



ACM SIGCOMM 2018 Workshop on Mobile Edge Communications  
(MECOMM'2018), Budapest, Hungary

# Anveshak: Placing Edge Servers In The Wild

Nitinder Mohan   Aleksandr Zavodovski   Pengyuan Zhou   Jussi Kangasharju

University of Helsinki

{firstname.lastname@helsinki.fi}

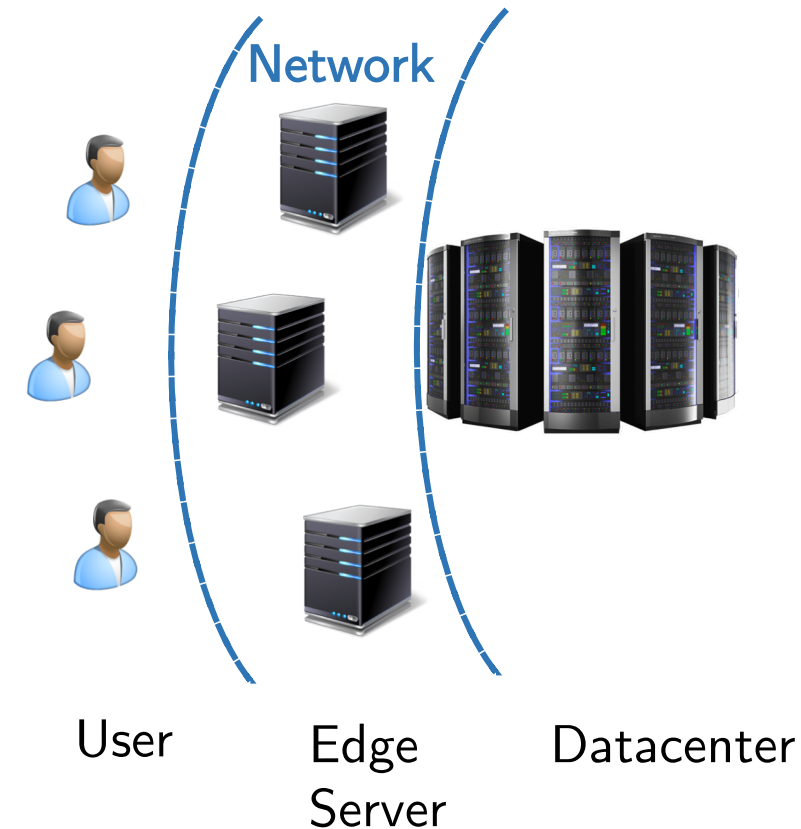
**Anv@shak**

# Edge Cloud Overview

**Edge Cloud:** Small-scale server(s) deployed at network edge to compute user data

**Motivation:**

- ✓ Decreased latency and network traffic
- ✓ Computing data of local relevance

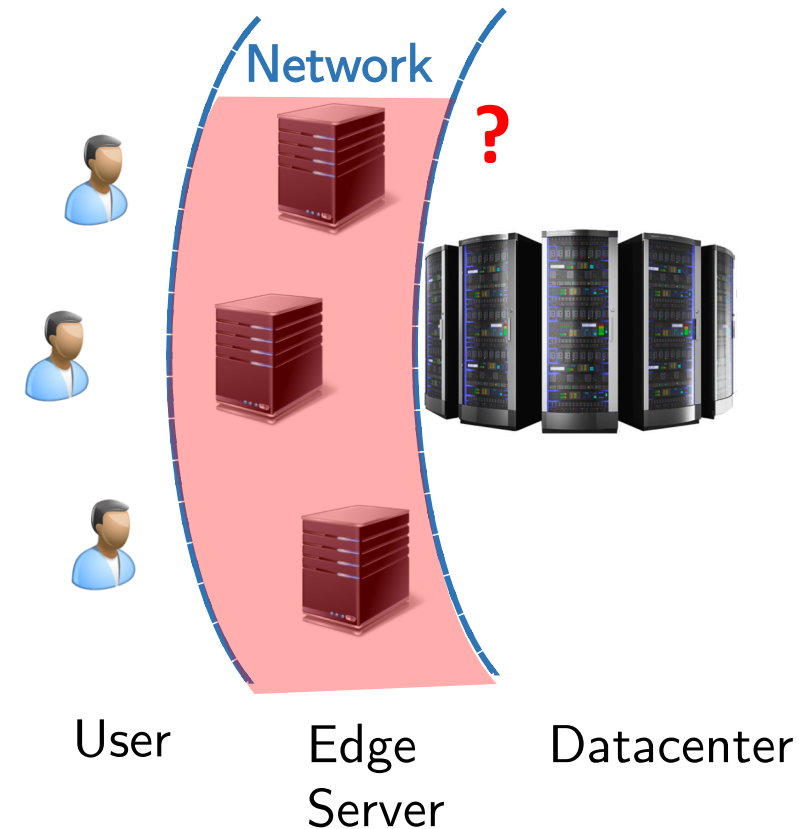


# Edge Cloud Overview

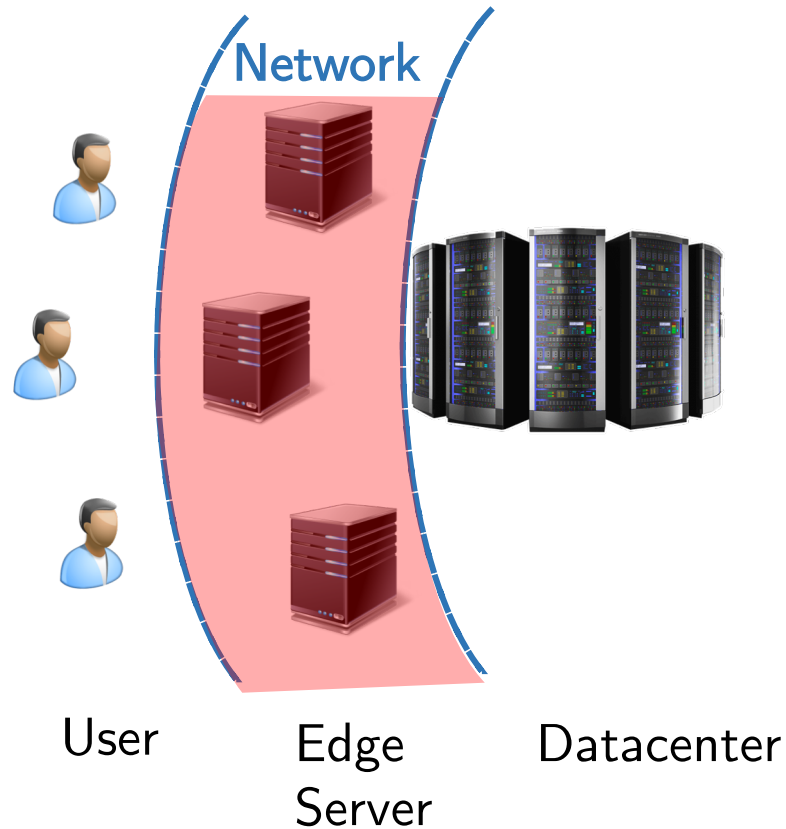
**Edge Cloud:** Small-scale server(s) deployed at network edge to compute user data

**Motivation:**

- ✓ Decreased latency and network traffic
- ✓ Computing data of local relevance



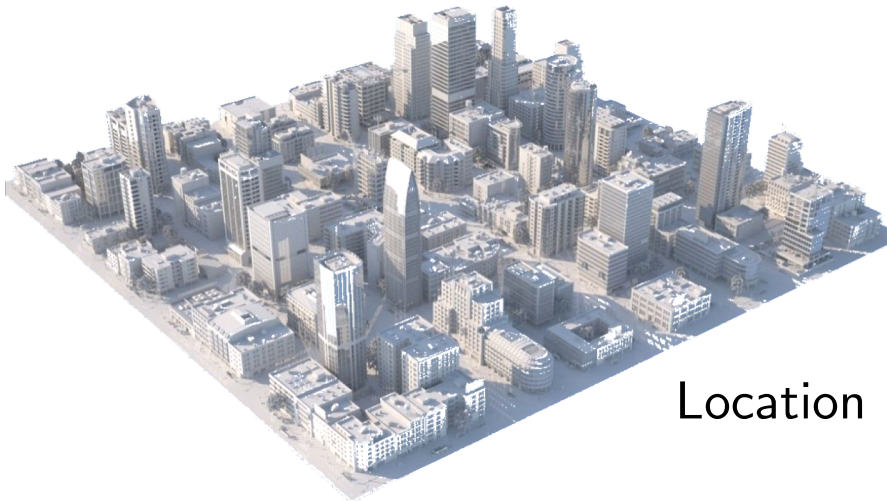
# Edge Cloud Deployment



1. What are these servers configuration?
2. Who owns and operates them?
3. Where are these servers going to be deployed in the physical world?

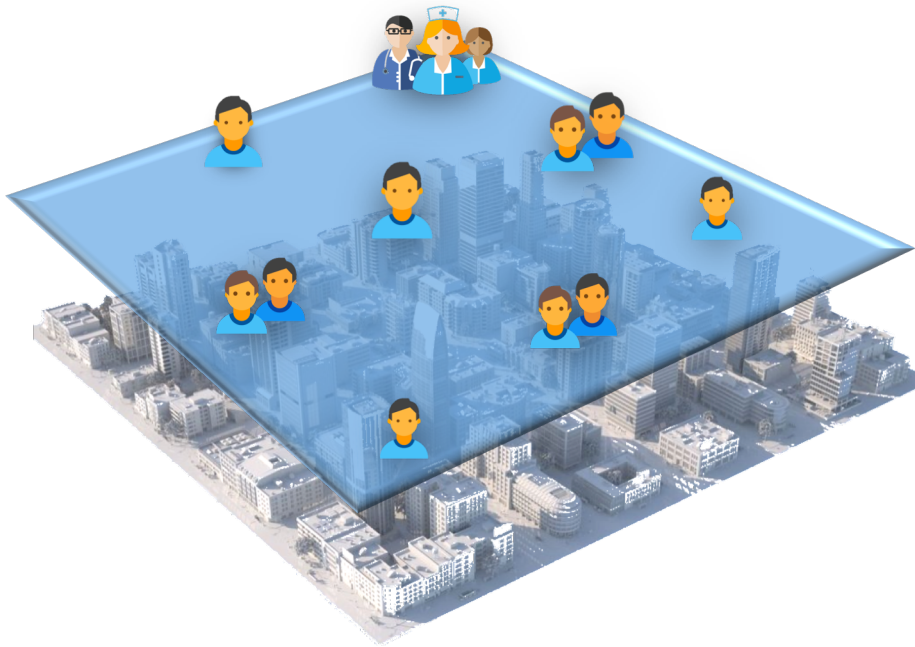
# Physical Edge Cloud Network

1. Multiple edge cloud deployments **can co-exist** in same physical space
2. Edge server availability will directly affect end utilization
  - Usage of cellular-based internet in areas where WiFi is available



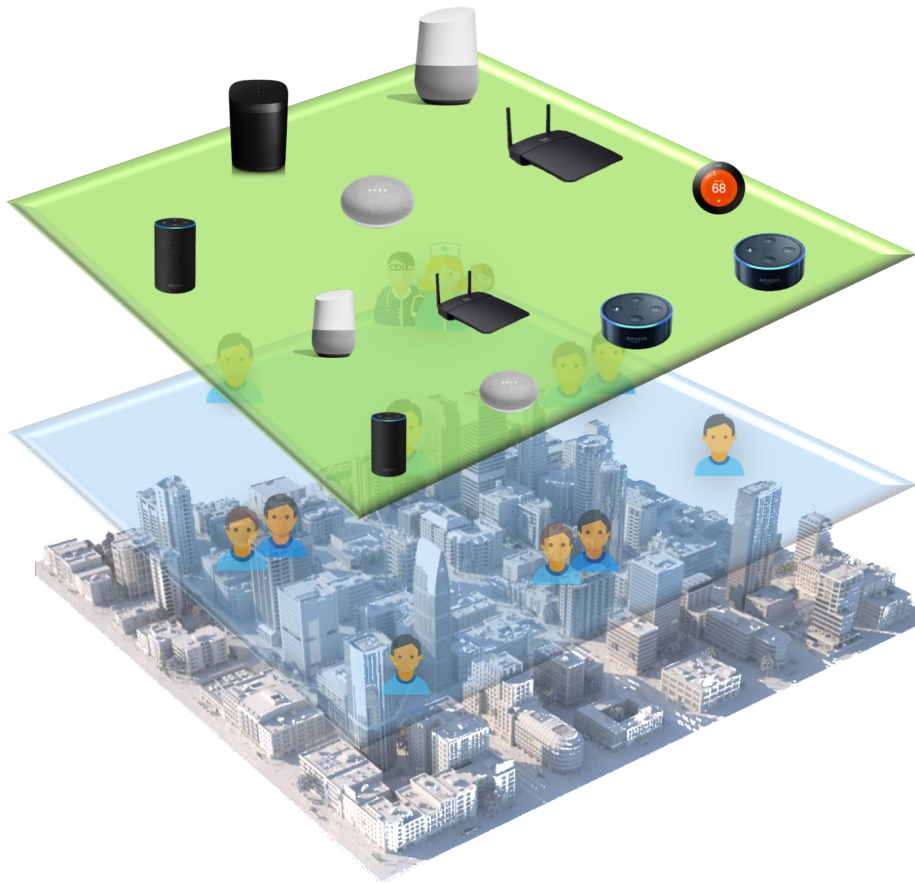
Location 'L'

