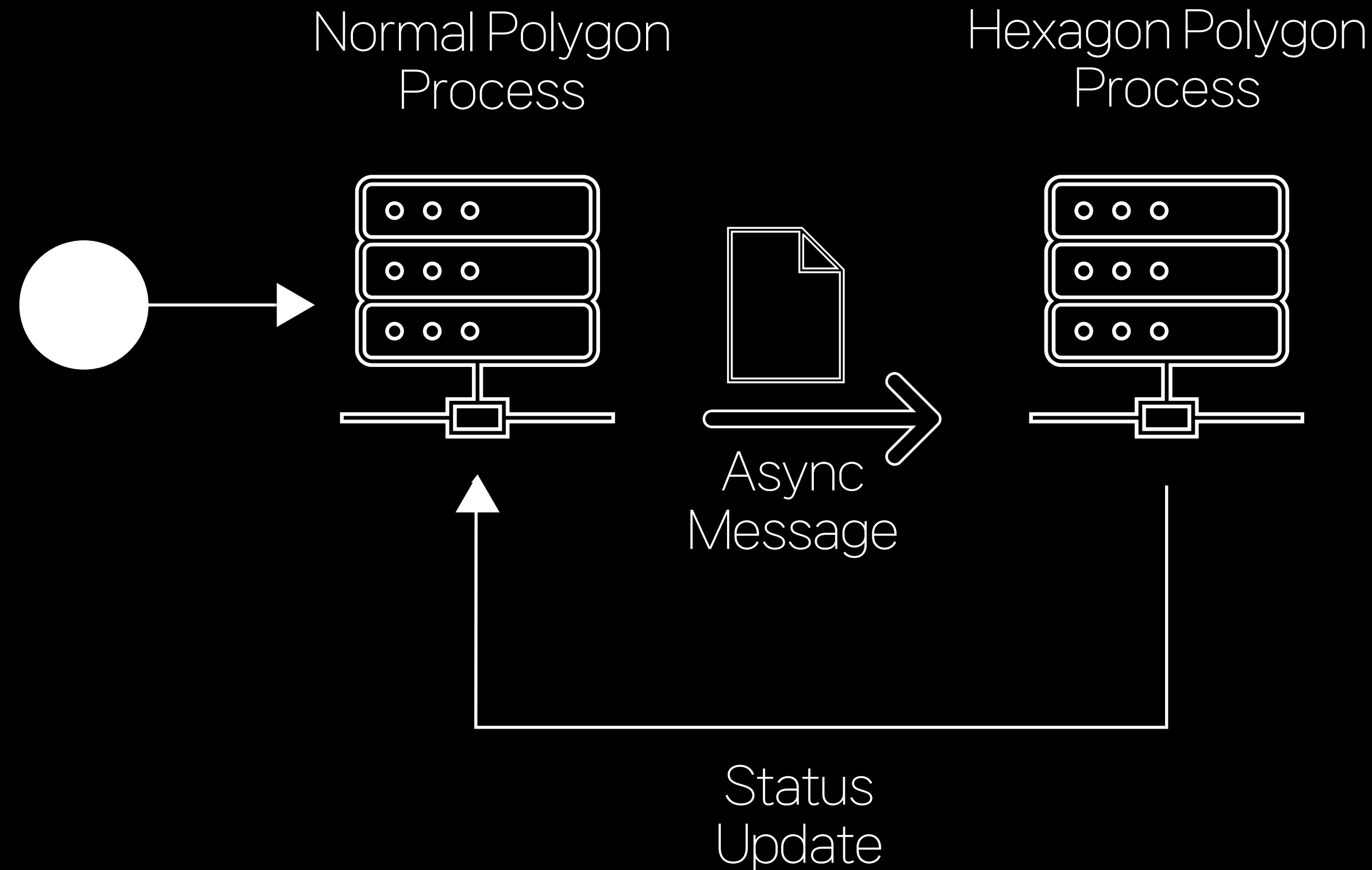
우편번호 28065  
넓이: 5,175,970 m<sup>2</sup>

## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## How to

- 비동기 변환



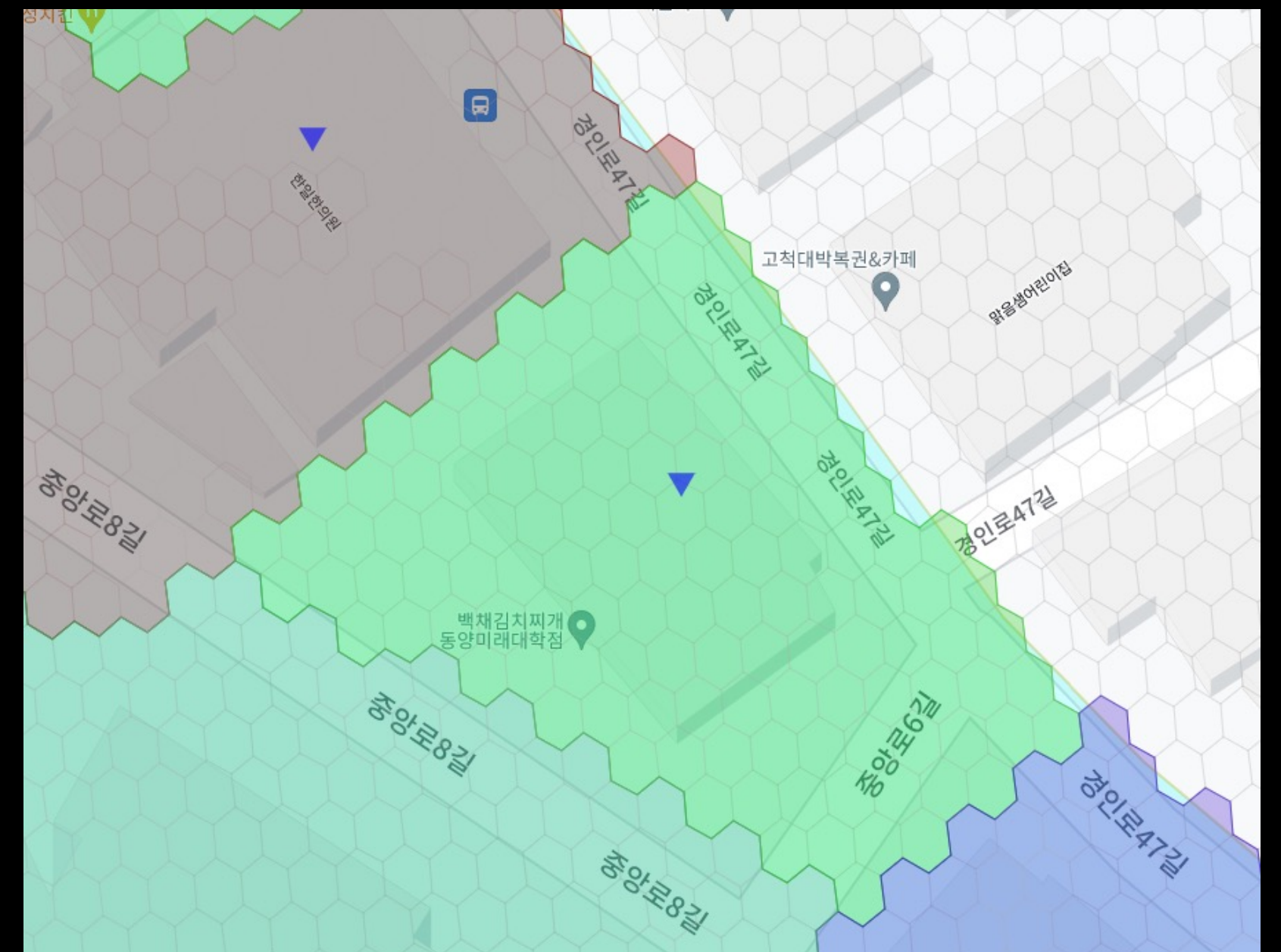
## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## 기존 geometry 데이터와의 호환성을 유지하는 법

일반 Polygon VS 격자 모양의 polygon

- Edge Point  
ex) 우편번호 28065
  - 일반 Polygon: 1,057개
  - 격자 모양의 polygon : 55,687



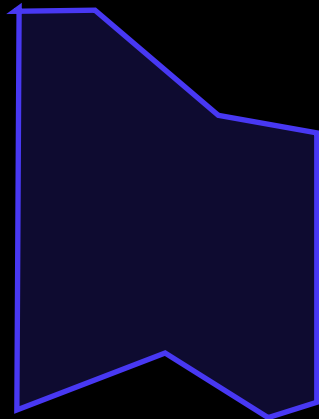
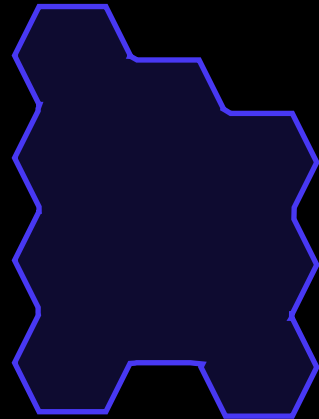


## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## 기존 geometry 데이터와의 호환성을 유지하는 법

Memory: 일반 Polygon VS 격자 모양의 polygon

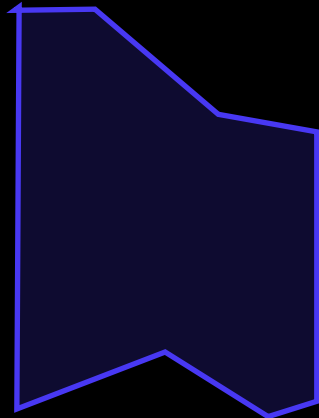
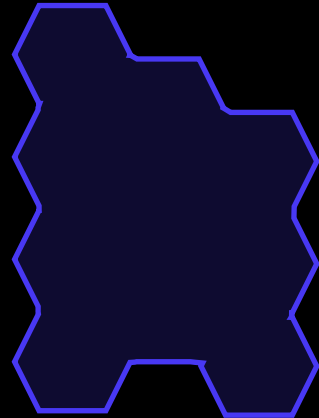
		Row Count	Memory
일반 Polygon		500,000	2.4GB
Hexagon-shaped Polygon		500,000	20GB

## 2.2 고해상도 영역을 표현하면서

# Index scalability 유지하기 [Polygonize]

## 기존 geometry 데이터와의 호환성을 유지하는 법

Latency: 일반 Polygon VS 격자 모양의 polygon

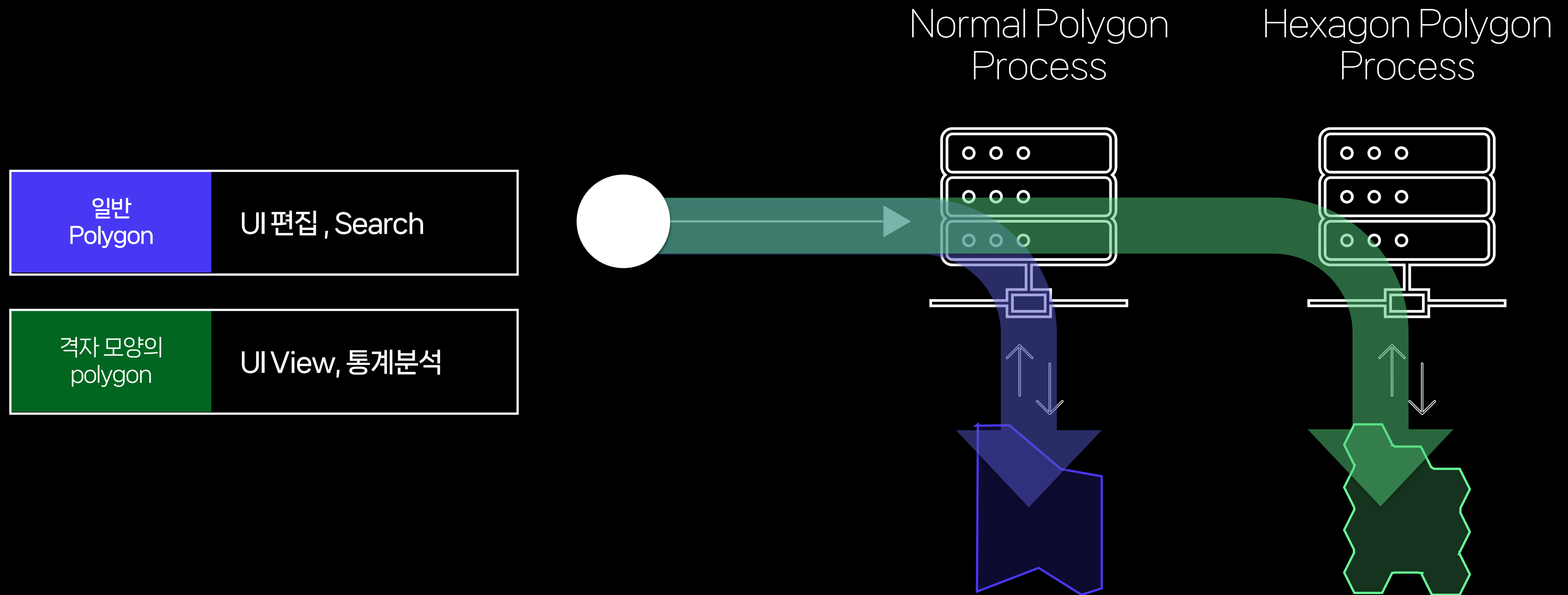
		Row Count	DWithin Avg Latency
일반 Polygon		100,000	30ms
Hexagon-shaped Polygon		50,000	150ms

