UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche,
Informatiche e Matematiche

IoT Systems

Crowdsourcing

Prof. Luca Bedogni

Crowdsensing

Crowd Sensing

Crowdsensing

Crowdsensing

Crowd Sensing

Who records the data

Crowdsensing

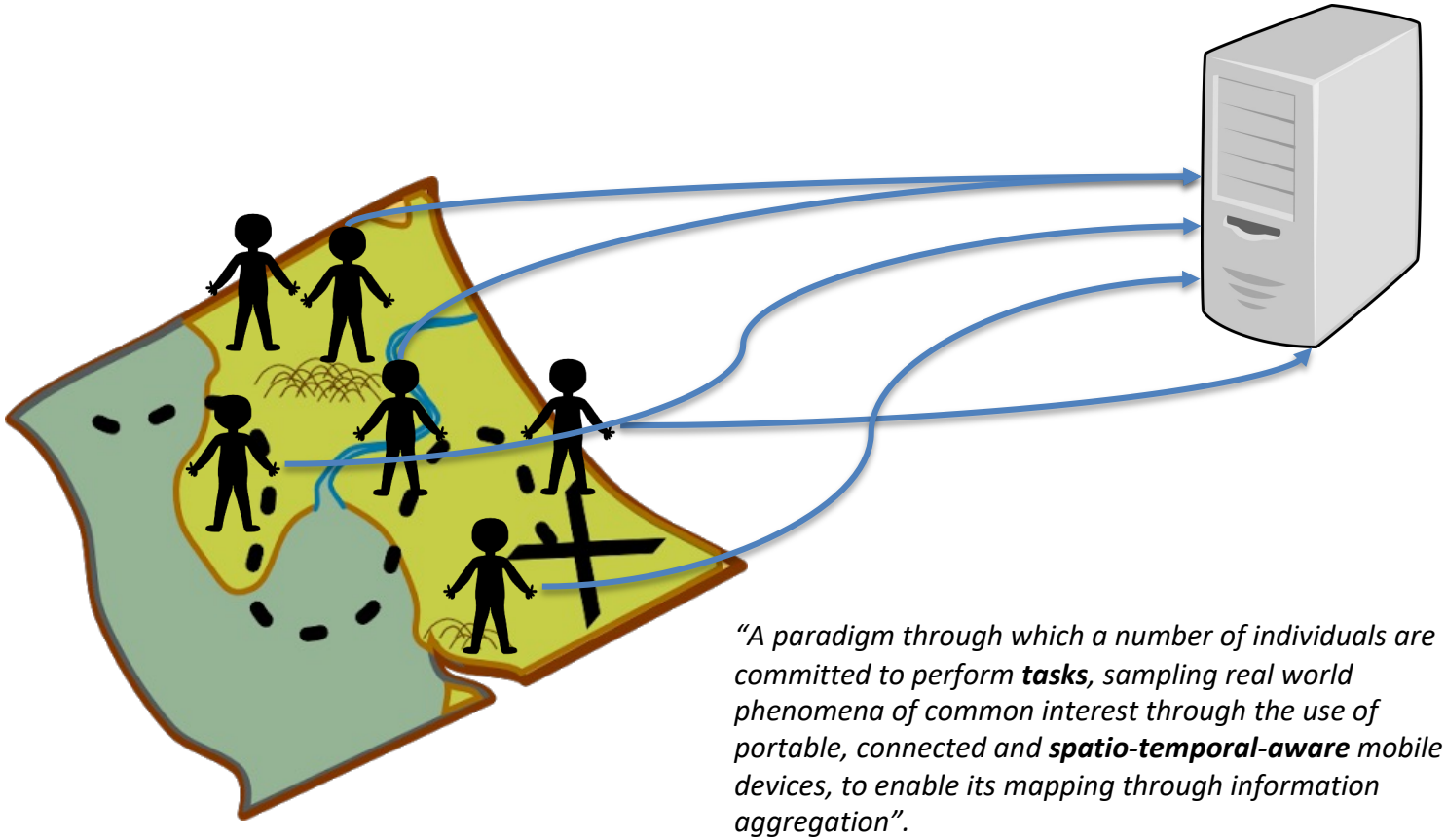
Crowdsensing

The action performed

Crowd Sensing

Who records the data

Overview



*"A paradigm through which a number of individuals are committed to perform **tasks**, sampling real world phenomena of common interest through the use of portable, connected and **spatio-temporal-aware** mobile devices, to enable its mapping through information aggregation".*

Crowdsensing overview

- Crowdsensing or crowdsourcing
  - For our purposes, we can consider them the same
  - To be more precise
    - Crowdsourcing: outsourcing some service (general)
    - Crowdsensing: getting sensor signals (more specific)
- Crowdsensing lets you get the data from several users