# Users



1. Subscribers and requesters of edge resources
2. User density directly impacts local edge server utilization
3. Request distribution is highly dependent on **time** and **behavior**
  - More user requests during daytime than in night
  - More user requests in city than suburban areas

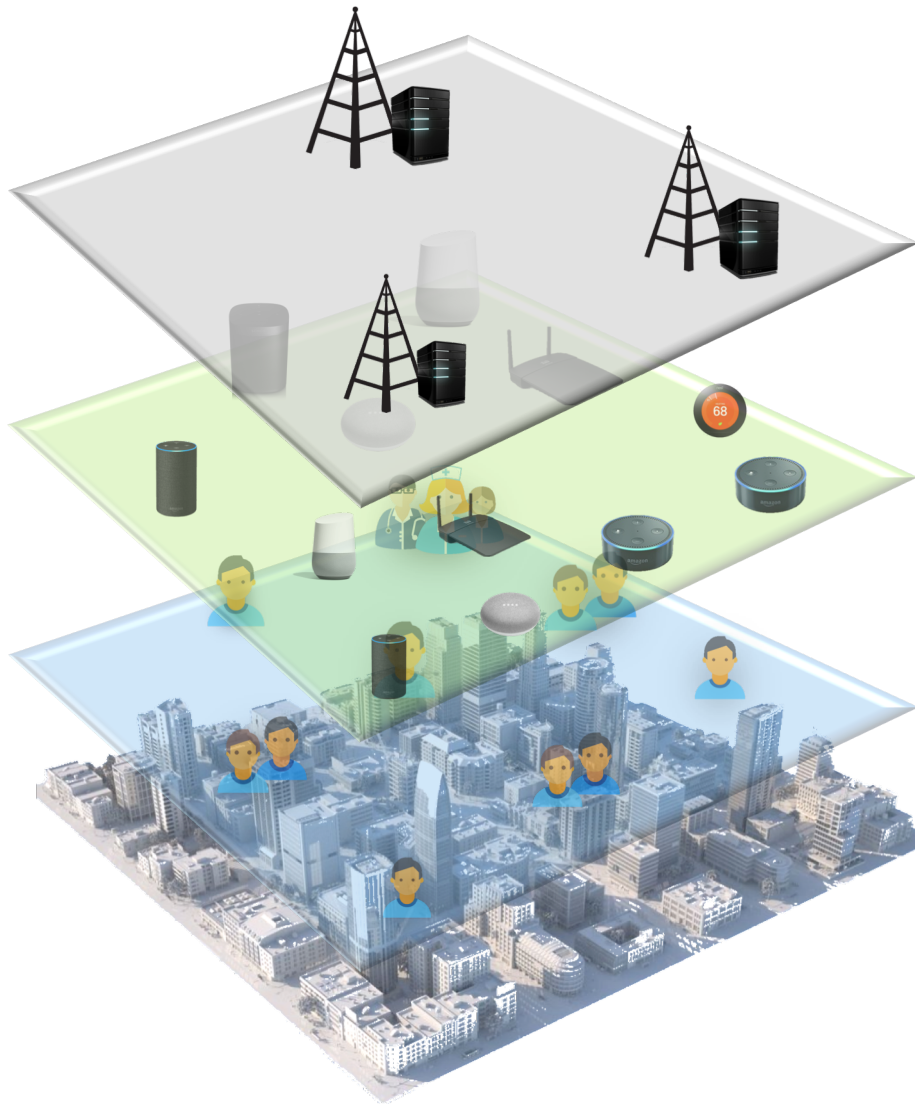
# User-Managed Edge



1. Composed of: self-managing, locally-relevant edge resources
  - e.g. smart speakers, home automation, intelligent WiFi hubs
2. Limited computational-power and network capability
3. High server density
  - Dependent on user population
  - One server caters to small set of users

**Management/Control → End-users**

# Service Provider Managed Edge



1. Composed of: high computation and network capable edge servers
2. Set up specifically by cloud provider in partnership with local ISP
3. Co-located/accessible with cellular base stations for *ease-of operation* and *maintenance*
4. Low server density
  - One server caters to large set of users

**Management/Control → Cloud provider/ISP**

# New Server Deployment\*

\*from Service Provider perspective



Which location to deploy new edge server?

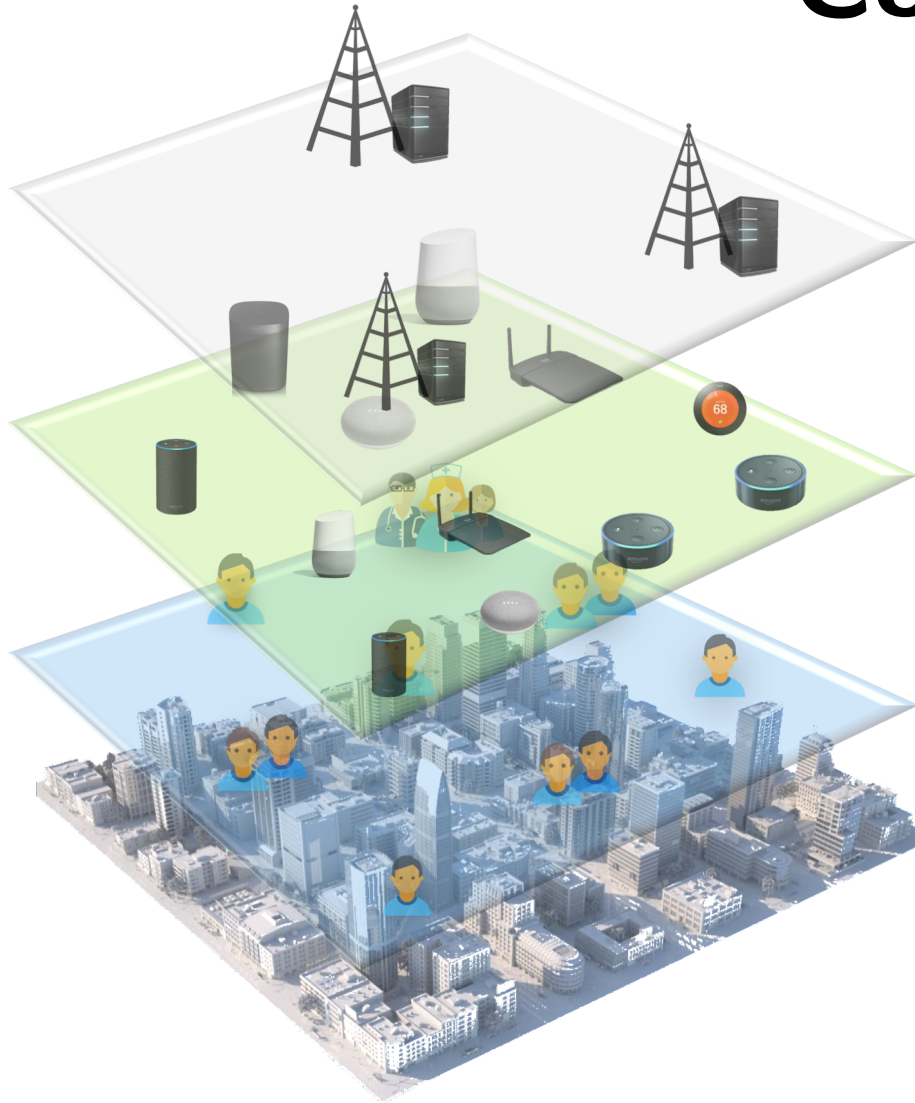
**Over-provisioning!**

- Connectivity can be overlapping
  - Future utilization can be minimal
- OR

**Under-provisioning!**

- Maintain a Quality-of-Service
- Must support peak request traffic

# Considerations



Deploying managed edge servers is expensive!

An efficient server deployment algorithm **must**:

# Considerations

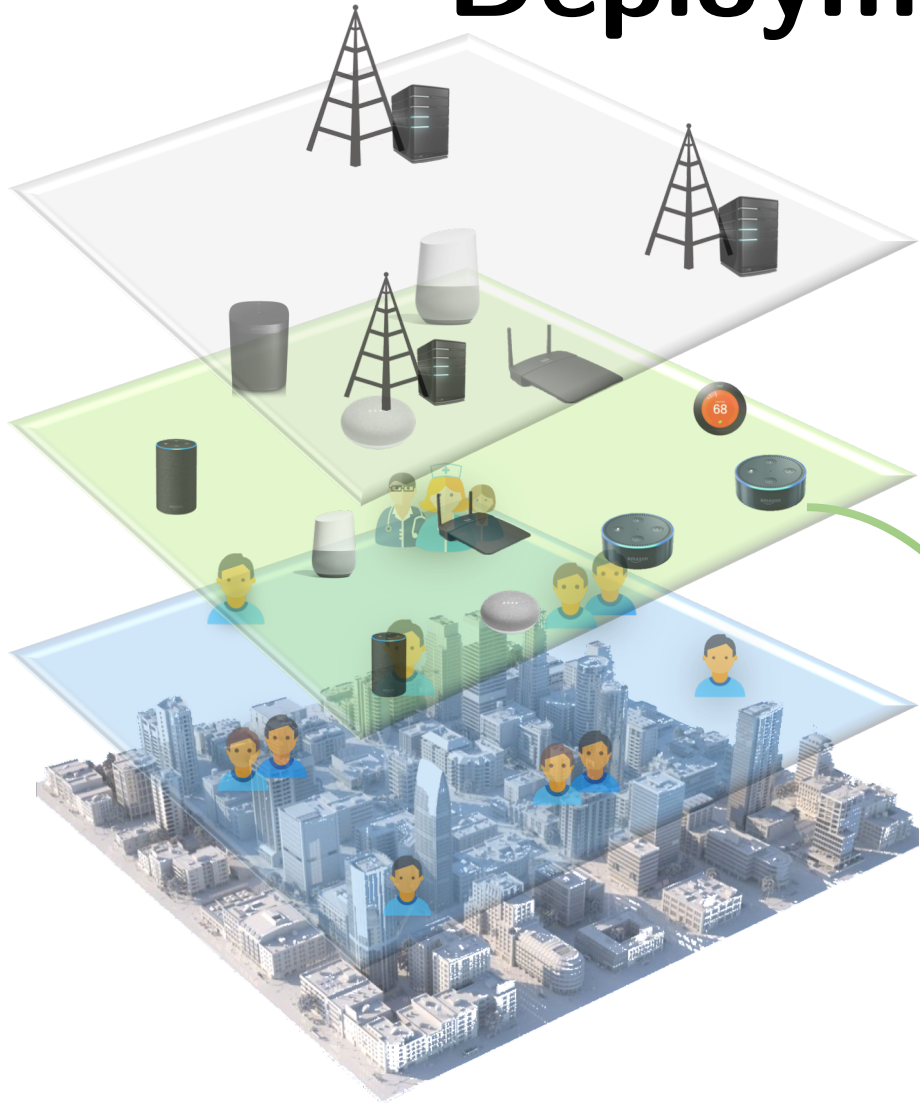


Deploying managed edge servers is expensive!

An efficient server deployment algorithm **must**:

1. Prioritize areas with high user requests

# Deployment Considerations



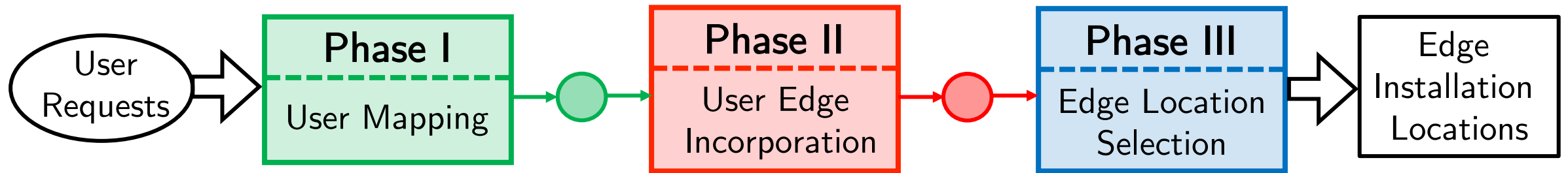
Deploying managed edge servers is expensive!

