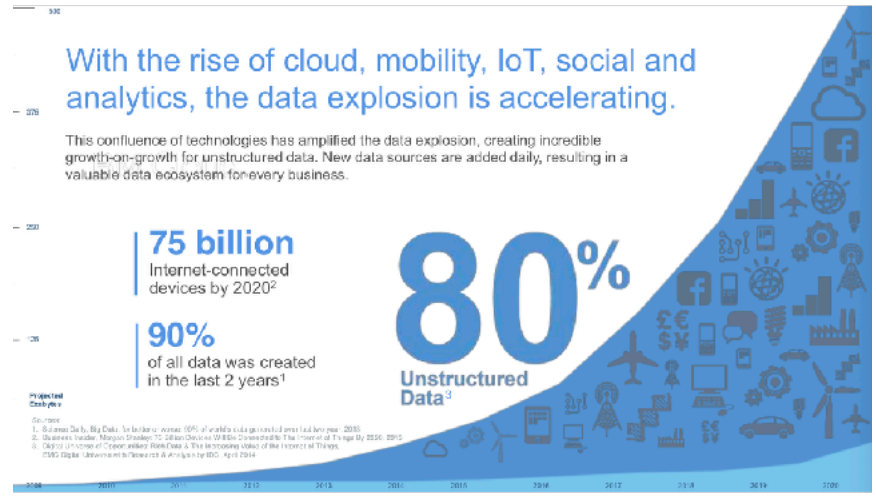
Local Differential Privacy and trade-off with Utility

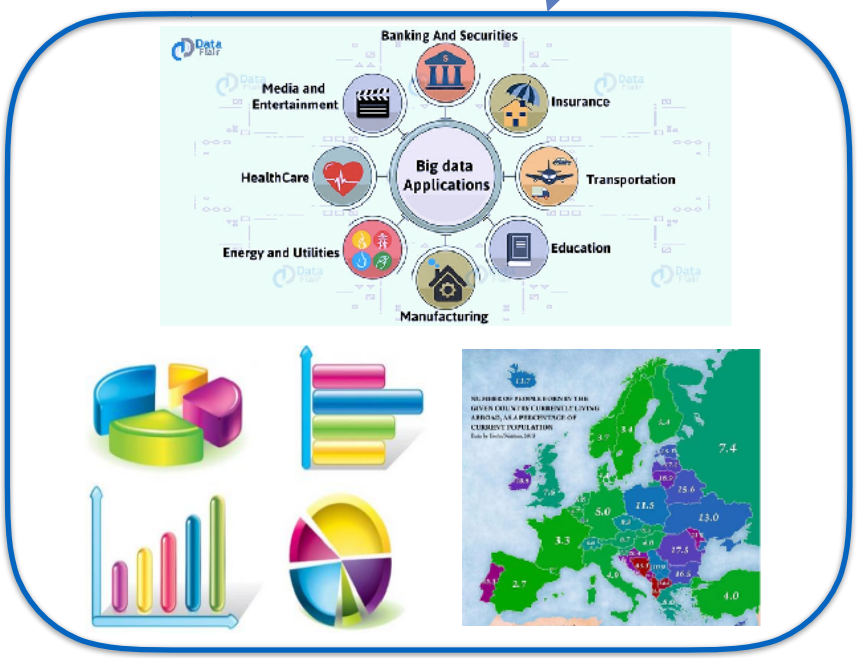
Catuscia Palamidessi

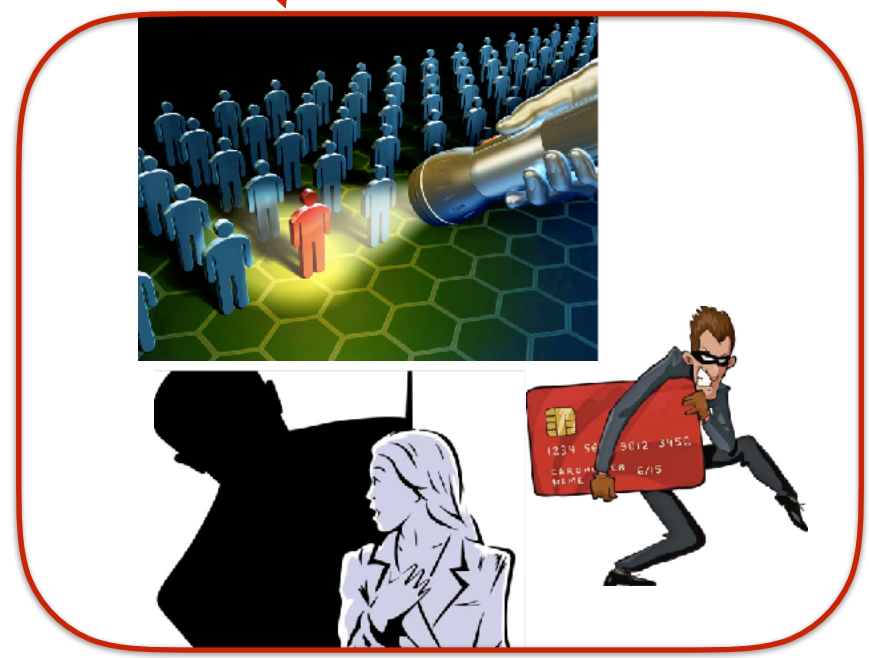
Information Age

90% of the world data have been generated in the last 2 years! (source: IMB, 2017)



Utility versus privacy





Utility

Two kinds of utility:

- Quality of service
- Statistical analysis

Privacy, QoS and Statistical analysis are interrelated: The user often releases his data in exchange of a service, but it should not pose a threat to his privacy. In turn, the service provider offers the service because it's interested in collecting the user's data, which are often used to derive statistics.

