# Approach 1

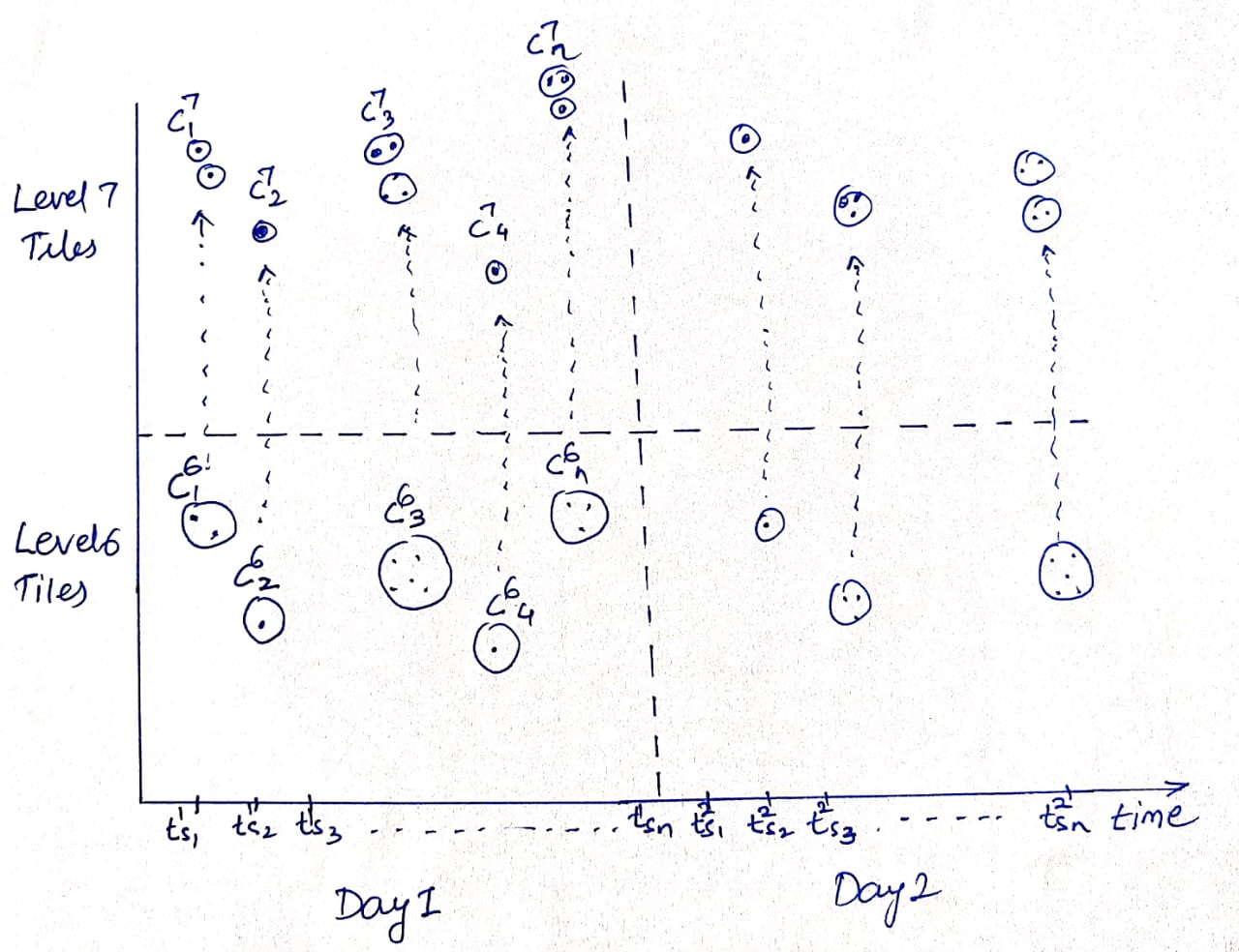
# Data as it exists

* Data is presented in tiles and these tiles are hierarchically organized. E.g. for London, *level 7*(6x6 meters) tiles are hierarchically one level below *level 6* tiles (25x25 meters)
* Each tile has the following structure:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date-time | Tile id | Devices | New devices | Average Lat | Average Lon | Ping | Average HPE | Approx Median HPE |

\*HPE = Horizontal Positioning Error

Based on this, we can organize the data day wise in 15 min timeslots across the various hierarchies for a City/ Area of city



Representation and flow of clusters are bi-directional

are bi-directional

* Each cluster represents a structure average lat-lon & each dot represents a device.
* Each cluster may have one or more devices i.e**. multiple devices represented by same lat-lon(average of their coordinates)**

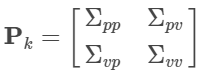
# Implementation using Kalman Filter

* Using Kalman filter, we can represent each state for an initial time slot as the

*where* ***p = [px, py]T*** *is the position vector****, v = [vx, vy]T*** *is the velocity vector and* ***d*** *is the initial number of devices at position p*.

* Velocity can be estimated from the unique device movements.
* Using the data across multiple timeslots, we will estimate the next position with some uncertainty using the previous state information from time k-1 to time k as:

And the covariance matrix is defined as below assuming that p and v are not correlated:



* Working across successive Bi-shots of time slots, we will modify the Covariance matrix and predict the next position of a device.   
  Assuming,

Which implies that,

**=   
 =**

* Adding external influence in the model

Which implies that,

*But currently we don’t have any way to understand or estimate Acceleration from data, so we’ll try to model it in terms of uncertainty*

* Distance across clusters can be computed using the average coordinates (latitude/longitude)
* For the clusters, where we have multiple devices assume a distribution (empirically or Bootstrapped) with mean latitude/longitude and standard deviation calculating, to estimate the surrounding/ more precise positions.
* Compute the distance thereafter to get an estimate of SDI which needs to be validated on the historical data.

*\*Note: Kalman filter will work best when each device id and its location will be given.*