An efficient server deployment algorithm **must**:

1. **Prioritize** areas with high user requests
2. **Avoid** areas with high user-managed edge resources

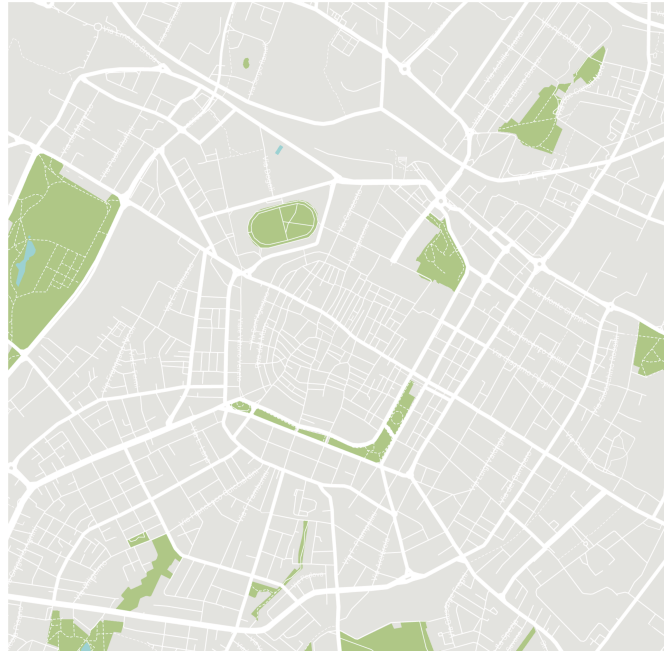
**Anv@shak**

# System Workflow



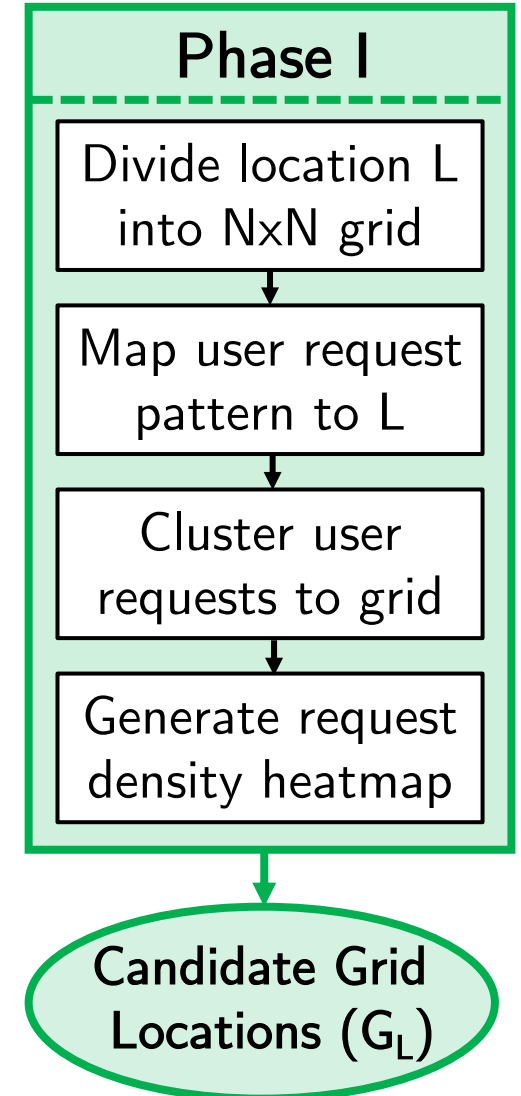
- Three-phase waterfall-based workflow
- Intermediate phase checkpoints for recoverability
- Swappable phase modules for incorporating improved algorithms and parameters
- Anveshak is *first-of-its-kind* framework which considers user-managed edge in location selection

# Phase I: User Mapping

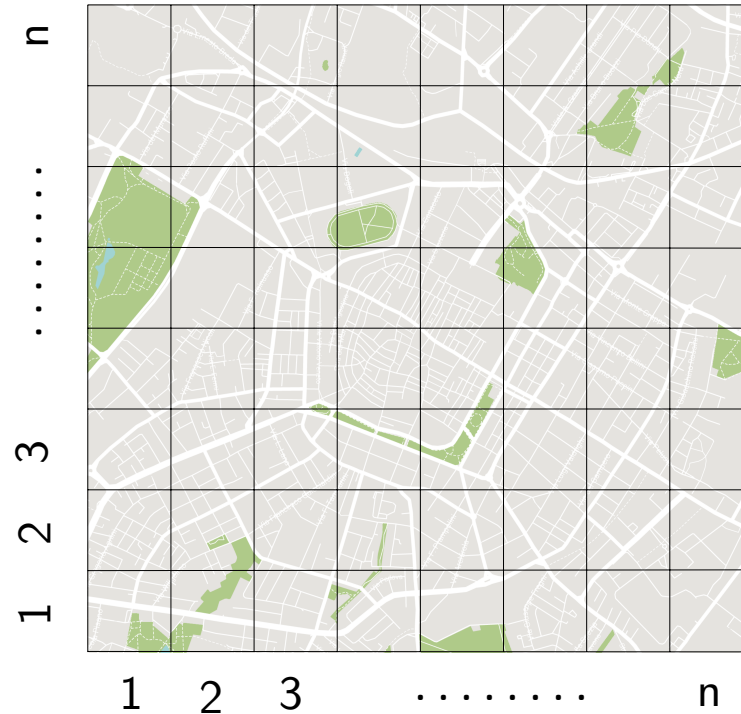


**AIM:** Identify and prioritize areas of high user communication requests

- High server utilization
- Low user-server connection latency

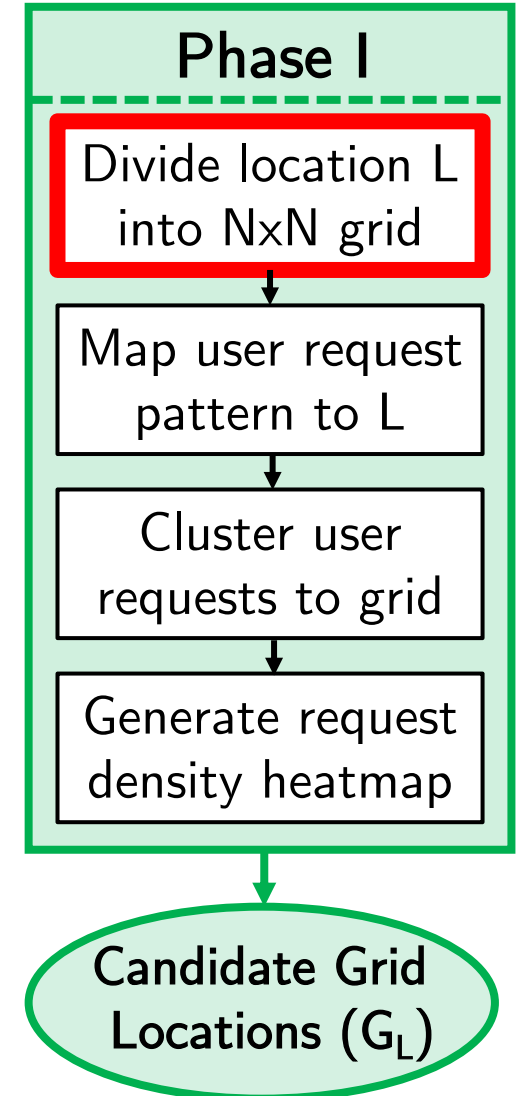


# Phase I: User Mapping



Divide map into equi-spaced  $N \times N$  grid

- Grid division allows for consistent clustering
- Grid size directly affects problem search space

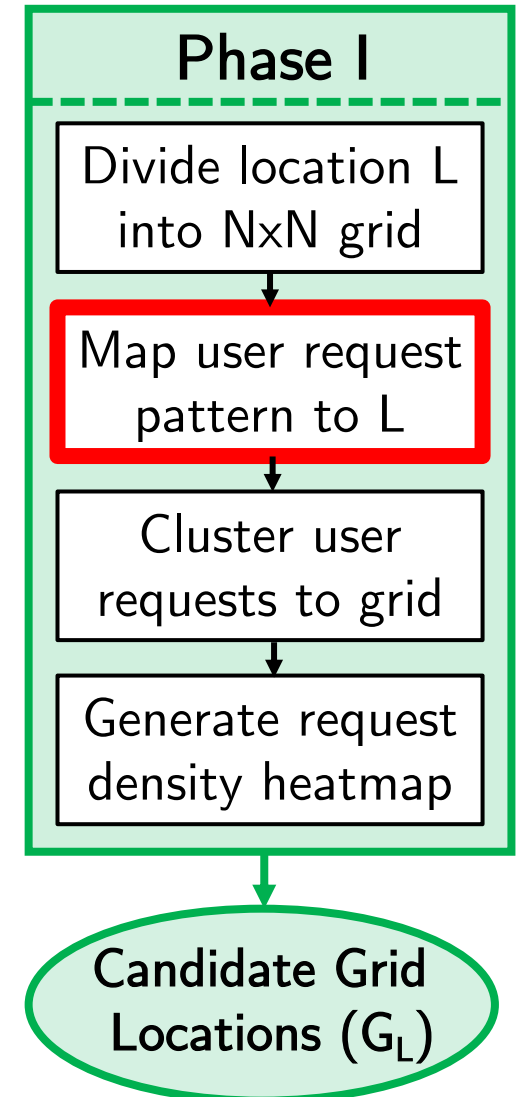


# Phase I: User Mapping

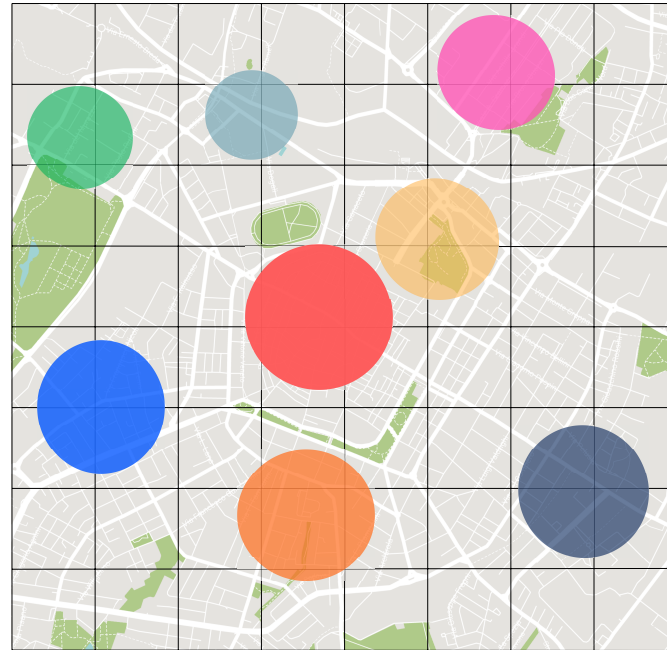


Past user communication requests such as *CDRs*, *internet initiation* are mapped on location

- Requests are averaged over time to remove temporal outliers

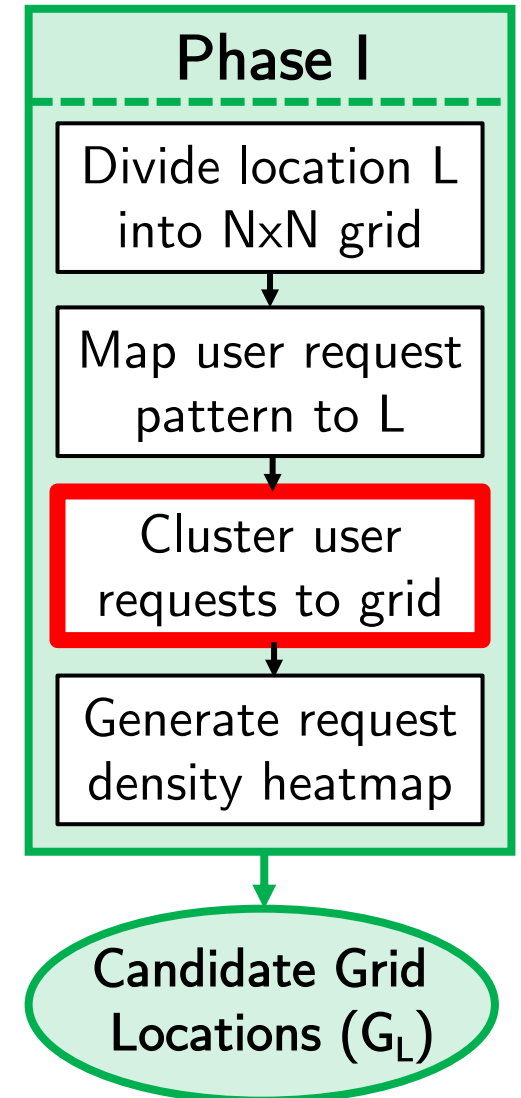


# Phase I: User Mapping

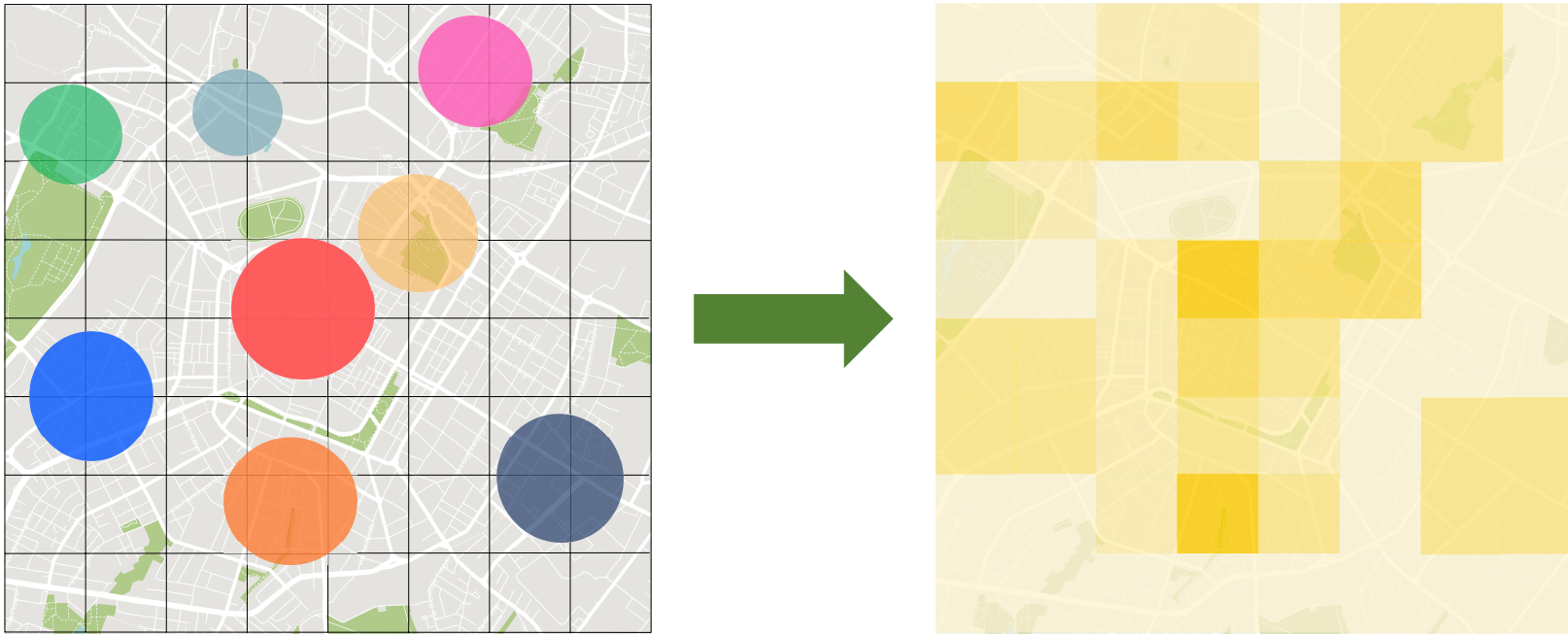


Cluster user requests based on inter-request distances and densities

- Choice of clustering algorithms and their parameters can be easily tweaked [***DBScan*** used as example]