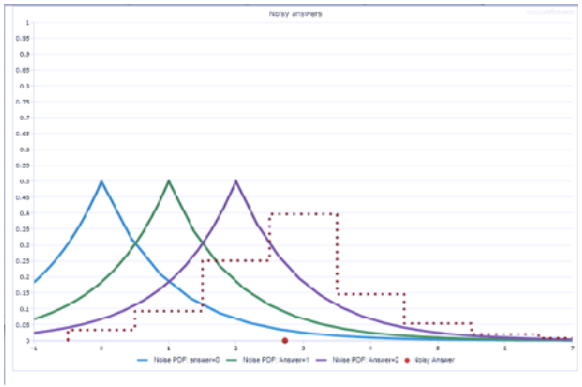
It is important to find mechanisms that optimize the trade-off between these three

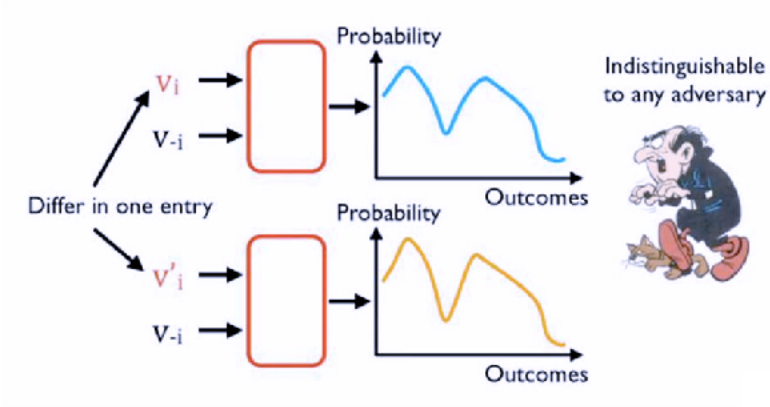
Privacy by randomization

Differential Privacy [Dwork et al., '96]

A mechanism $\mathcal{K}$ (for a certain query) is ε-differentially private if for every pair of *adjacent* datasets x and x' and every possible answer y

$$P[\mathcal{K}(x) = y] \leq e^{\varepsilon}\ P[\mathcal{K}(x') = y]$$

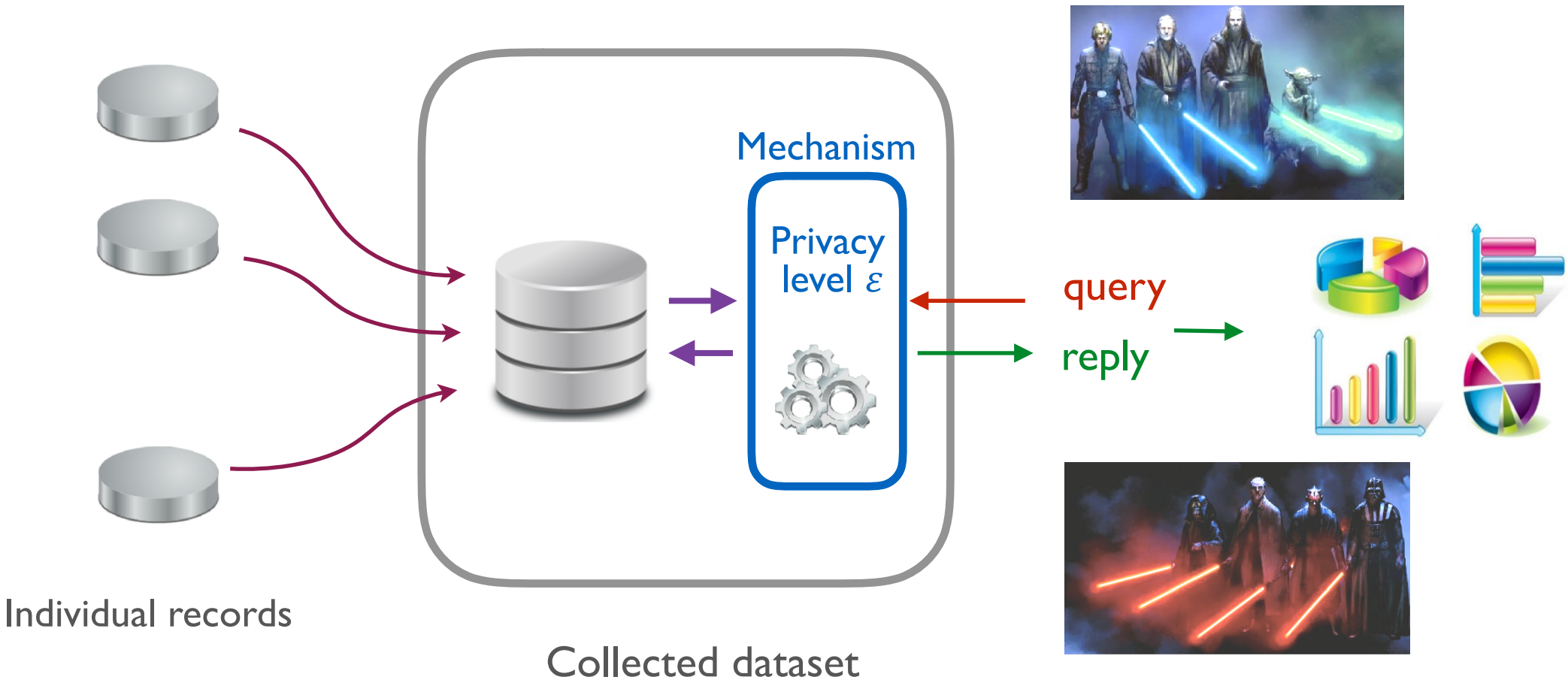




- **Compositionality**: the combination of two mechanisms which are e_1 and e_2 differentially private is $e_1 + e_2$ differentially private