- There are (mainly) two types of crowdsensing

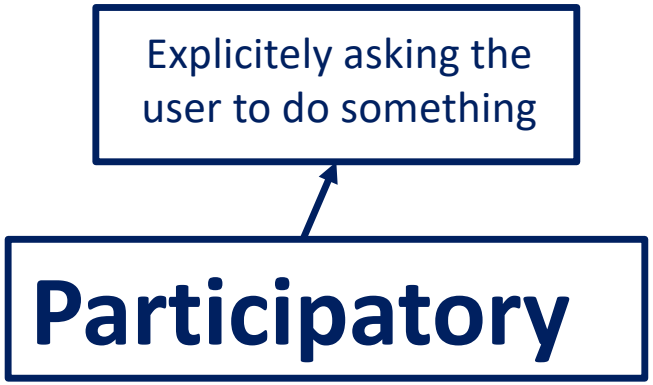
Participatory or Opportunistic

Participatory

Opportunistic

CrowdSensing

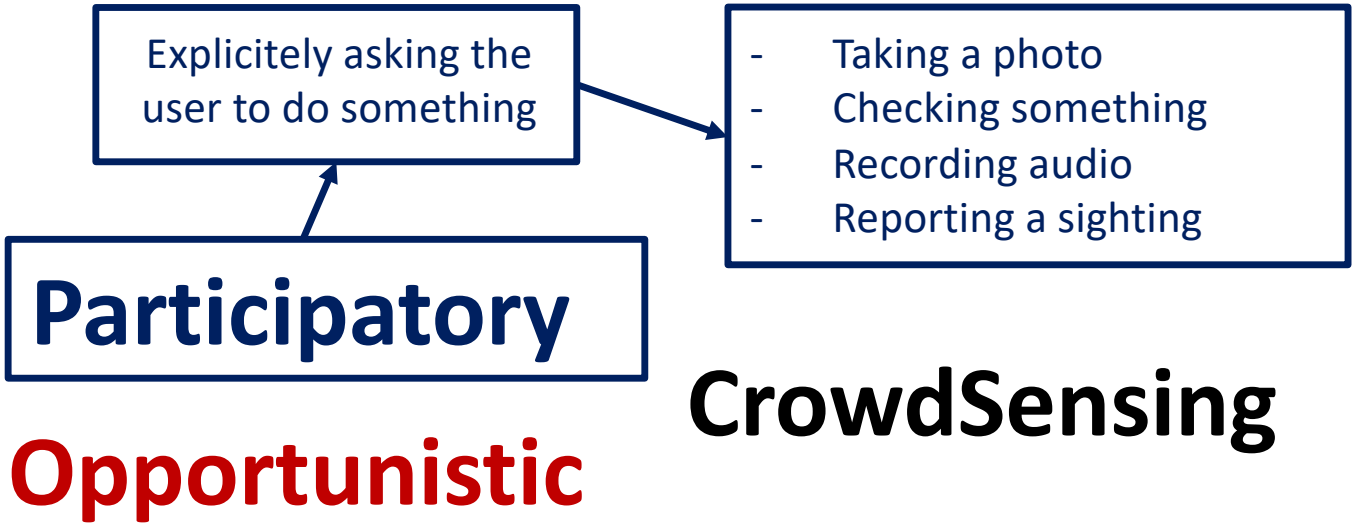
Participatory or Opportunistic



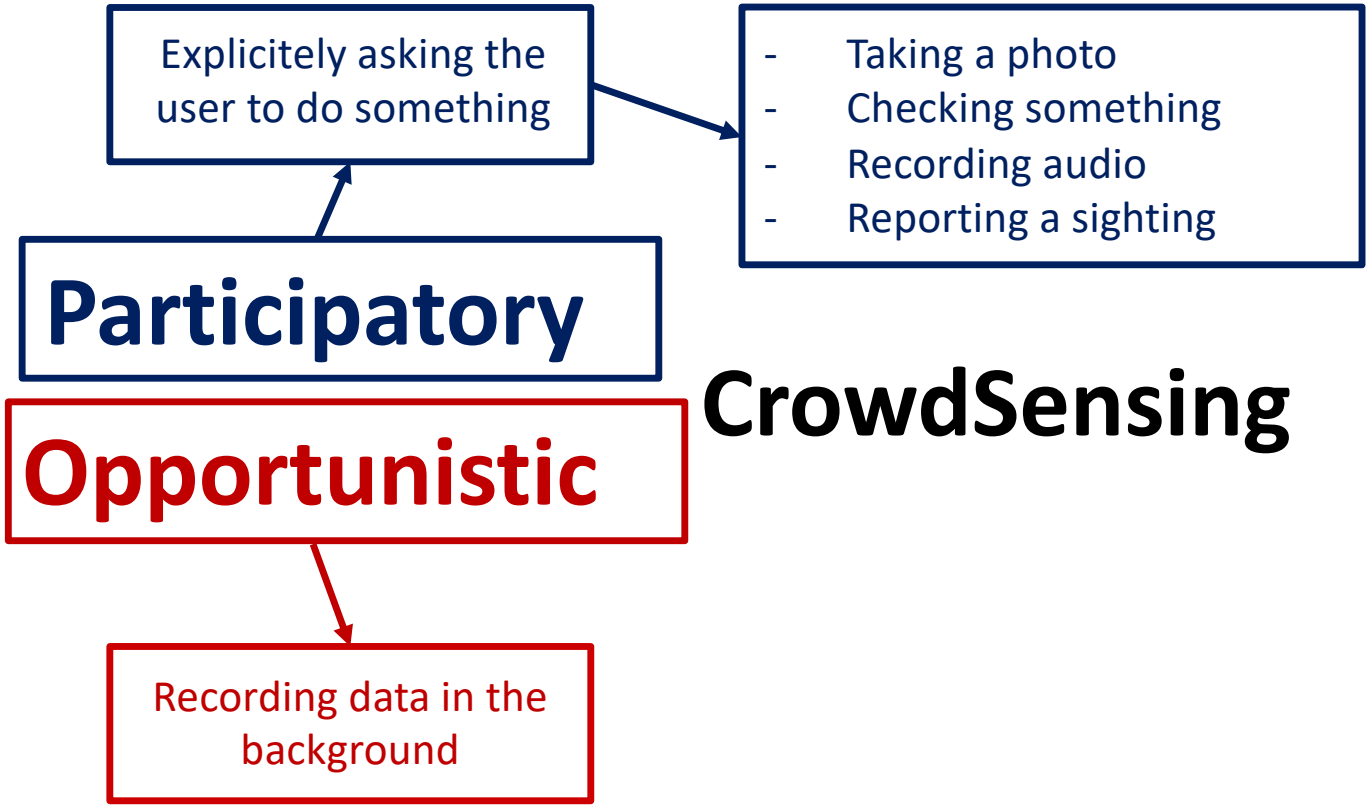
Opportunistic

CrowdSensing

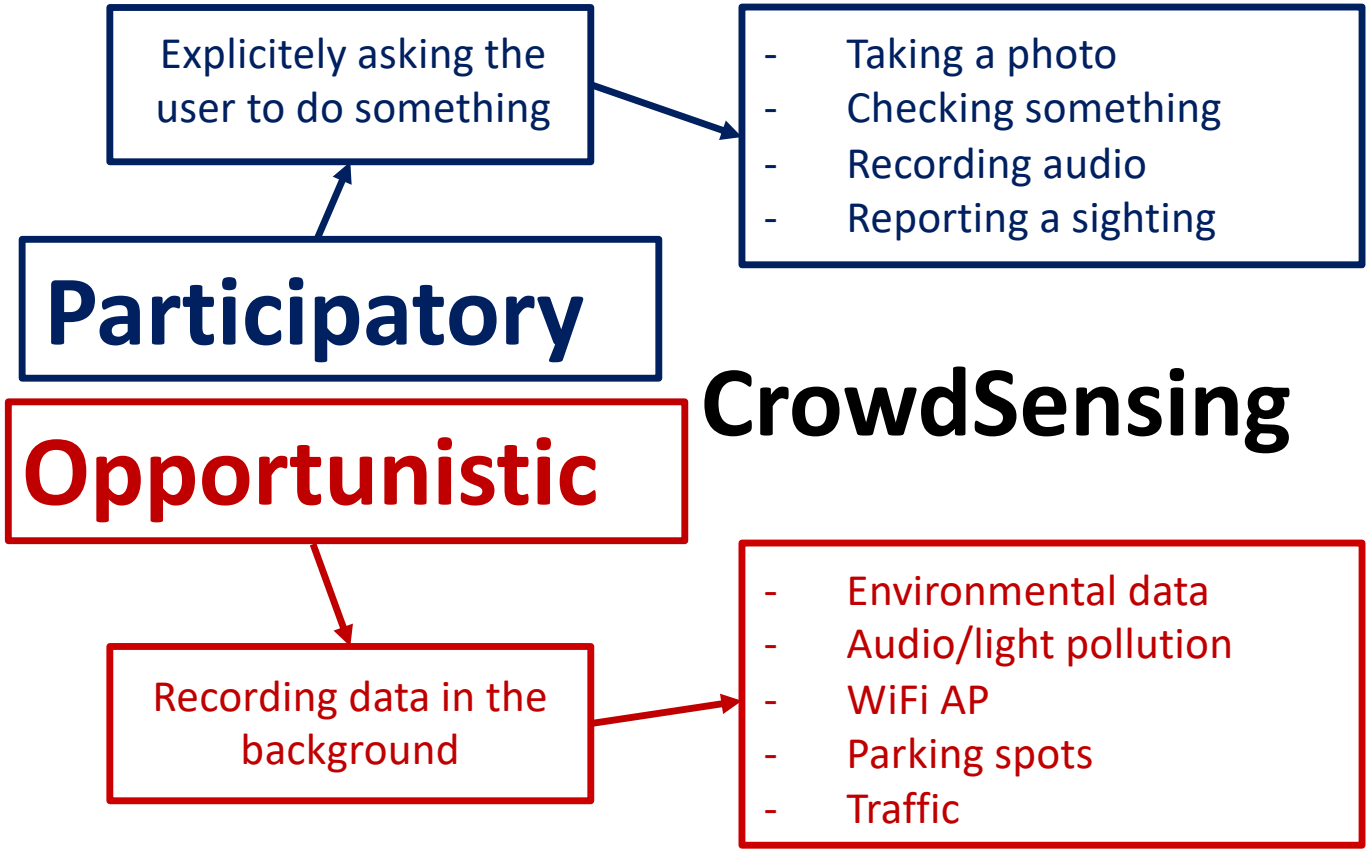
Participatory or Opportunistic