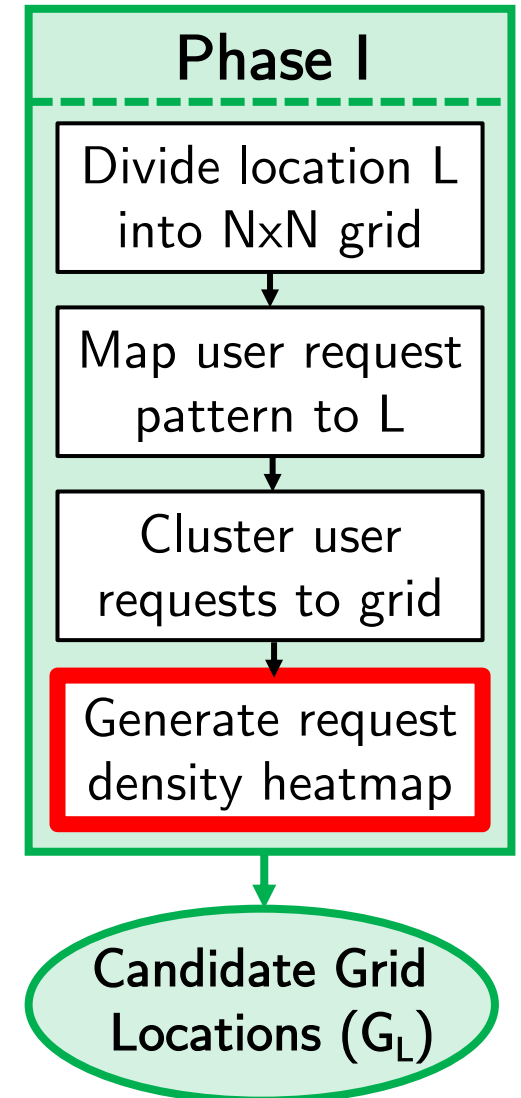


# Phase I: User Mapping

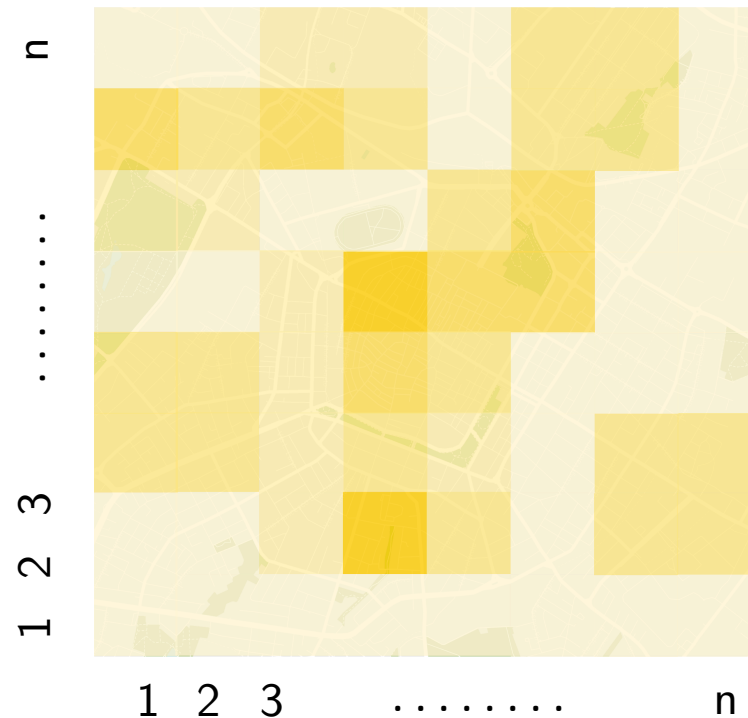


Generate heatmap for arbitrary cluster shapes

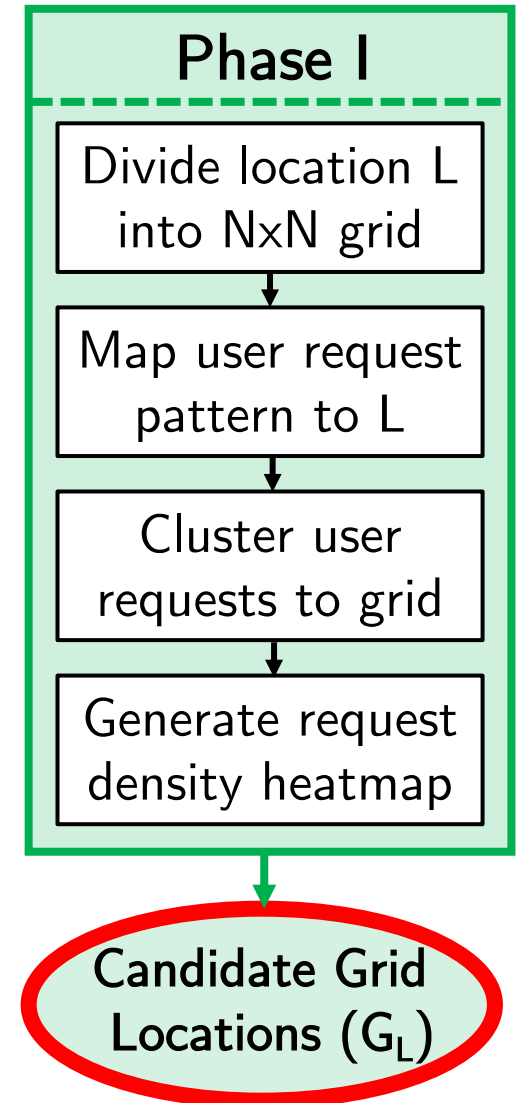
- Handles over-lapping shapes, small/dense clusters
- Handles any inefficiency of clustering algorithm



# Phase I: User Mapping



$$G_L = \{\text{Grid ID, Request Density}\}$$

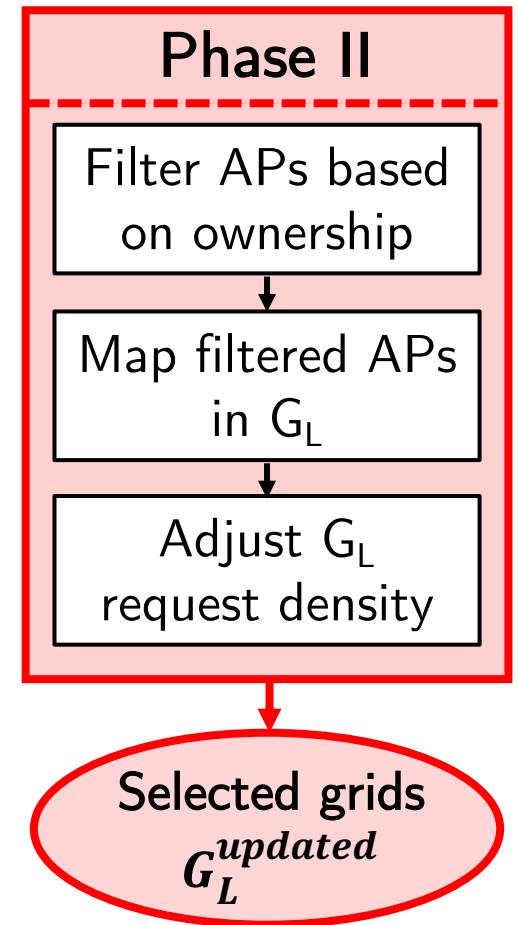


# Phase II: User Edge Incorporation

- Anveshak estimates future deployment of user-managed edge resources
- Availability of user-edge servers will **limit** utilization of deployed edge in same location
- Such devices are highly dependent of user population and interaction in an area

Estimated via current deployment of WiFi access points

INPUT:  $G_L = \{\text{Grid ID, Request Density}\}$

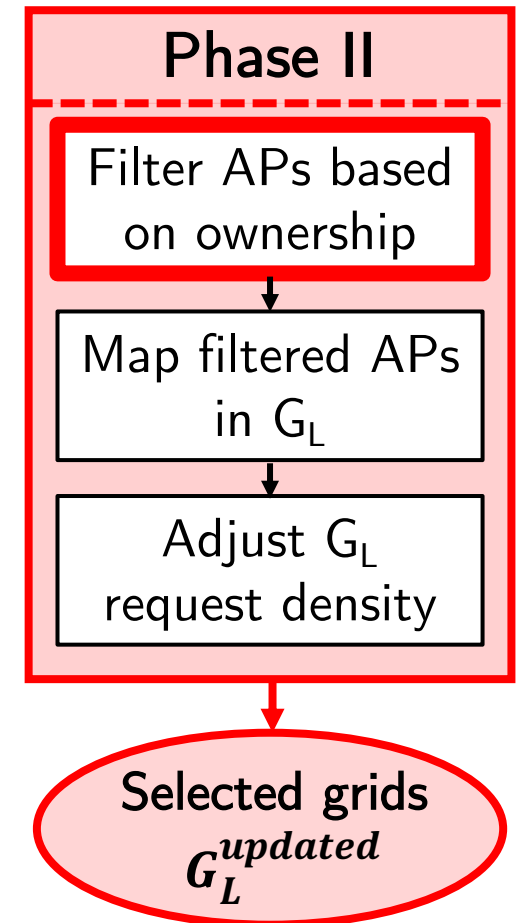


# Phase II: User Edge Incorporation

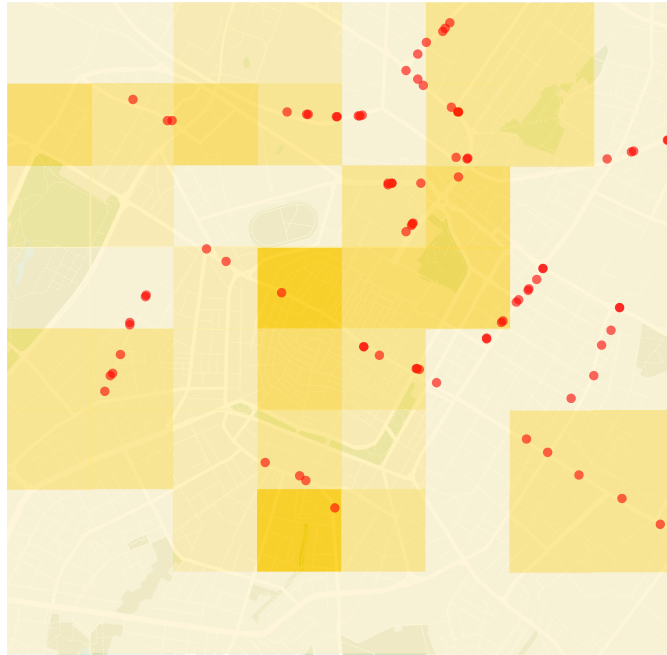
Map currently deployed WiFi access points in the same area

- Utilize open datasets for WiFi access points such as wigle.net
- Filter out *mobile and temporary access points*