- **Independent** from side knowledge

Standard Differential Privacy



Standard Differential Privacy

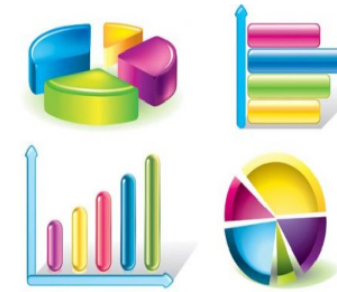
Mechanism

Privacy level

ε

query

reply



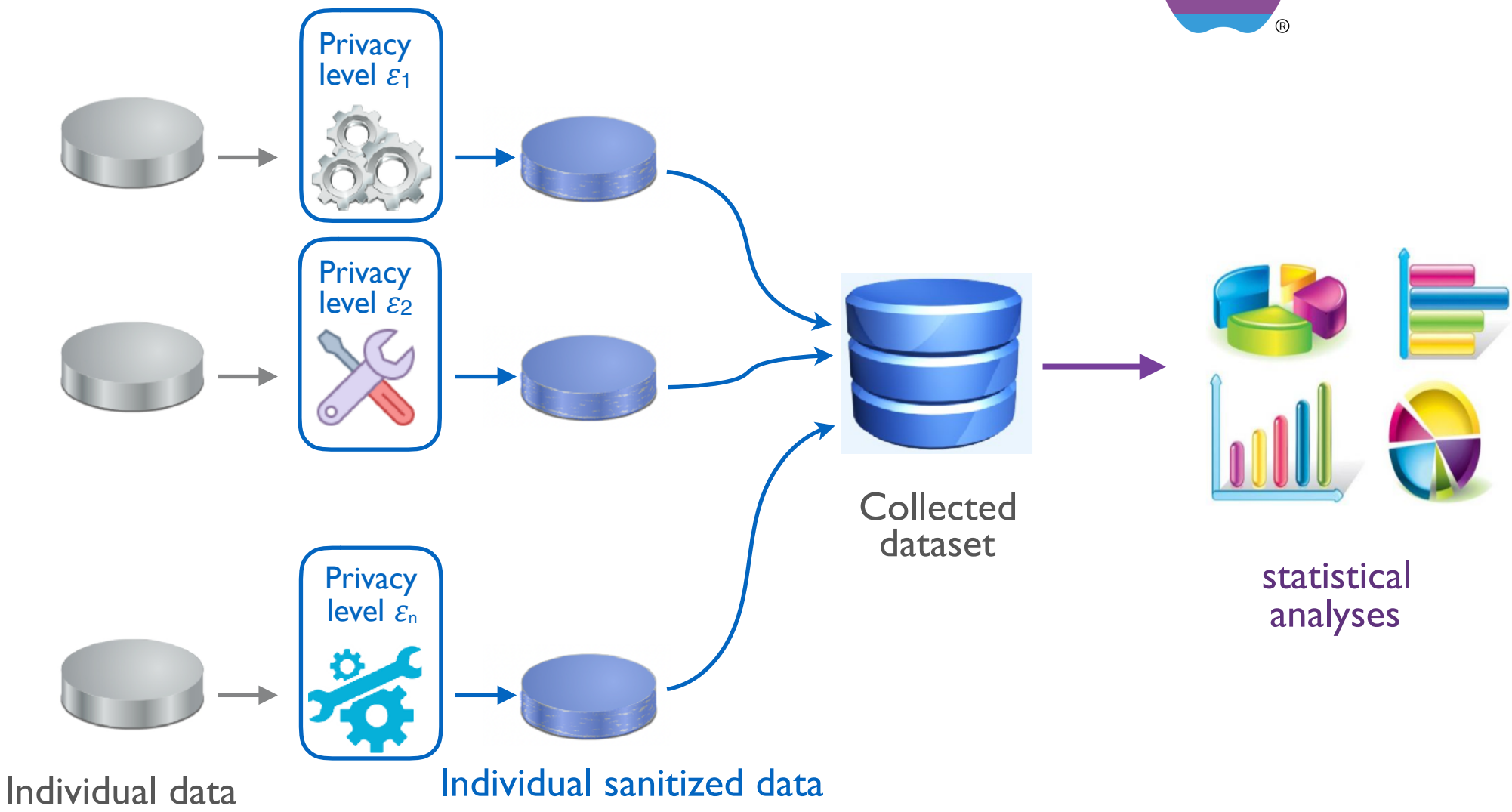
Individual records

Collected dataset

Local Differential Privacy

Google

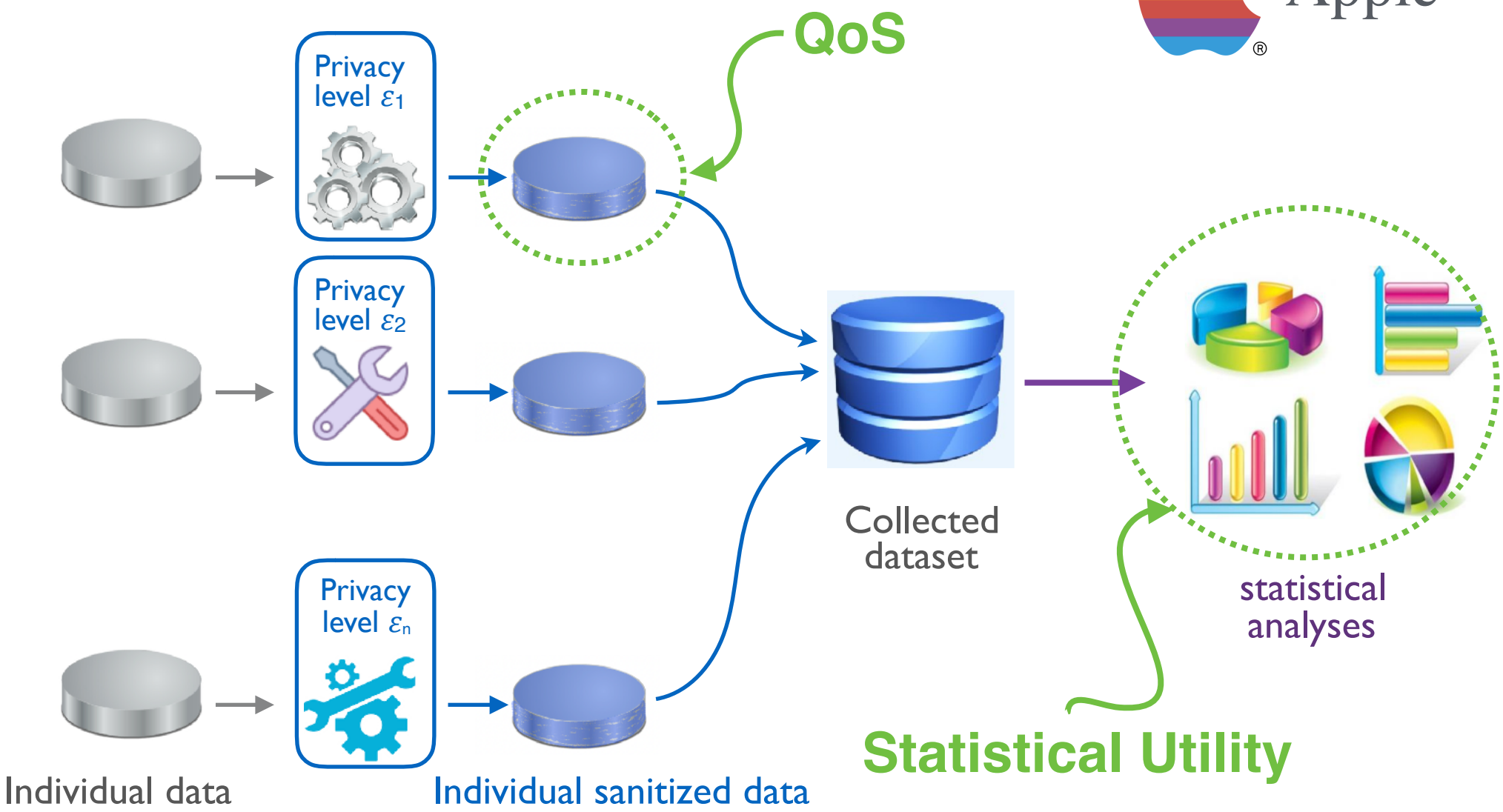
Apple



Local Differential Privacy

Google

Apple



Standard Local Differential Privacy

[Jordan &Wainwright '13]

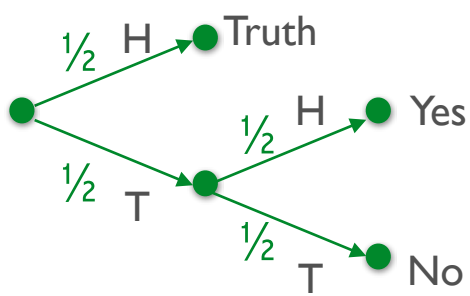
Definition Let $\mathcal{X}$ be a set of possible values and $\mathcal{Y}$ the set of noisy values. A mechanism $\mathcal{K}$ is ε-locally differentially private (ε-LDP) if for all $x_1, x_2 \in \mathcal{X}$ and for all $y \in \mathcal{Y}$

$$P[\mathcal{K}(x) = y] \leq e^{\varepsilon}\ P[\mathcal{K}(x') = y]$$

or equivalently, using the conditional probability notation:

$$p(y \mid x) \leq e^{\varepsilon}\ p(y \mid x')$$

For instance, the Randomized Response protocol is (log 3)-LPD



x

	y: yes	y: no
yes	¾	¼
no	¼	¾

The flat mechanism (aka k-RR)

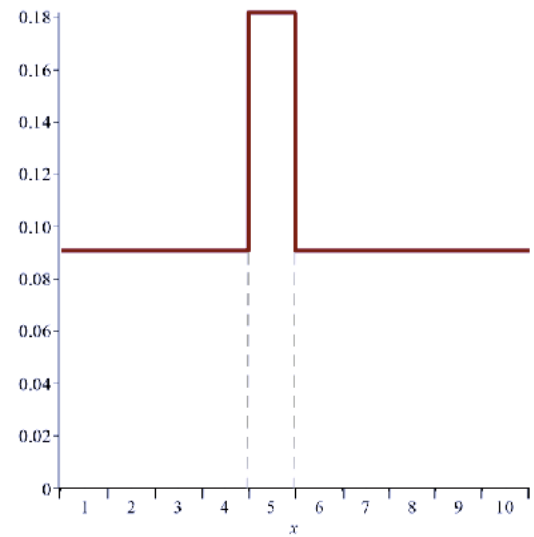
[Kairouz et al, '16]

The flat mechanism is the simplest way to implement LPD. It is defined as follows:

$$p(y|x) = \begin{cases} c\,e^{\varepsilon} & \text{if } x = y \\ c & otherwise \end{cases}$$

where c is a normalization constant.

namely $c = \dfrac{1}{k - 1 + e^{\varepsilon}}$ where k is the size of the domain



Privacy Properties:

- Compositionality
- Independence from the side knowledge of the adversary

What about Utility ?

- Statistical Utility
- QoS

Statistical Utility

Statistical utility: The matrix inversion method
[Kairouz et al, '16]

- Let C be the stochastic matrix associated to the mechanism
- Let q be the empirical distribution (derived from the noisy data).
- Compute the approximation of the true distribution as $r = q\,C^{-1}$

Example Assume $q(Yes) = \frac{6}{10}$ and $q(No) = \frac{4}{10}$. Then:

$$\frac{3}{4}\,p(Yes) + \frac{1}{4}\,p(No) = \frac{6}{10}$$

$$\frac{1}{4}\,p(Yes) + \frac{3}{4}\,p(No) = \frac{4}{10}$$

From which we derive $p(Yes) = \frac{7}{10}$ and $p(No) = \frac{3}{10}$

x \ y	yes	no
yes	¾	¼
no	¼	¾

Statistical utility: The matrix inversion method

Problem 1: C must be invertible

Problem 2: Assume $q(Yes) = \frac{4}{5}$ and $q(No) = \frac{1}{5}$. Then:

$$\frac{3}{4}\, p(Yes) + \frac{1}{4}\, p(No) = \frac{4}{5}$$
$$\frac{1}{4}\, p(Yes) + \frac{3}{4}\, p(No) = \frac{1}{5}$$