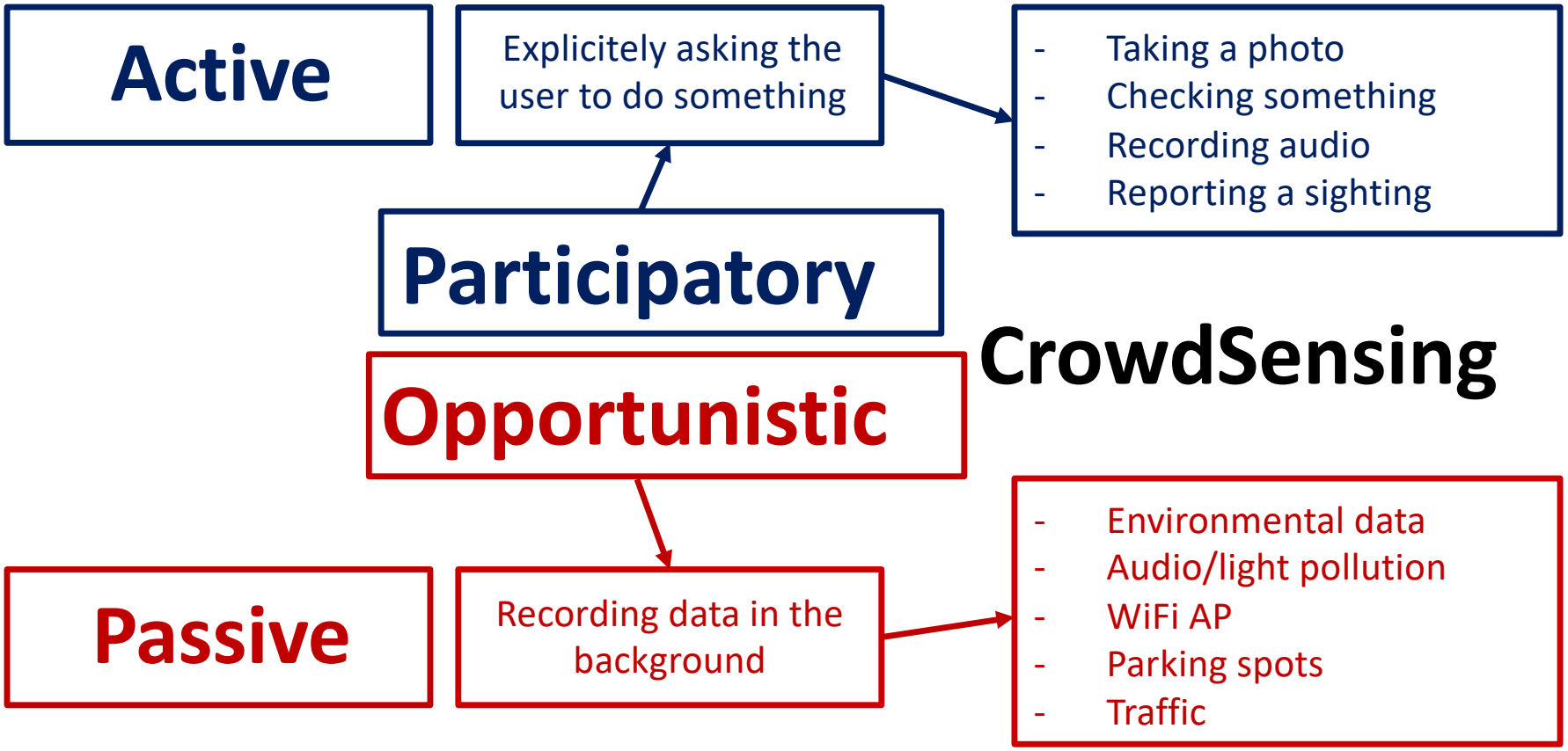
Participatory or Opportunistic



Participatory or Opportunistic



Participatory or Opportunistic



Why crowdsensing

- Because sometime it is not feasible to build an infrastructure
  - Expensive to build, to maintain, very rigid
- Sometimes you need sporadic data
  - E.g.: photos of natural events
- Sometimes you need the power of the crowd