**WIGLE.NET**<sup>TM</sup>

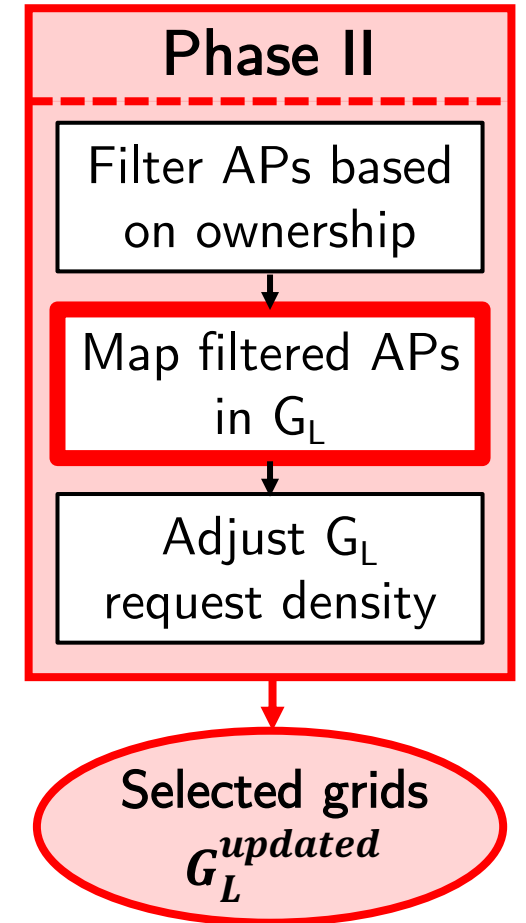


# Phase II: User Edge Incorporation

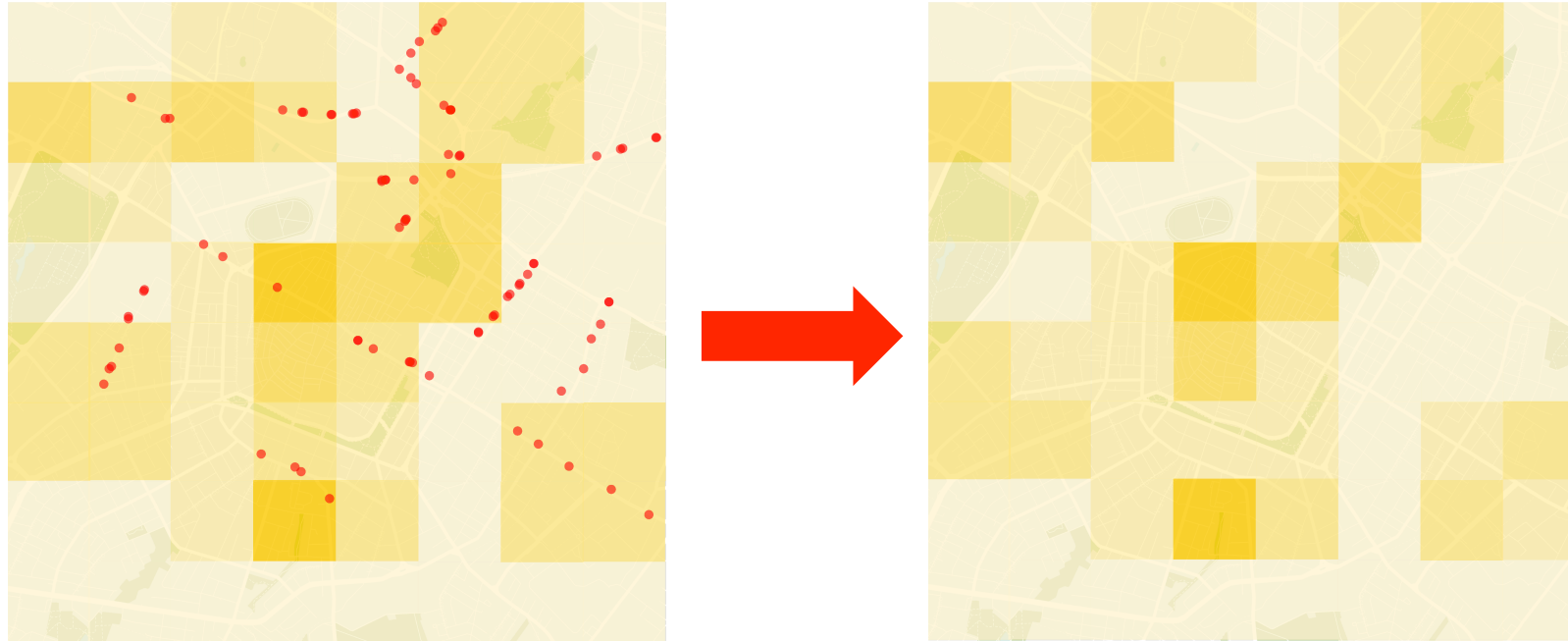


Map all filtered access points on **Phase I** heatmap

➤ Cluster nearby access points based on densities

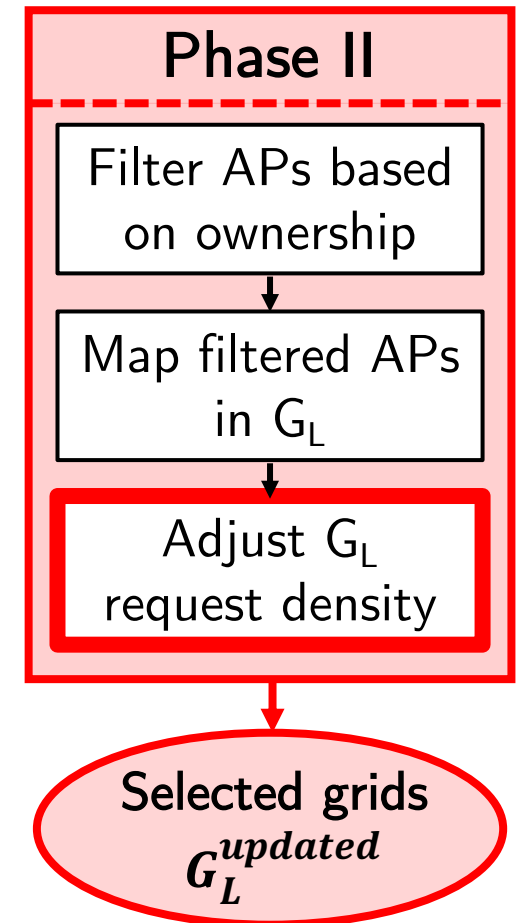


# Phase II: User Edge Incorporation

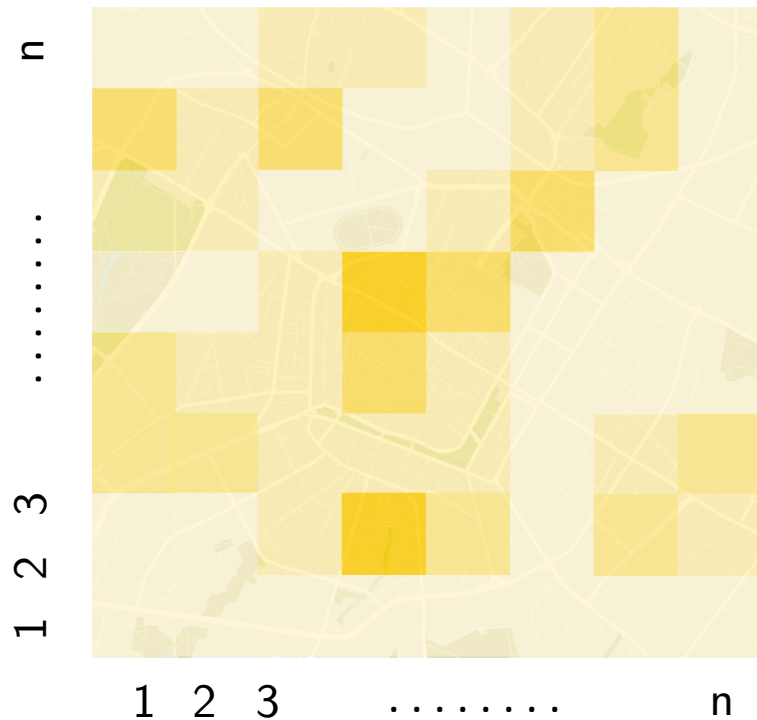


Reduce grid densities based access point availability density

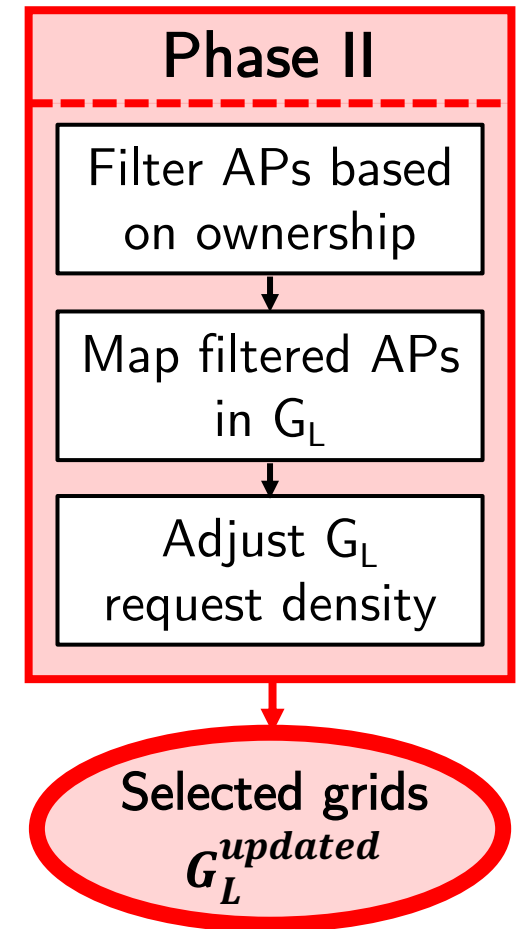
- Resulting heatmap denotes grids with overflowing user requests



# Phase II: User Edge Incorporation



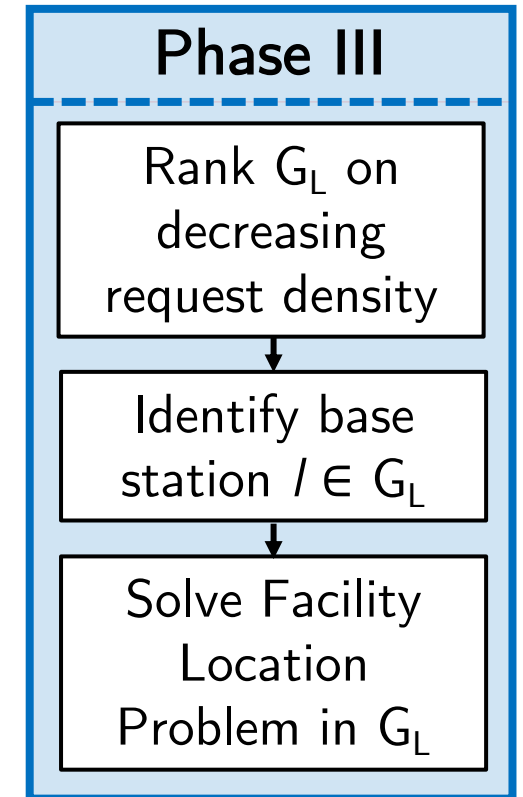
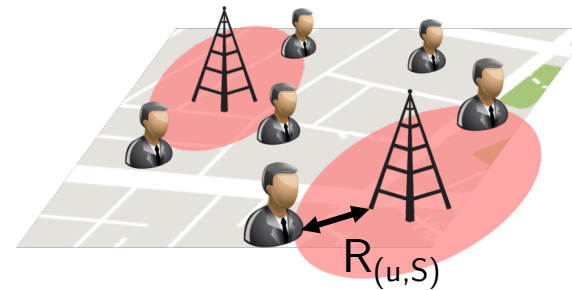
$$G_L^{updated} = \{g_{11}, d_{11}; \\ g_{12}, d_{12}; \\ g_{13}, d_{13}; \\ \vdots \\ g_{nn}, d_{nn}\}$$



# Phase III: Edge Location Selection

1. Select the **best set** of deployment locations considering connectivity to end users
2. Base stations are taken as possible deployment locations

Location selection best resembles  
**Facility Location Problem (FLP)**

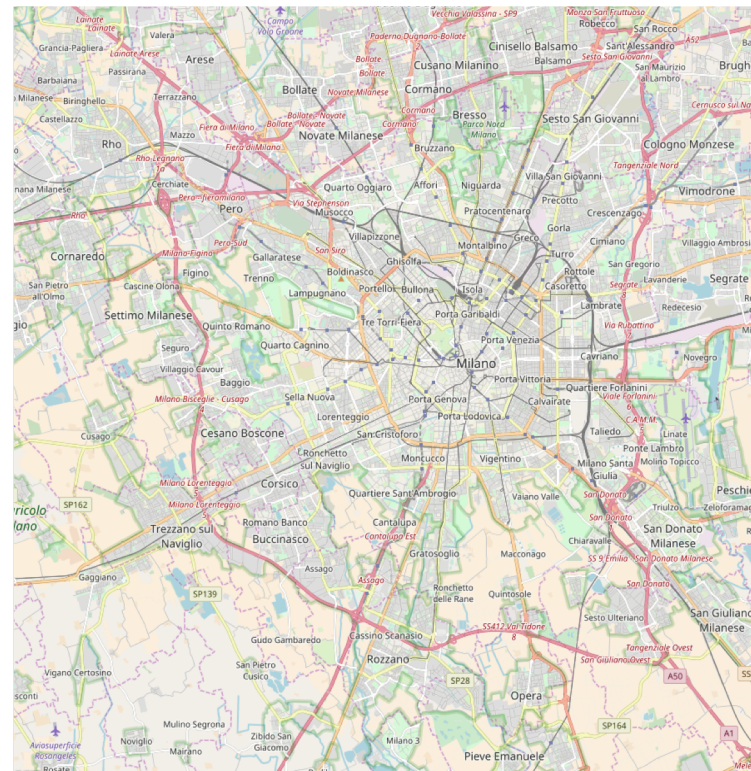


Please check our paper for more details

# Evaluation

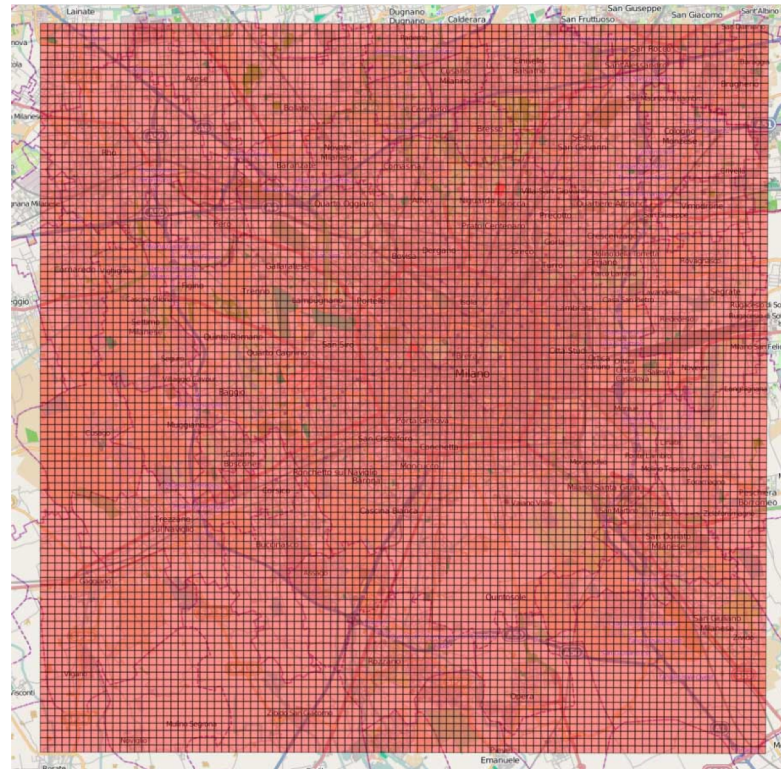
# Evaluation

Evaluate Anveshak's placement of ' $k$ ' edge servers on ' $n$ ' possible locations in Milan, Italy