x \ y	yes	no
yes	¾	¼
no	¼	¾

From which we derive $p(Yes) = \frac{11}{10}$ and $p(No) = -\frac{1}{10}$

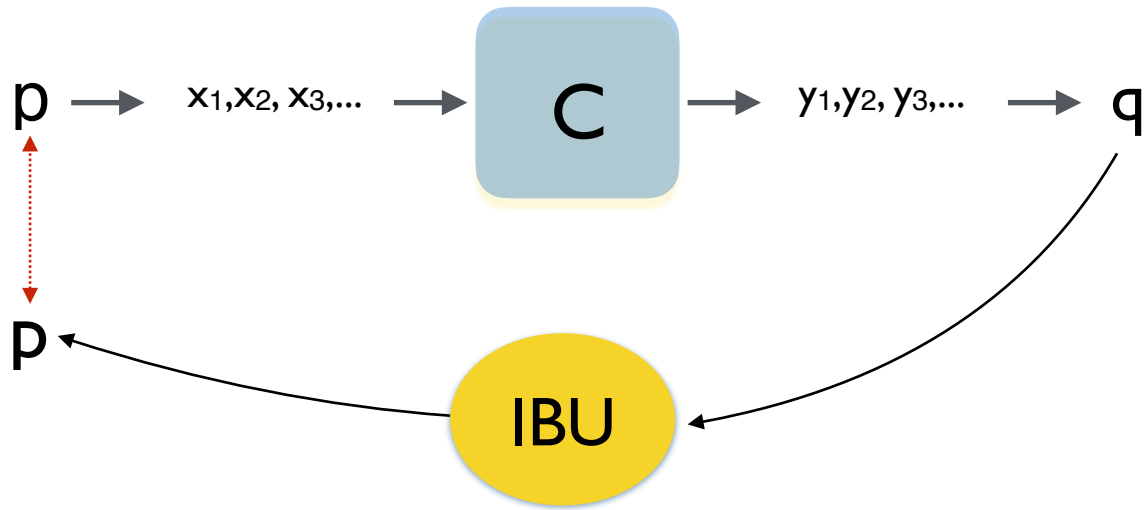
Statistical utility: The matrix inversion method

$r = q\,C^{-1}$ may not be a distribution because it may contain negative elements. In order to try to obtain the true distribution π we can either:

- set to 0 all the negative elements, and renormalize, or
- project r on the simplex.

The resulting distribution however usually is not the best approximation of the original distribution.

Our approach: Iterative Bayesian Update



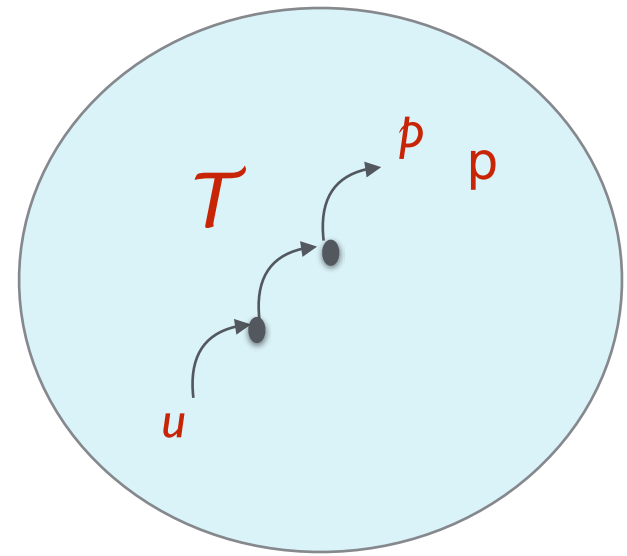
The IBU:

- is based on the **Maximization-Expectation** method
- produces a **Maximum Likelihood Estimator** $\hat{p}$ of the true distribution p
- Under certain conditions on C, the MLE is unique and converges to p

The Iterative Bayesian Update

- Define $p^{(0)}$ = any distribution (for ex. the uniform distribution)
- Repeat: Define $p^{(n+1)}$ as the Bayesian update of $p^{(n)}$ weighted on the corresponding element of q, namely:

$$p_x^{(n+1)} = \sum_y q_y \frac{p_x^{(n)} C_{xy}}{\sum_z p_z^{(n)} C_{zy}}$$

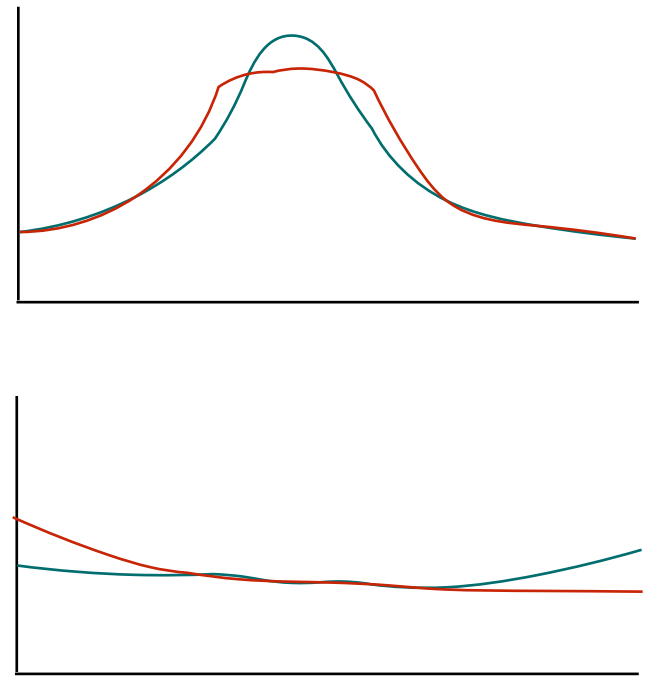


- Note that $p^{(n+1)} = T(p^{(n)})$
- We study the conditions on C under which T is a contraction
- If T is a contraction then there is a unique fixed point p and it converges to p (as the size of the dataset grows). Furthermore, when $p^{(n+1)}$ does not differ much from $p^{(n)}$ we know that we are close to the fixed point, and we can stop.

Measuring the quality of the approximation

There are many measures of distance between distributions.
A typical one is the total variation distance. Our proposal, however is to use the **Kantorovich distance** (aka Earth Movers distance).

- The Total Variation distance measures only the area between the two probability distributions
- The Kantorovich takes into account also the ground distance; it measures the "transportation effort" to make the two distributions equal. Cfr. "Earth moving distance"
- In these two examples the TV is the same, while the Kantorovich is larger in the second case



- The Kantorovich metric is particularly suitable when we are interested in statistics that are sensitive to the underlying distance.
  Example: placement of hotspots.

$$K_d(\mu,\nu) = \sup_{f \in Lip} \left| \sum_x \mu_x f(x) - \sum_x \nu_x f(x) \right|$$

where Lip is the set of Lipshitz functions wrt d

Quality of Service

Quality of Service

An **abstract** notion of utility loss:

- Following the approach of Shokri et al. we consider as utility loss the **rate distortion**, namely the expected distance between the true value and teh obfuscated value.