  - Hard to achieve the granularity of data to match crowdsensing

Infrastructure not feasible

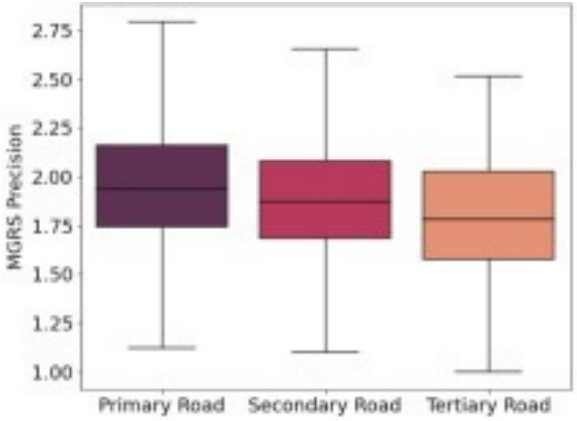
- Too large area
- Expensive to design, maintain
- Requirements may change over time

Sporadic data

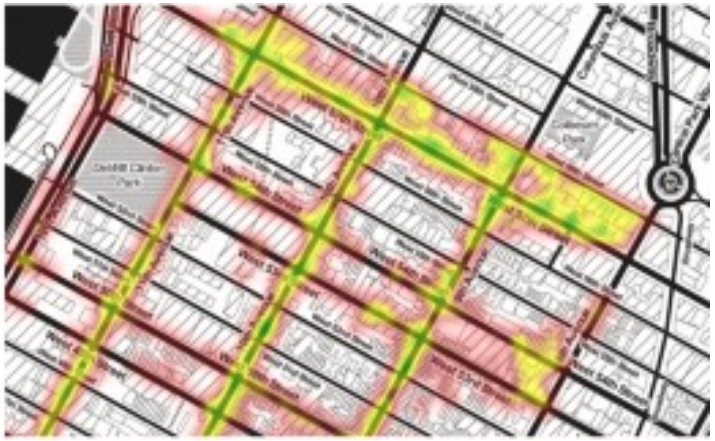
- Not meaningful to maintain a fixed infrastructure for a prolonged time
- Concerts, events, sporadic phenomenons
- Easier to deploy and destroy