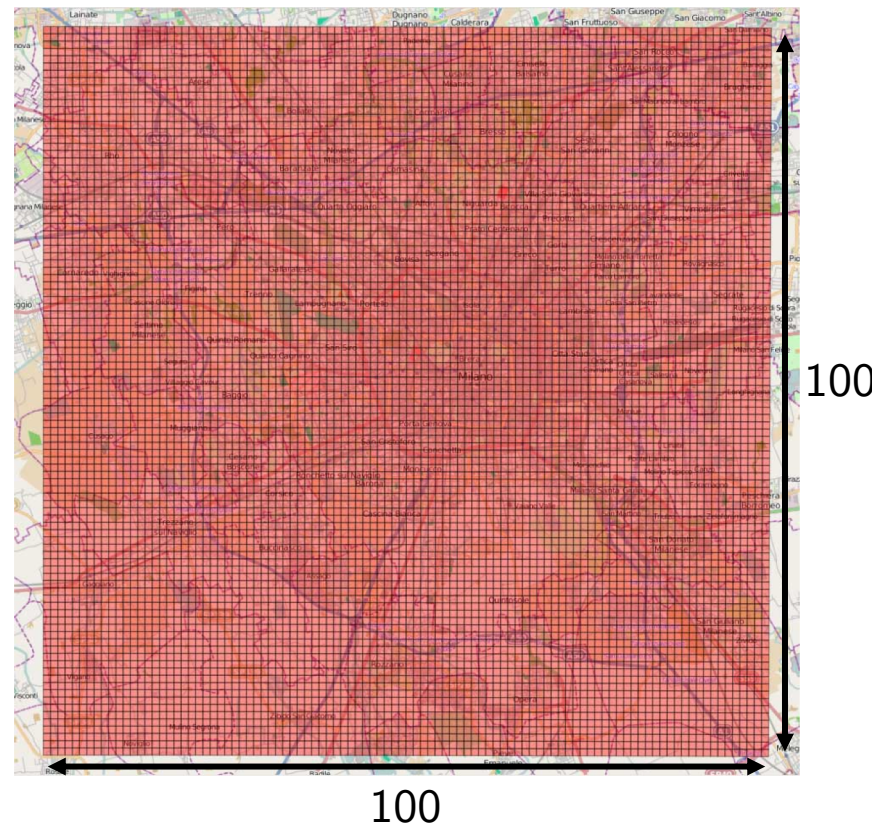
# Evaluation

Evaluate Anveshak's placement of ' $k$ ' edge servers on ' $n$ ' possible locations in Milan, Italy



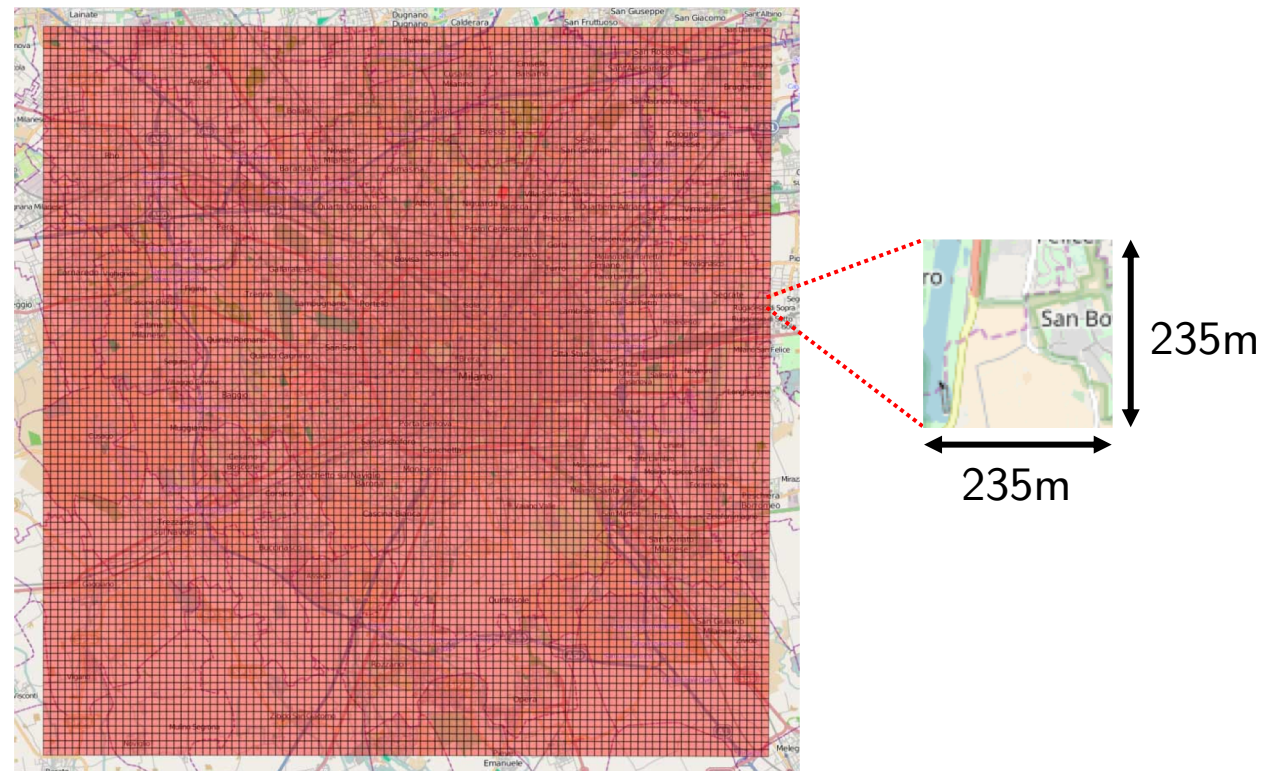
# Evaluation

Evaluate Anveshak's placement of ' $k$ ' edge servers on ' $n$ ' possible locations in Milan, Italy



# Evaluation

Evaluate Anveshak's placement of ' $k$ ' edge servers on ' $n$ ' possible locations in Milan, Italy



# Evaluation

Evaluate Anveshak's placement of ' $k$ ' edge servers on ' $n$ ' possible base-stations in Milan, Italy

## *A. With real datasets*

1. User requests: Published by Telecom Italia
2. Service-provider edge locations: Published by OpenCellid
3. User-managed edge locations: Published by WiGLE

# Evaluation

Evaluate Anveshak's placement of ' $k$ ' edge servers on ' $n$ ' possible base-stations

## ***B. With two approaches***

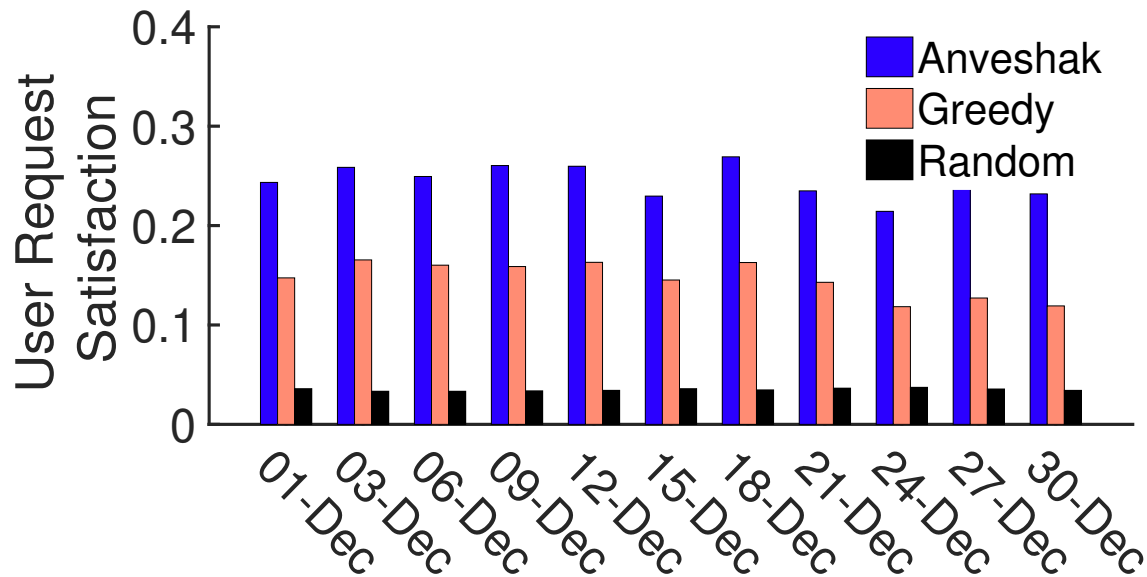
1. **Greedy:** Allocates user request densities to grids and selects *top- $k$*  maximum serving base-stations
2. **Random:** Randomly chooses  $k$  valid base-stations

# Evaluation

## Setup

1. User is mapped to nearest base station using coordinate based latency approximation
2. Deployment framework selects **50** edge location out of 850+

# User Request Satisfaction

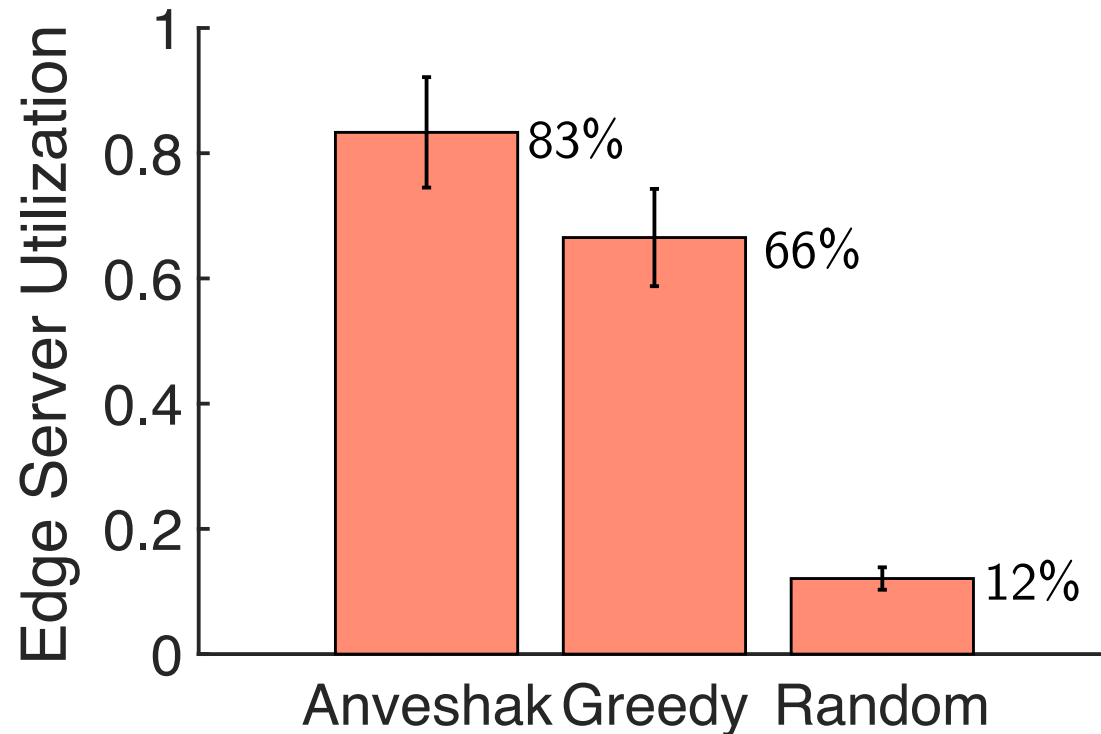


Q. How many user requests are handled by *Anveshak's* placed edge servers?

1. 67% more requests than *Greedy*
2. 25% of total requests handled by 8% of selected base stations

Can achieve 90% user satisfaction by placing 124 servers over 218 and 300+ by *Greedy* and *Random*

# Server Utilization



**Q.** How busy are the deployed servers?

**Assumption:** All user requests in a grid are first handled by user-managed edge

More user request density areas have more user-managed edge resources leading to less utilization of deployed server