- This makes sense for all those services that depend on the accuracy of data. Of course, in practical applications things can be more complicated.

Our approach to LDP

d-privacy

d-privacy: a generalization of DP and LDP

d-privacy

On a generic domain $\mathcal{X}$ provided with a distance d:

$$\forall x, x' \in \mathcal{X}, \forall z \quad \frac{p(z \mid x)}{p(z \mid x')} \leq e^{\varepsilon\, d(x,x')}$$

generalizes

Differential Privacy

- x, x' are databases
- *d* is the Hamming distance

Local Differential Privacy

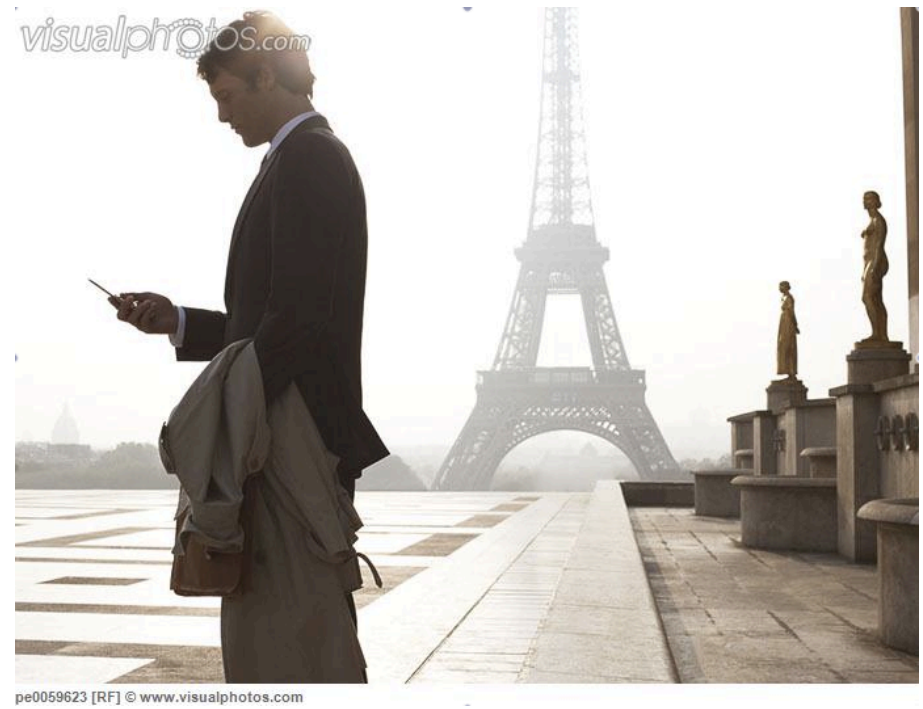
- *d* is the discrete distance

Properties

- Like LDP, it can be applied at the user side
- Like DP and LDP, it is compositional

QoS: we extensively studied *d*-privacy in the case of Location Privacy for Location Based Services

- Example of LBS: find the restaurants near the user
- Revealing the exact location may be dangerous: profiling, inference of sensitive information, etc.
- Revealing an approximate location is usually ok
- QoS: decreases with the expected distance between the real location and the noisy one.



Location privacy: geo-indistinguishability

d : the Euclidean distance

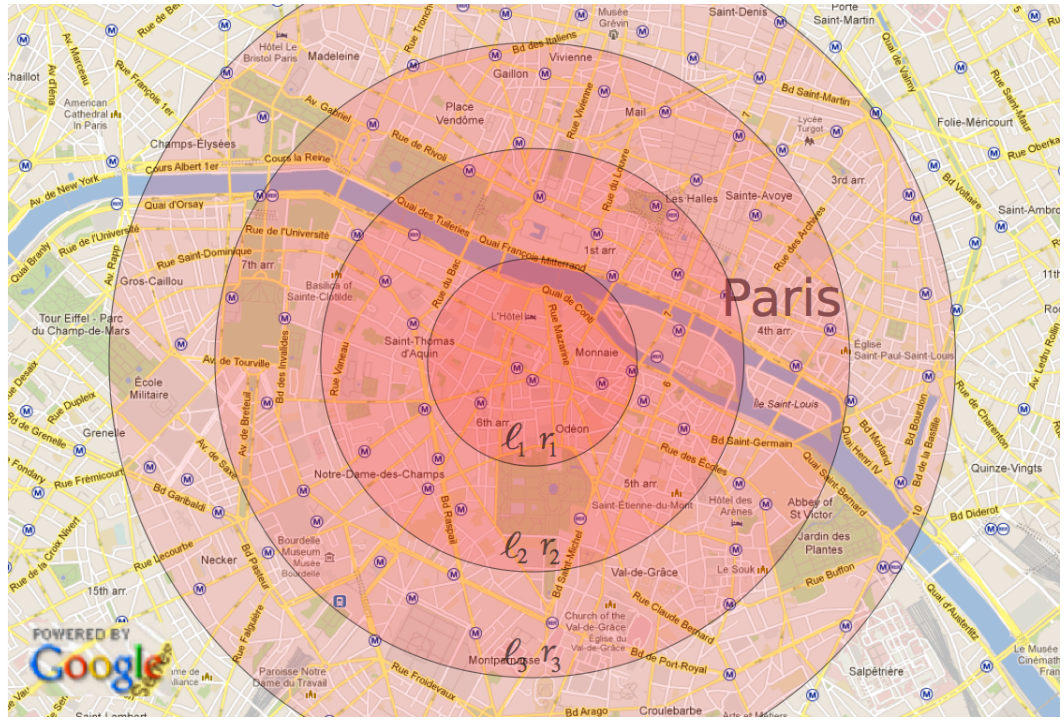
x : the exact location

z : the reported location

$d-$ privacy

$$\frac{p(z|x)}{p(z|x')} \leq e^{\epsilon r}$$

where r is the distance between x and x'