Power of the crowd

- Phenomenon widespread
- Difficult to predict where it may be originated



(a) Study on road type



(b) Map example

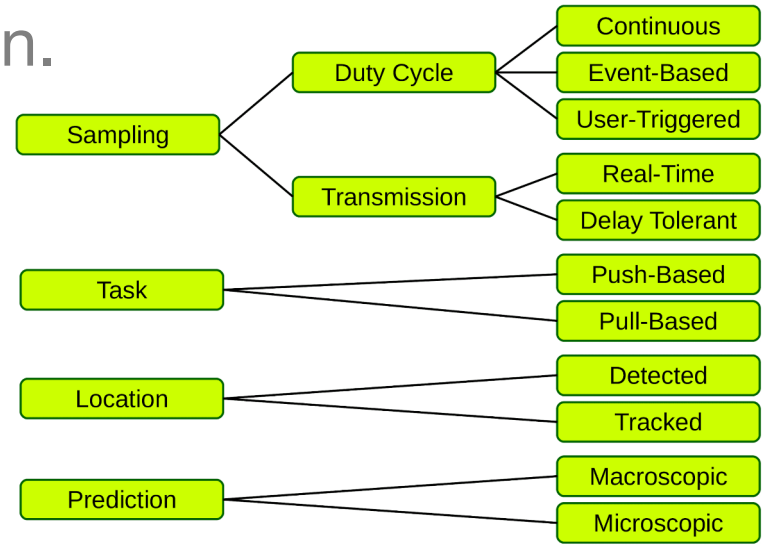
Mobile or not

- Mobile crowdsensing records data on the move
- Typically, this is done through smartphones
- But it can be done also with specialized hardware

- Crowdsensing can also be static
  - Though in this case scenarios are more limited

MCS Taxonomy

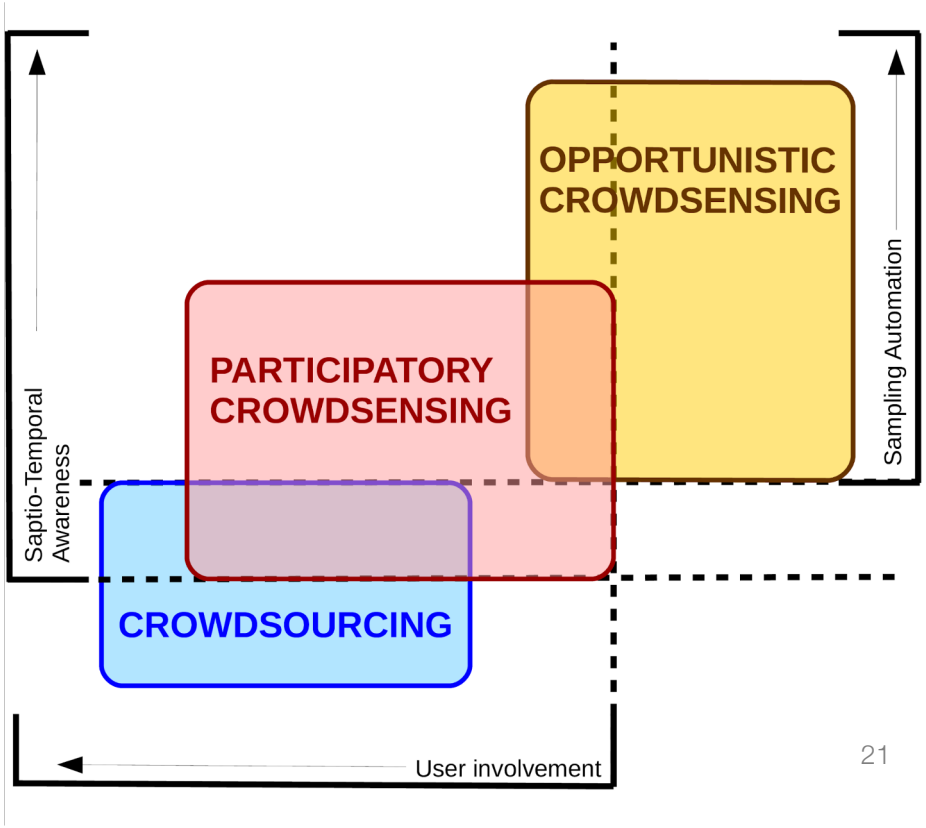
- There are many many different factors that model an MCS scenario.
- Difficult to create a universal paradigm to include them all.
- Dependance on the field of application.
- E.g: traffic detection
  - Gmaps
  - Waze



Collective Awareness

- MCS is a Collective Awareness Paradigm (**CAP**).

- Categorized upon:
  - Spatio temporal awareness
  - Degree of User Involvement
  - Sampling Automation



Challenges in crowdsensing

- There are many
- We will focus here on three of them
  - **Data quality and Data collection**
    - How to collect data, and how to keep quality data? How much data is enough? How to gather it efficiently? How many participants can contribute?
  - **Reward**
    - No one is ever doing anything for free. How to reward your users? How to involve participants? How not to penalize certain users?
  - **Privacy**
    - By giving away data, you are giving away private information. How to deal with this?

Data quality and Data noise

- When collecting a lot of data, not all of it may be worth keeping
  - Some users may report higher quality data than others
    - Better sensors, better camera, better skills, more precise users
  - Some users may report too much data
    - Not meaningful to report 10 temperature readings 2 meters apart from each other
  - Some data may not be useful (or even negative to keep)
    - Imagine getting noise data on the street or inside a telephone box. Same location, much different results
- So how to deal with this?

Data quality and Data noise

- Let's start from WHERE we have to perform the recognition
- **Before sensing**
  - Sensor quality, frequency of change
- **After having sensed the valued**
  - Edge computing, keep in mind devices constraints
- **On the cloud**
  - Aggregating values with others to find (and delete) outliers

Integrate heterogeneous data

Nowadays petabytes of data regarding the world we are living in are produced all around us.

Regardless on the platform we will use, we will just see a small part of it.