## 2.2 고해상도 영역을 표현하면서

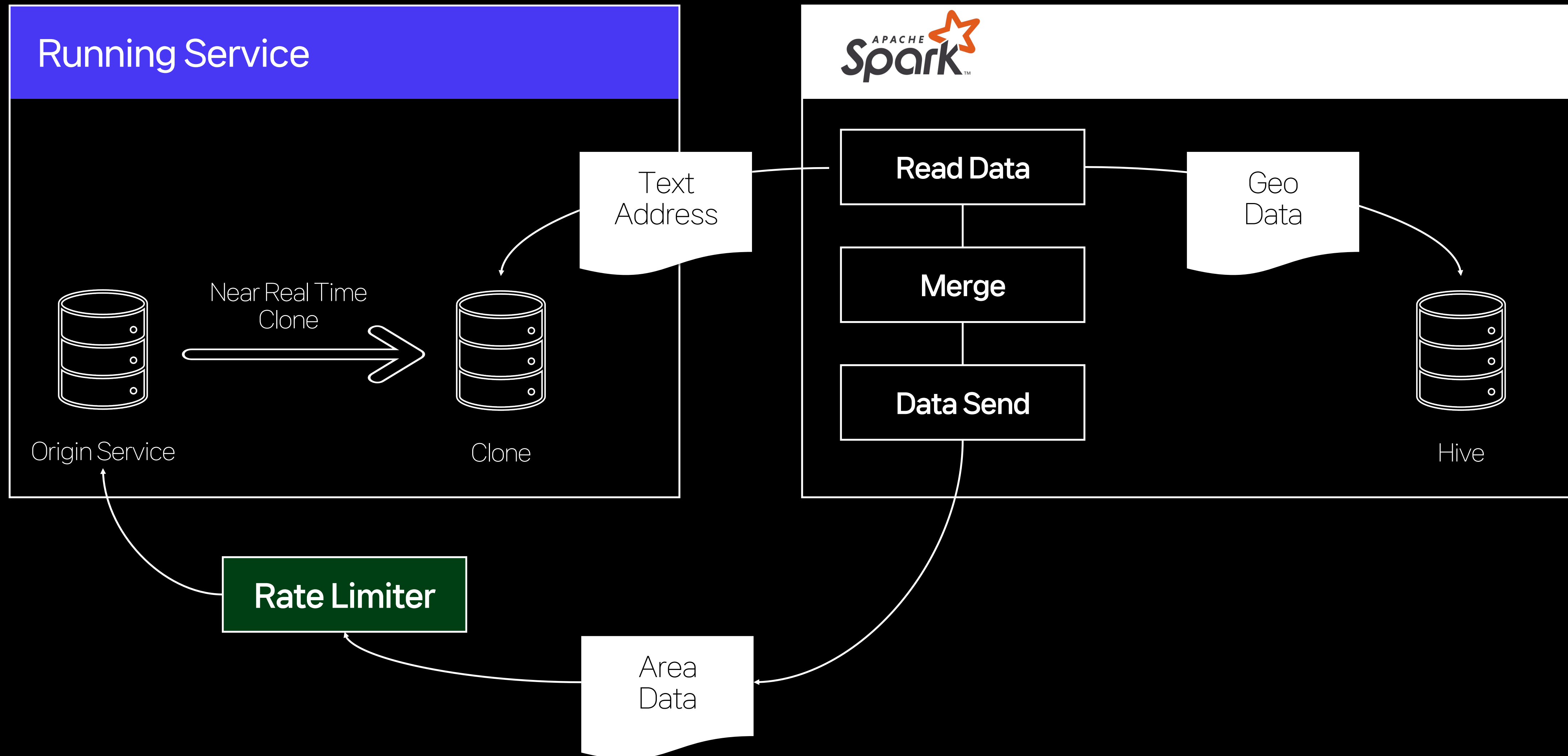
# Index scalability 유지하기 [Polygonize]