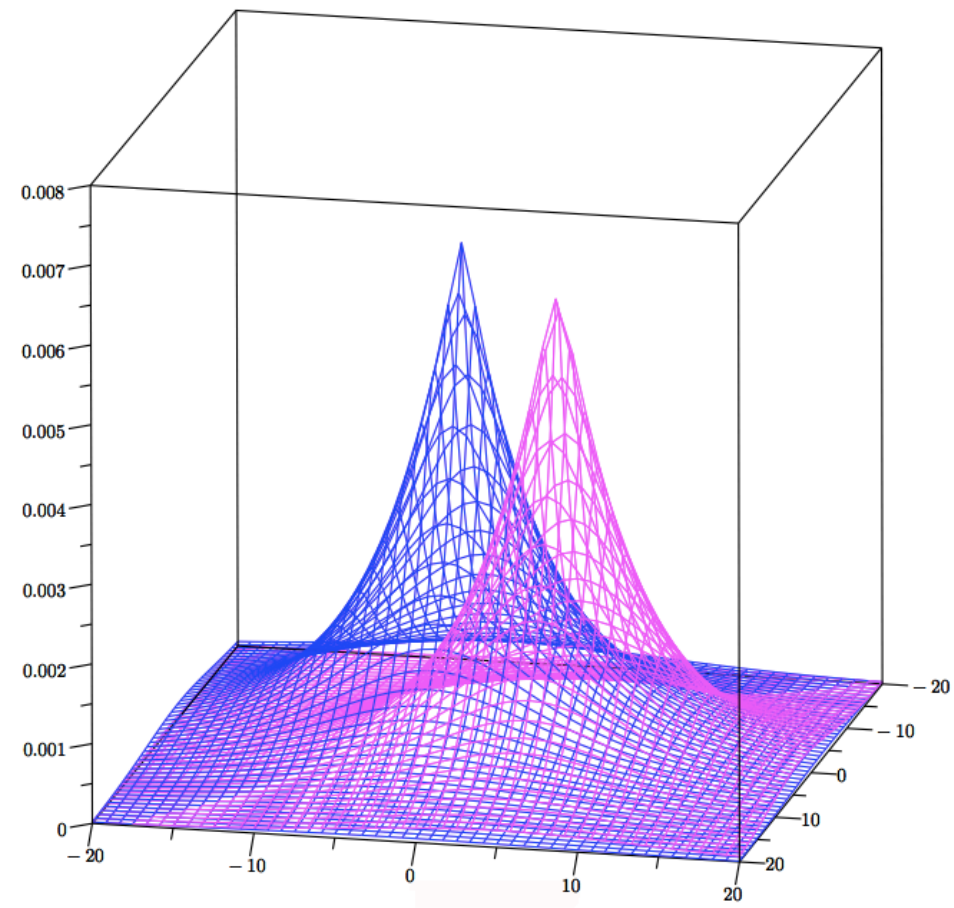
We call this property **geo-indistinguishability.** Like DP, it is:

1) independent from the prior,
2) compositional

d-private mechanisms for LBS:
Planar laplacian and Planar Geometric

$$dp_x(z) = \frac{\epsilon^2}{2\pi} e^{\epsilon\, d(x,z)}$$

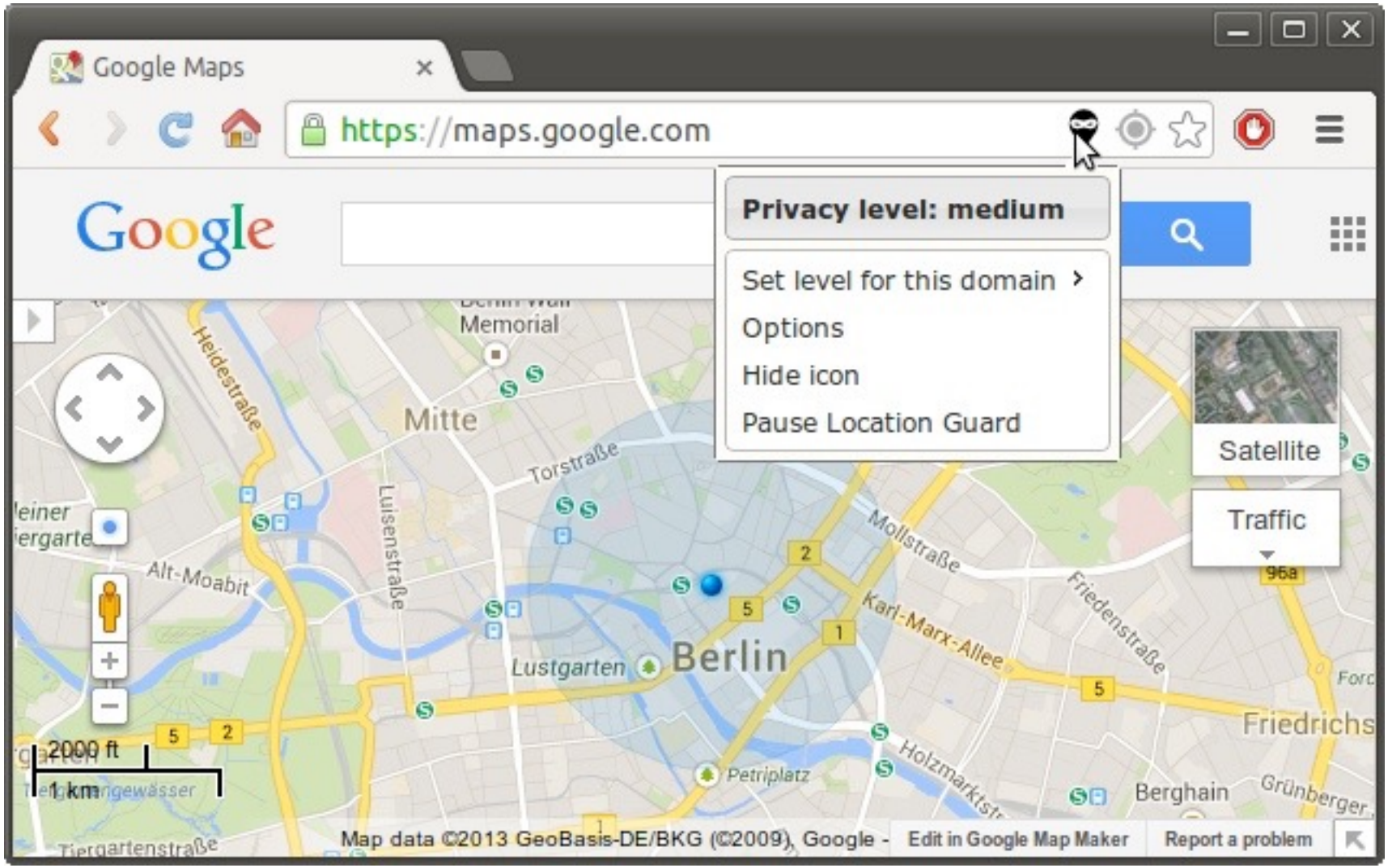
- Efficient method to draw noisy locations based on polar coordinates
- Then we translate from polar coordinates to standard (latitude, longitude) coordinates.
- Some degradation of the privacy level in single precision, but negligeable in double precision.