Too many separated ecosystems are working within their data islands → No interoperability

There is plenty of data published by institutions on available platforms → Reliable Open Data

Users can publish their data to open platforms too → Crowdsensed Data

IoT? Island of things

UNIMORE

SenSquare

Environmental monitoring

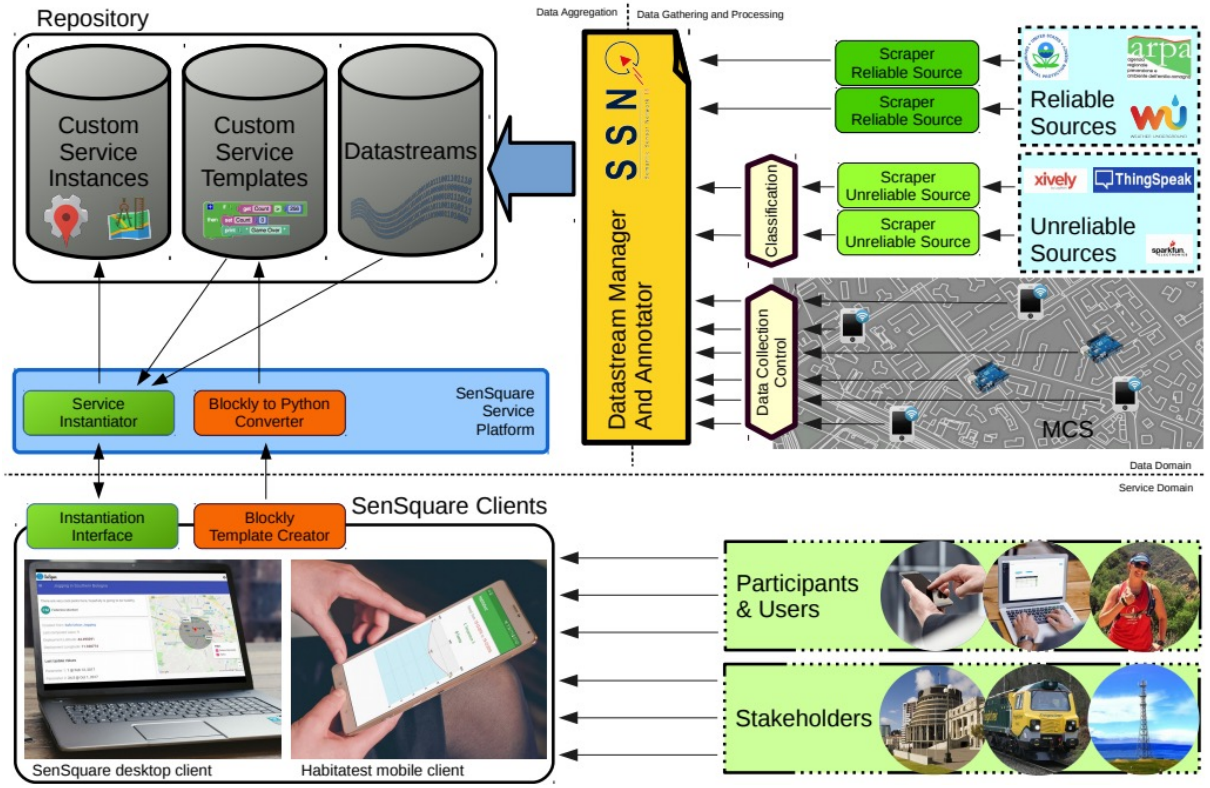
Traffic status

Public transportation conditions

Cellular connectivity

Noise pollution

Leveraging MGRS for hierarchical measurements



UNIMORE

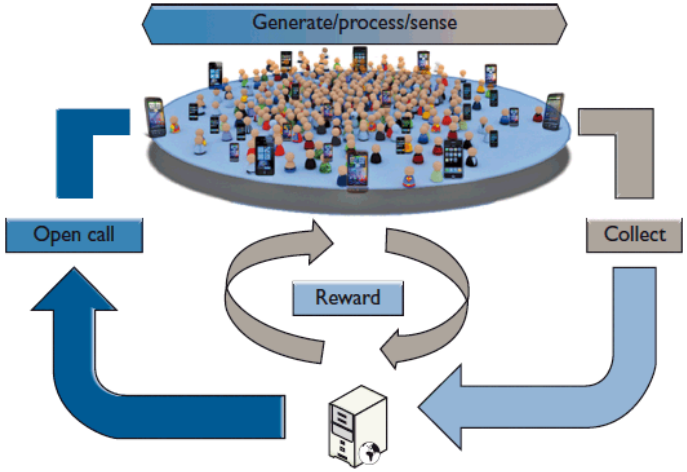
Issue solved?

- Almost…
- Relying totally on opportunistic crowdsensing may leave areas unexplored
  - More measurements where people travels the most
- There is no coverage guarantee
- Typically specific tasks are allocated to users (participatory crowdsensing)
  - How to select the appropriate tasks?
  - How to appropriately reward users?

Crowdsensing systems
Reward

- For each measurement, you get a simple reward
- Accumulate N rewards and get a prize
- Even more complex systems in which there are different levels of rewards
  - Based on location popularity
  - Based on measurement rarity
  - Based on specific requests