기존 geometry 데이터와의 호환성을 유지하는 법

- How to Resolve
  - 테이블 분리



# 2.3 Text 기반 주소체계 → 공간 영역 전환



# 2.3 Text 기반 주소체계 → 공간 영역 전환

Text Address	
Delivery Area	Address
001	부산 해운대구 해운대로 1 101동, 102동
002	부산 해운대구 해운대로 1 103동
003	부산 해운대구 동백로 1
...	



Spatial Database					
city	district	roadName	roadNumber	buildingDong	geom
부산	해운대구	해운대로	1	101	...
부산	해운대구	해운대로	1	102	...
부산	해운대구	해운대로	1	103	...
부산	해운대구	동백로	1-1	104	...
...					

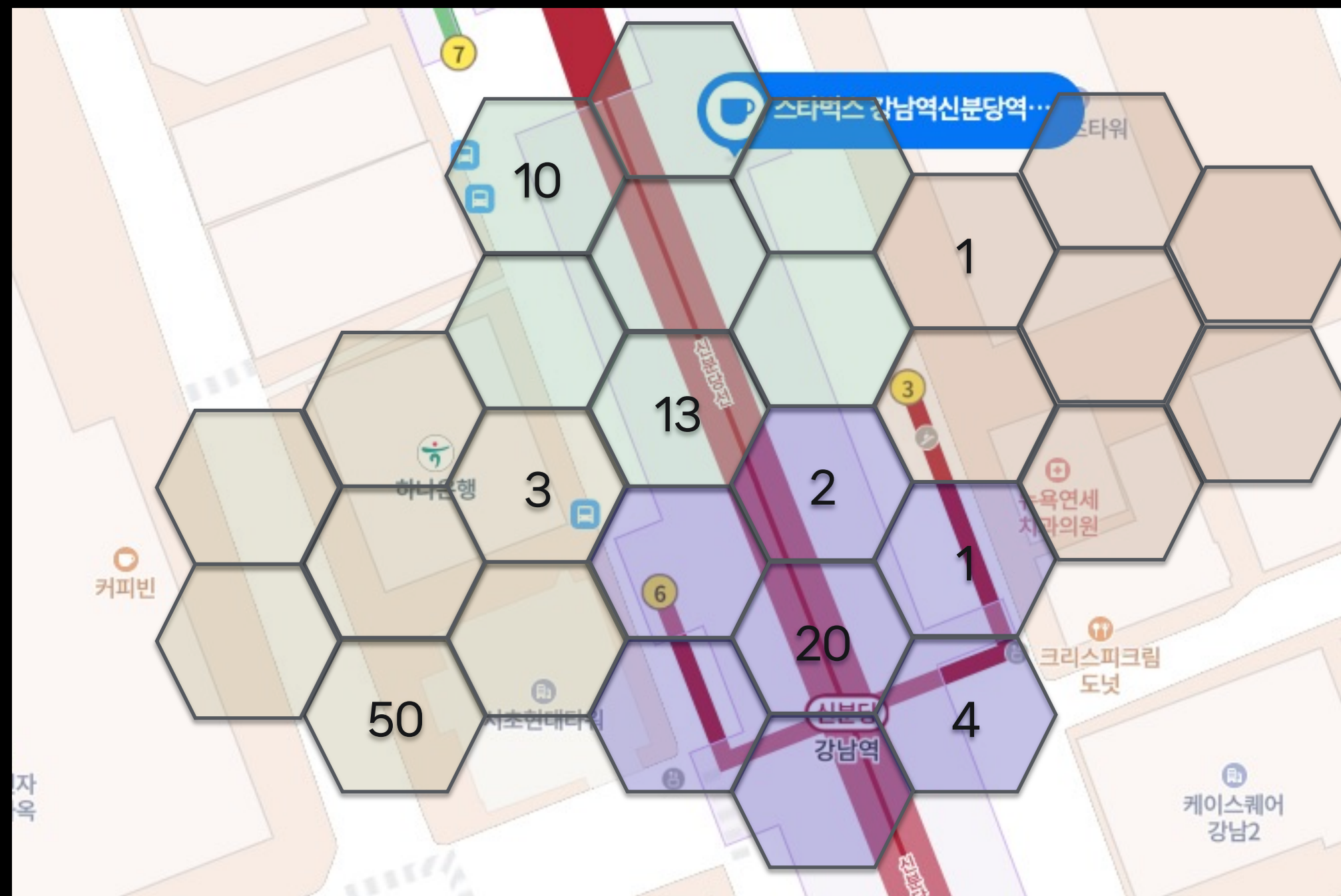


# 3. What's Next? + Q&A

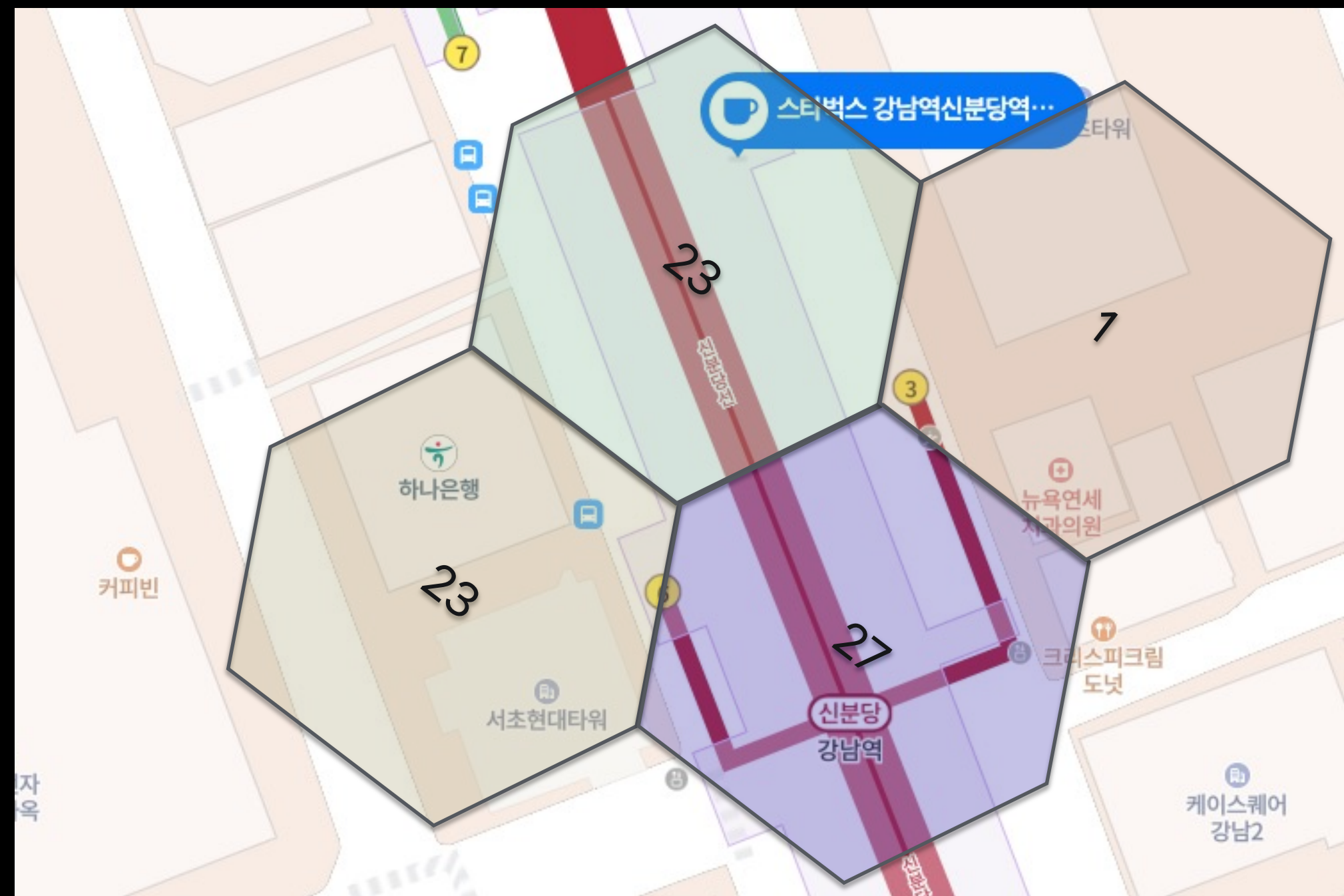
# 3.1 What's Next

## 통계 데이터

- 경/위도 좌표를 변환하여 지도 위에 Hexagon grid 단위의 통계 데이터를 보여준다.