# Conclusion

1. Anveshak is a deployment framework designed to assist service providers
2. It efficiently identifies optimal locations for edge server placement while considering:
  - Density of user requests
  - Density of future deployment of user-managed edge resources
3. We evaluate Anveshak and other deployment algorithms on real datasets
4. Anveshak achieves 67% increase in user satisfaction with 83% server utilization



# Thank You! Questions?

[nitinder.mohan@helsinki.fi](mailto:nitinder.mohan@helsinki.fi)

# Backup Slides

# How is Anveshak different from CDN server placement?

## Similarities:

1. Both problems must ensure consistent and least connectivity to end-clients
2. Both problems must optimize for cost of deployment

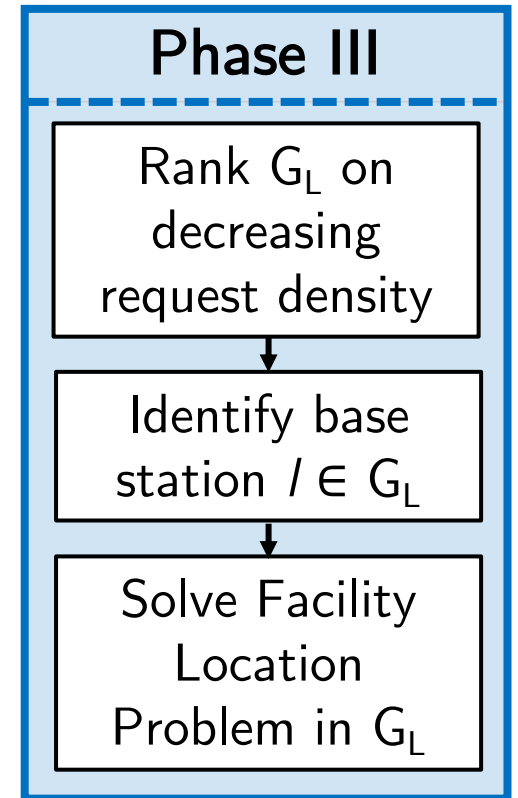
## Differences:

1. Unlike content, edge server handles requests which are short-lived and locally-relevant
2. Focuses more on server availability and network latency than network bandwidth

# Phase III: Edge Location Selection

Anveshak assumes that the deployed edge server will be co-located with a base station

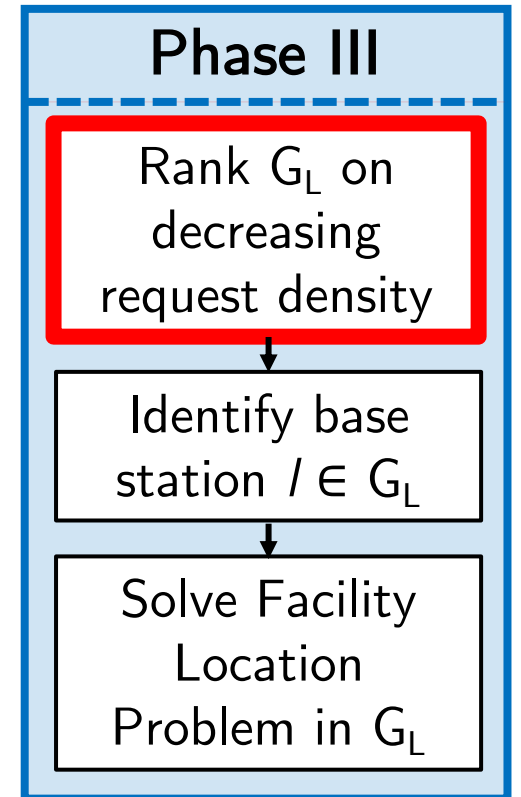
- From **Phase II** heatmap, select exact base station location which satisfies maximum user requests
- Selected base station must be able to service maximum users in 1-2 hops
- List of exact base station locations can be made available with partnering ISP



# Phase III: Edge Location Selection


Prioritize the grid locations based on *updated* user request density

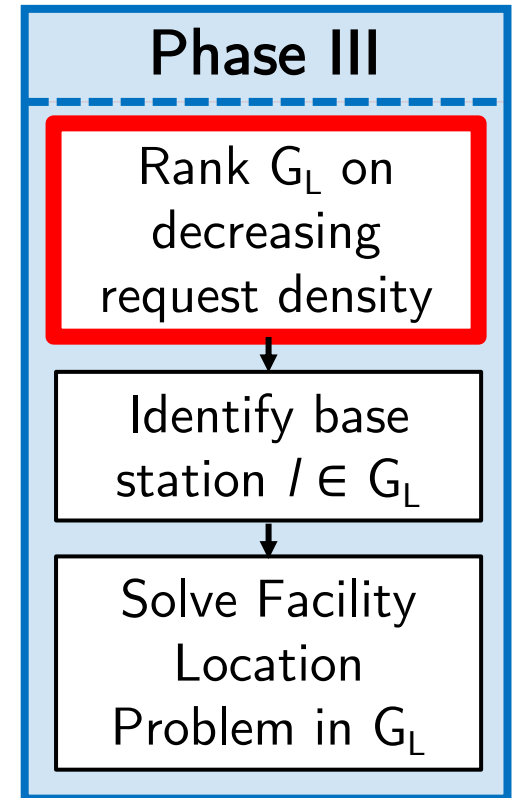
$$G_L = \{g_{11}, d_{11}; \\ g_{12}, d_{12}; \\ g_{13}, d_{13}; \\ \vdots \\ g_{nn}, d_{nn}\}$$



# Phase III: Edge Location Selection

Prioritize the grid locations based on *updated* user request density

$$G_L = \left\{ \begin{array}{l} g_{11}, d_{11}; \\ g_{12}, d_{12}; \\ g_{13}, d_{13}; \\ \vdots \\ g_{nn}, d_{nn} \end{array} \right\}$$


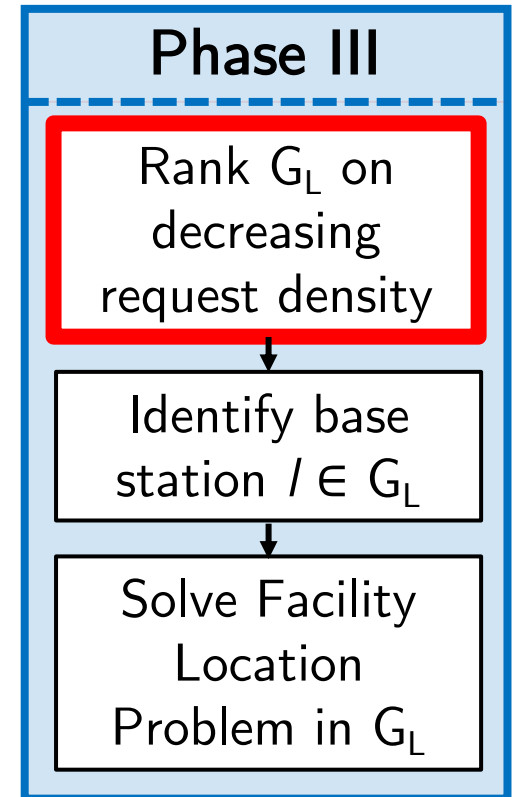


# Phase III: Edge Location Selection

Prioritize the grid locations based on *updated* user request density

$$G_L = \{g_{13}, d_{13}; \\ g_{32}, d_{32}; \\ g_{61}, d_{61}; \\ \cdot \\ \cdot \\ \cdot \\ g_{n7}, d_{n7}\}$$

where,  $G_L\{d_1\} > G_L\{d_2\} > \dots > G_L\{d_{last}\}$



# Phase III: Edge Location Selection

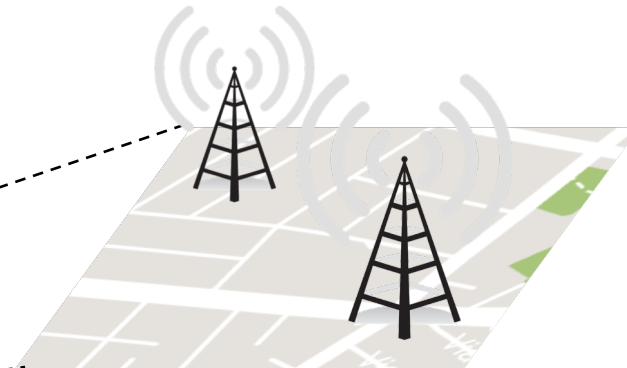
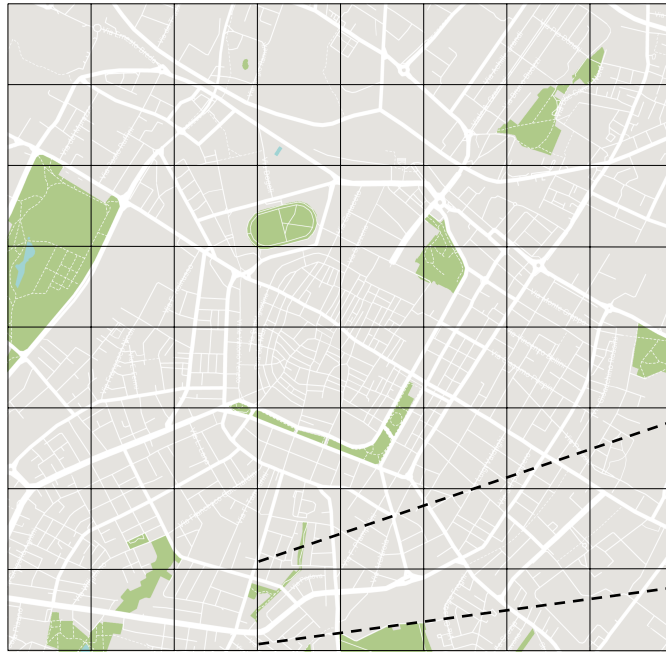
$$G_L = \{g_{13}, d_{13};$$

$g_{32}, d_{32};$

$g_{61}, d_{61};$

$\vdots$

$g_{n7}, d_{n7}\}$



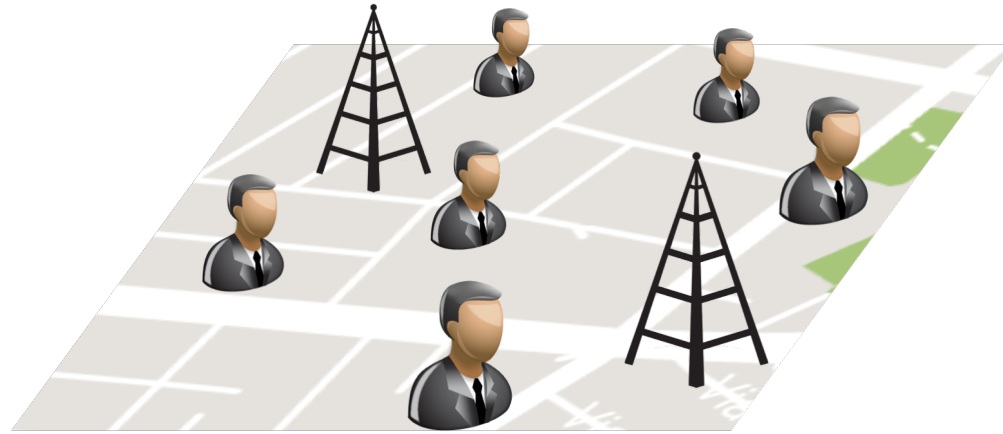
## Phase III

Rank  $G_L$  on decreasing request density

Identify base station  $i \in G_L$

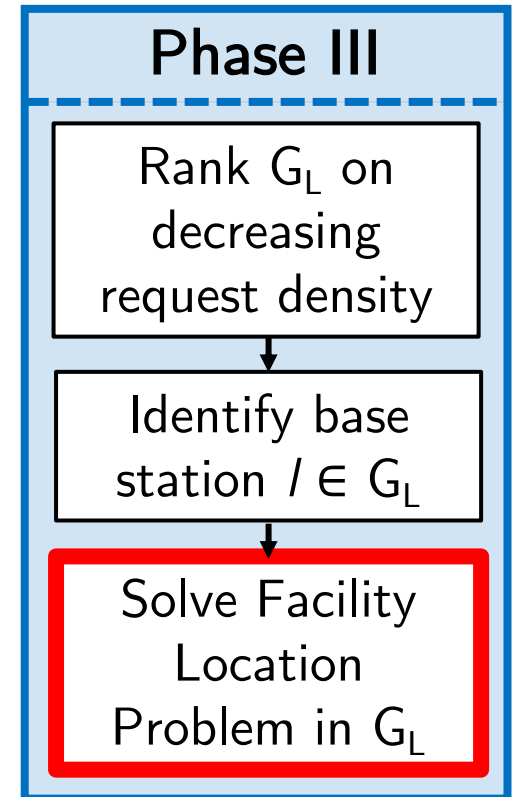
Solve Facility Location Problem in  $G_L$