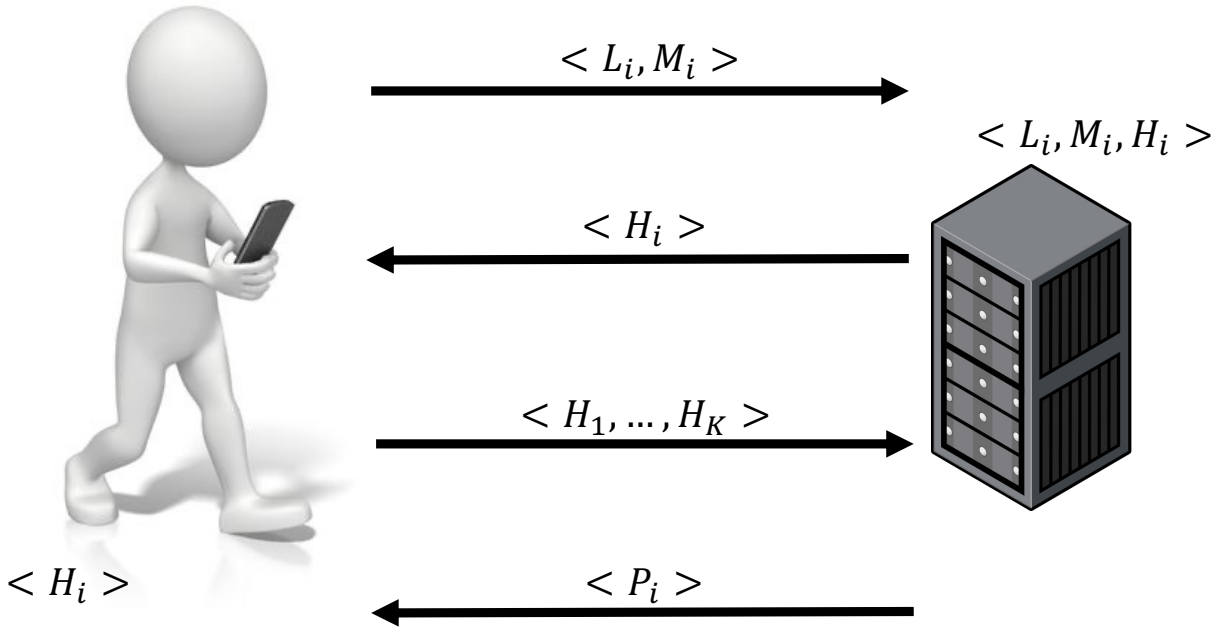
- How to assign rewards?
  - **Flat rate**
    - User who work more are paid as others
  - **Track users**
    - Keep reward information
  - **No user tracking**
    - ?



Crowdsensing systems
Reward

- The goal of the campaign owner is also to select the lowest number of users
  - Which satisfy the campaign needs
  - Which fit within the budget
  - Which provide a certain data quality
- Usually there are auctions to advertise a task
  - Not all users may be asked to participate
- Challenges:
  - **How to assign rewards without online tracking users**
  - **And without giving the possibility of doing it offline**

Idea



- Client sends measurement along with its location
- Server stores it locally, together with an hash, which is sent back to the client and stored locally.
- Upon reaching at least K hashes, the client can select the reward by sending them to the server
- The server checks that all the hashes are correct. If so, it deletes them from its internal memory and sends back a prize to the client.

Example

Collect several measurements (N)

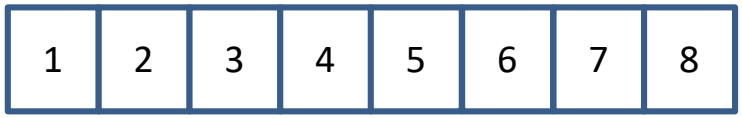
- More than what it is needed for the prize

When sending data to the central entity, store an hash for the reward

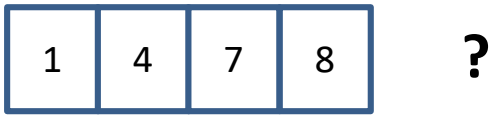
Upon collecting the prize, send K hashes

How to select the K best possible hashes?

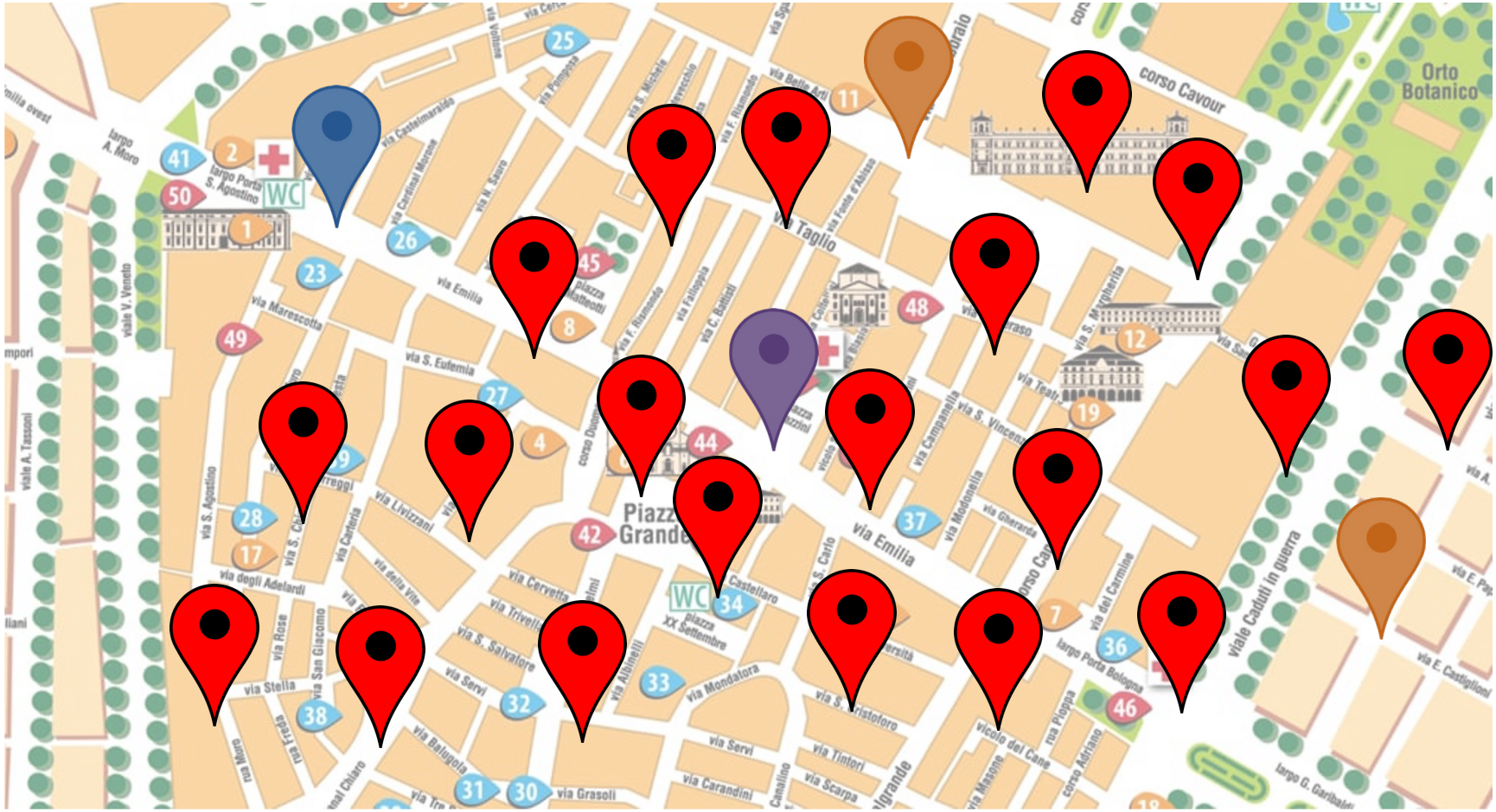
- Find the best subset K among N elements
  - The one which minimizes the privacy metric