하위 레벨

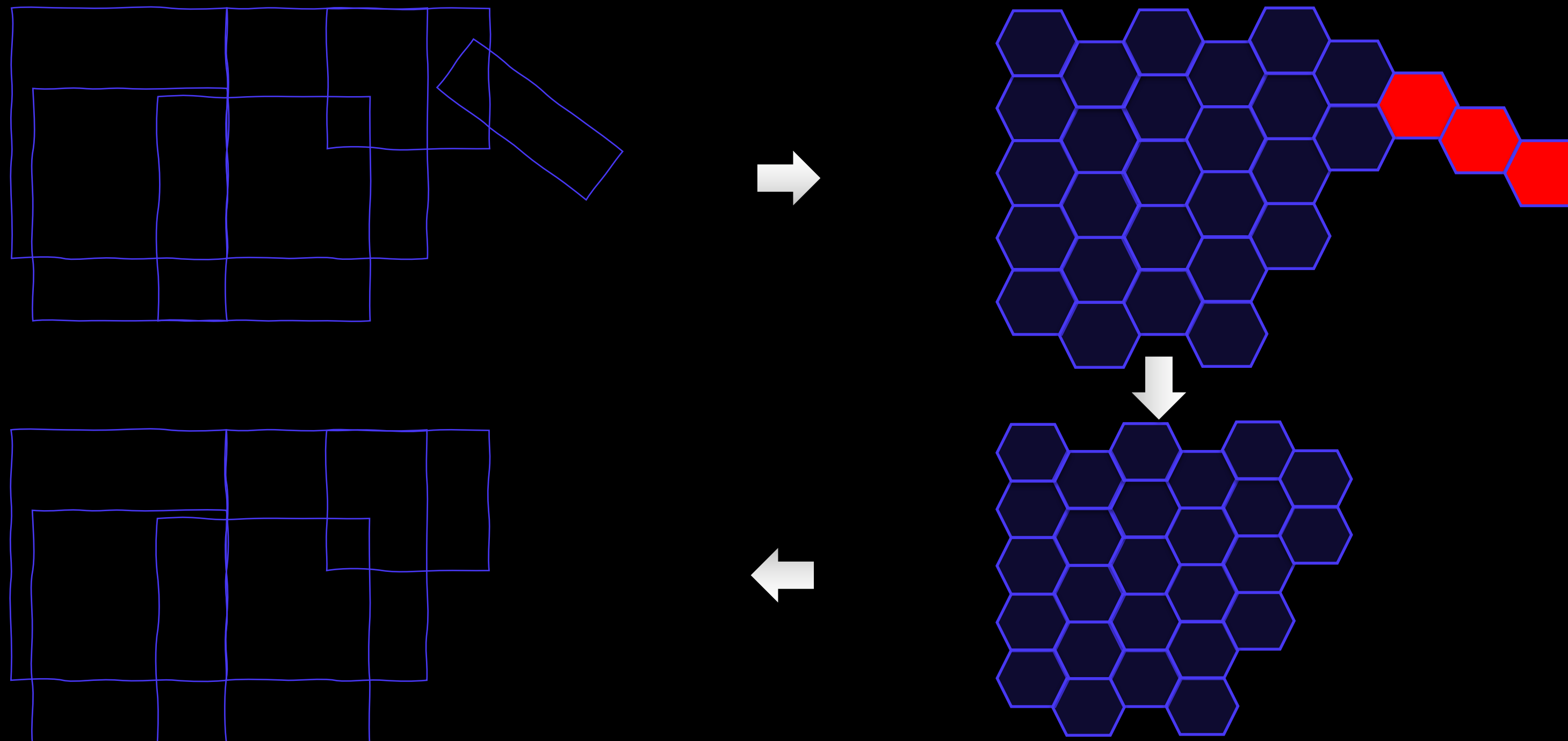


상위 레벨

# 3.1 What's Next

## 폴리곤 정제

- 일반 폴리곤으로 정제가 까다로운 폴리곤을 정제한다.
- 헥사곤의 접한면이 2개 이하일때 해당 폴리곤은 파편화로 인식하고 제거할수있다.





**Q & A**

**Thank You**