# Phase III: Edge Location Selection

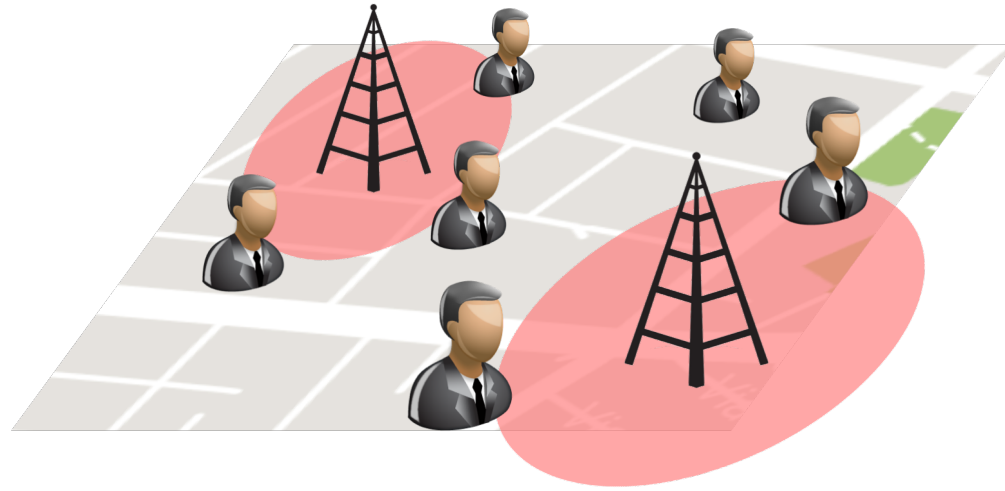


Select optimal base station ( $S_i$ ) in  $G_L[i]$  satisfying maximum user requests [Phase I]

➤ We focus on **location** and not density

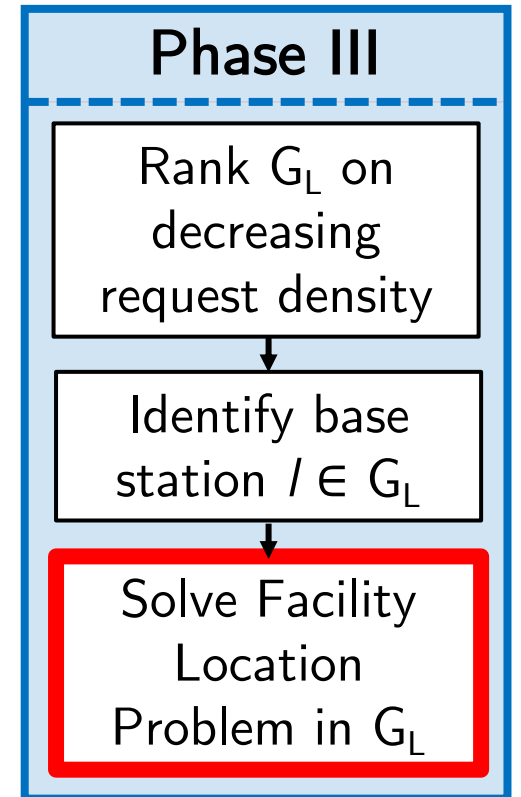


# Phase III: Edge Location Selection

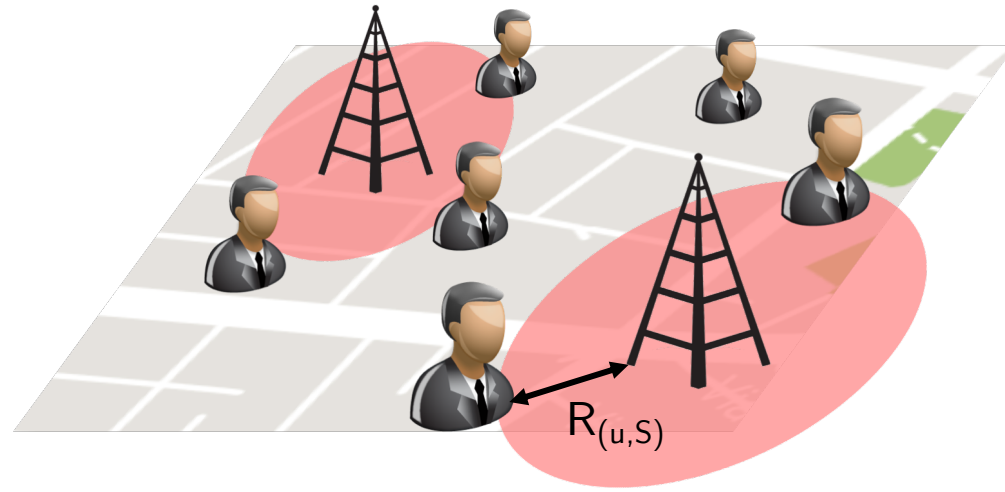


Select optimal base station ( $S_i$ ) in  $G_L[i]$  satisfying maximum user requests [Phase I]

➤ We focus on **location** and not density



# Phase III: Edge Location Selection

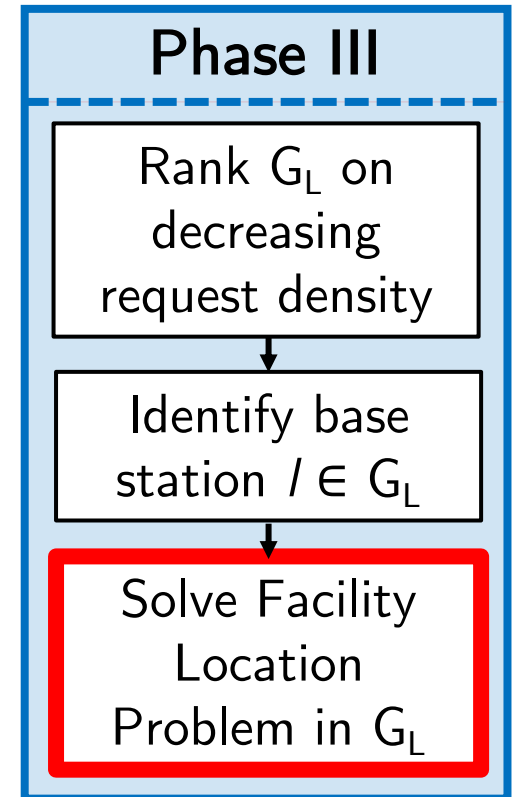


Network cost of connectivity ( $n_{(u,S)}$ )

$$n_{(u,S)} = \alpha * R_{(u,S)}$$

where,  $R_{(u_l, S_l)}^{max} = \max[u - S_l] \quad \forall u_l \in U \quad \& \quad l \in L$

$\alpha$  = maximum server access cost

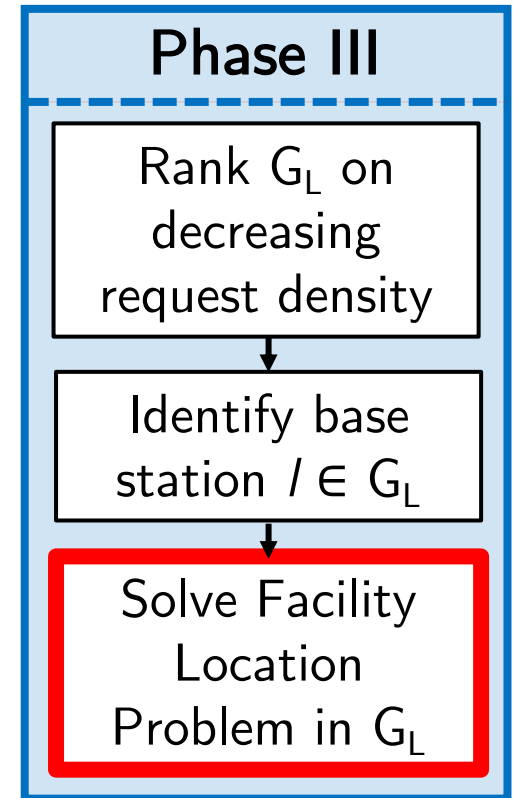


# Phase III: Edge Location Selection

**Objective:** Minimize one-hop latency between users and selected server location

$$S_u = \min \sum_{l \in L} \{S_l \mid S_l \in S, n_{(u,S_l)} < n_{max}\} x_l$$

where,  $x_l \in \{0,1\}$



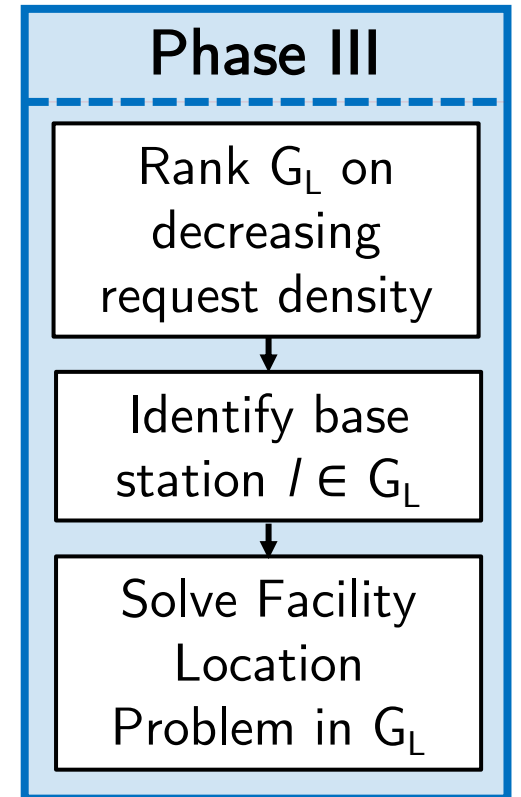
# Phase III: Edge Location Selection

$$[S_u = \min \sum_{l \in L} \{S_l \mid S_l \in S, n_{(u, S_l)} < n_{max}\} x_l]$$

- NP-hard problem
- Approximation-based solvers available [ODL, Sitation]

## Anveshak's Advantage:

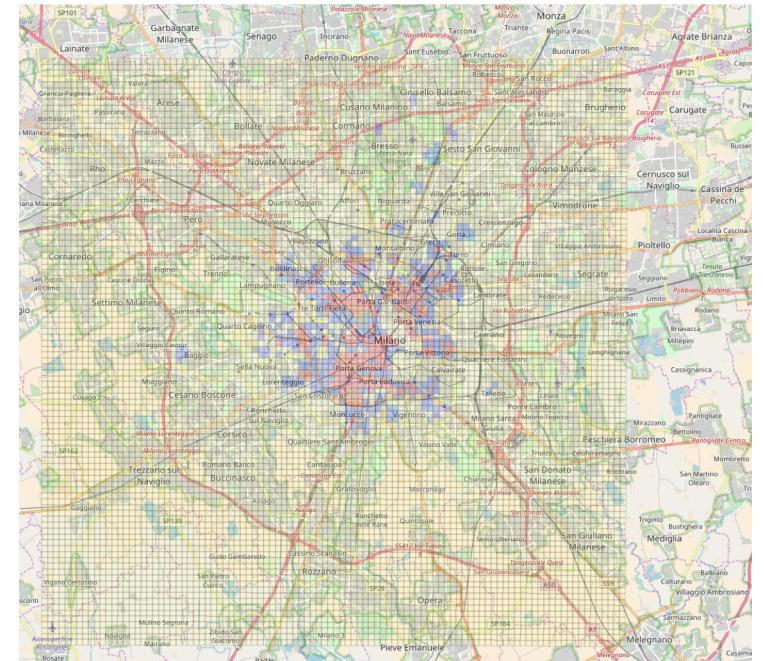
- Grid size as additional constraint, reduce problem size
- Due to problem's future outlook, exact optimization is **not** required!



# Datasets

## (1/3) User Request Dataset

- Published by Telecom Italia\* for Milan, Italy
- Anonymized details of Call Detail Records (CDRs), internet connectivity of users in the region
- Data of November 1<sup>st</sup> to December 31<sup>st</sup> 2013

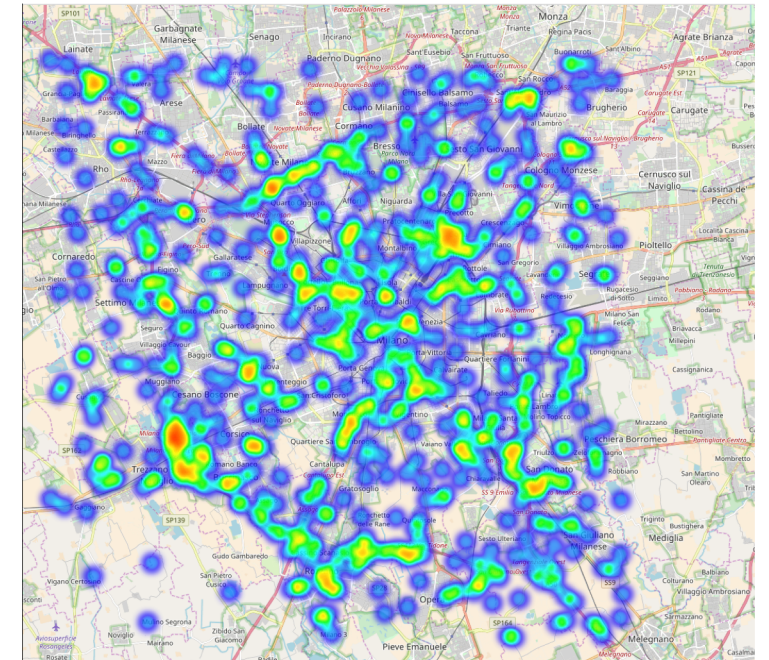


\*<https://dandelion.eu/datamine/open-big-data/>

# Datasets

## (2/3) Service-provider Edge Dataset

- Published by OpenCellid\*
- Details of cellular base stations such as connectivity type (3G/4G), area etc. along with their GPS coordinates
- *Post filtration*: 800+ LTE base stations
- Each base station is associated to the map grid



# Datasets

## (3/3) User-managed Edge Dataset

- Published by WiGLE\*
- Details of WiFi access points such as SSID, signal strength, channel number etc. along with GPS coordinates
- 800+ base stations Milan, Italy serving LTE connection