Metric based on instantaneous and predictive privacy

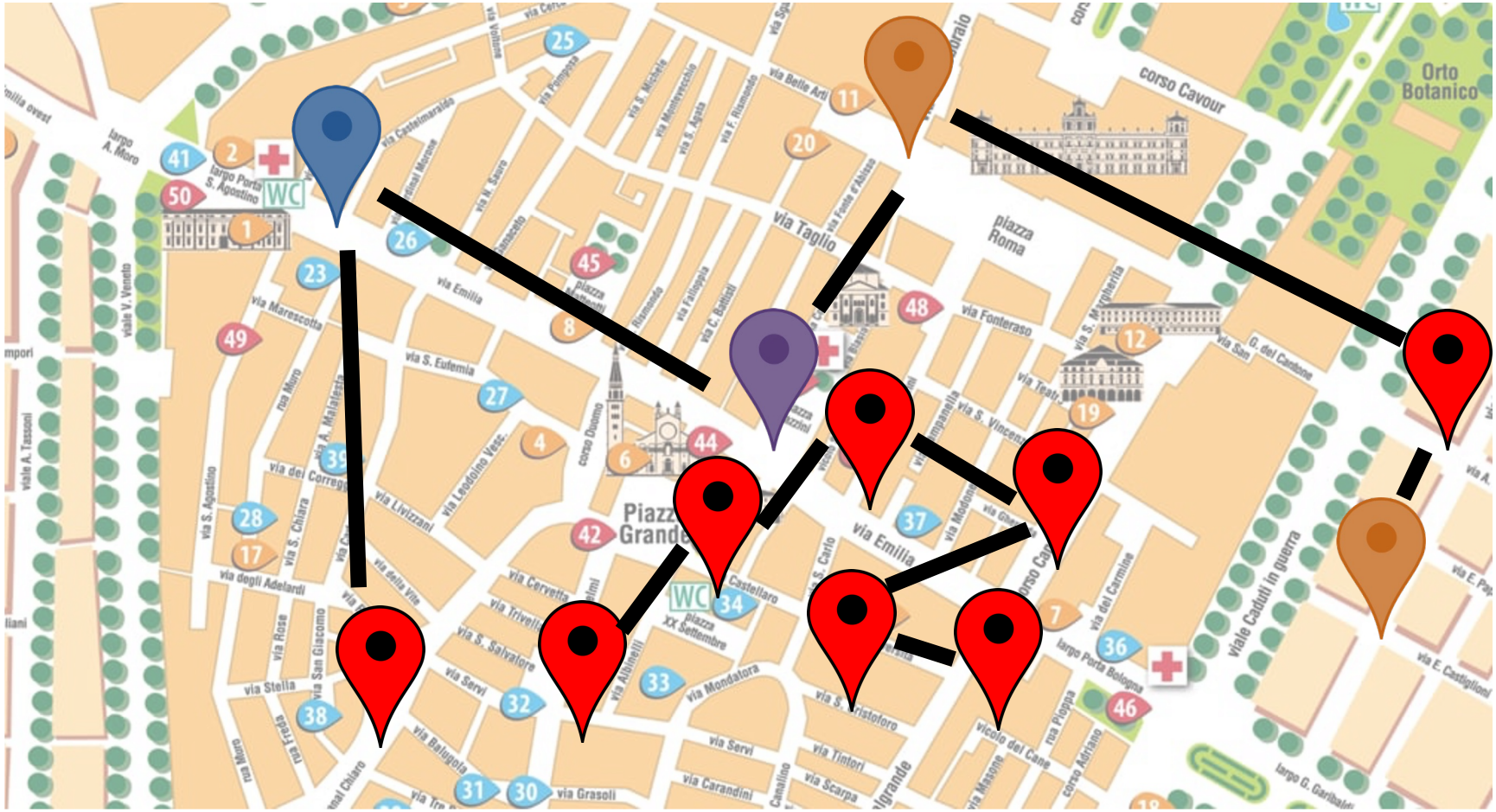


Prize set at 6 measurements



UNIMORE

Prize set at 6 measurements



UNIMORE