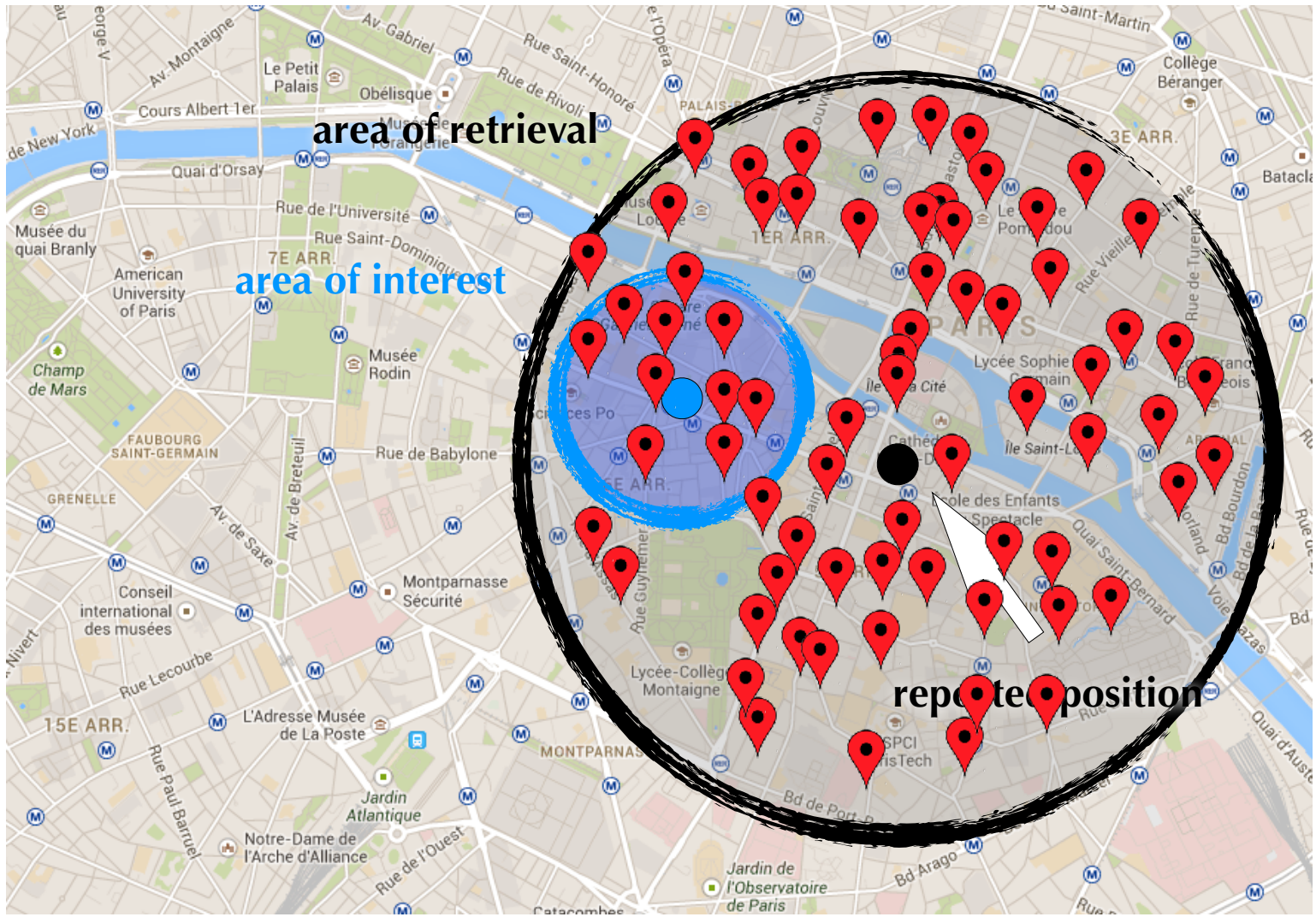
Tool: "Location Guard"

http://www.lix.polytechnique.fr/~kostas/software.html

Extension for Firefox, Chrome, and Opera. It has been released about two yeas ago, and nowadays it has about 60,000 active users.



How it works



Trade-off privacy-QoS

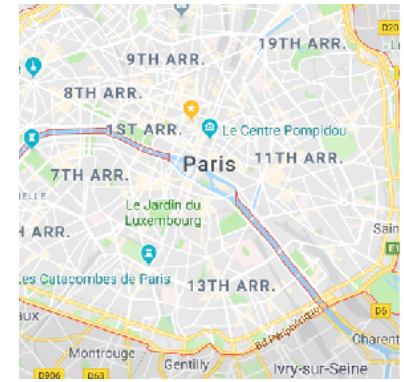
Comparison with other methods for location privacy

Four mechanisms:

- Ours (Planar Laplacian)
- Cloaking
- Optimal by [Shroki et al. 2012] for uniform prior
- Optimal by [Shroki et al. 2012] for a given prior

No k-RR: it has a very bad QoS

(a)



(b)

Evaluation:

- Gowalla dataset, various towns, divided in a grid 10 x 10
- The levels of privacy are calibrated so that our method offers the same level of privacy according to the definition of privacy of Shokri et al (Bayesian adversary)

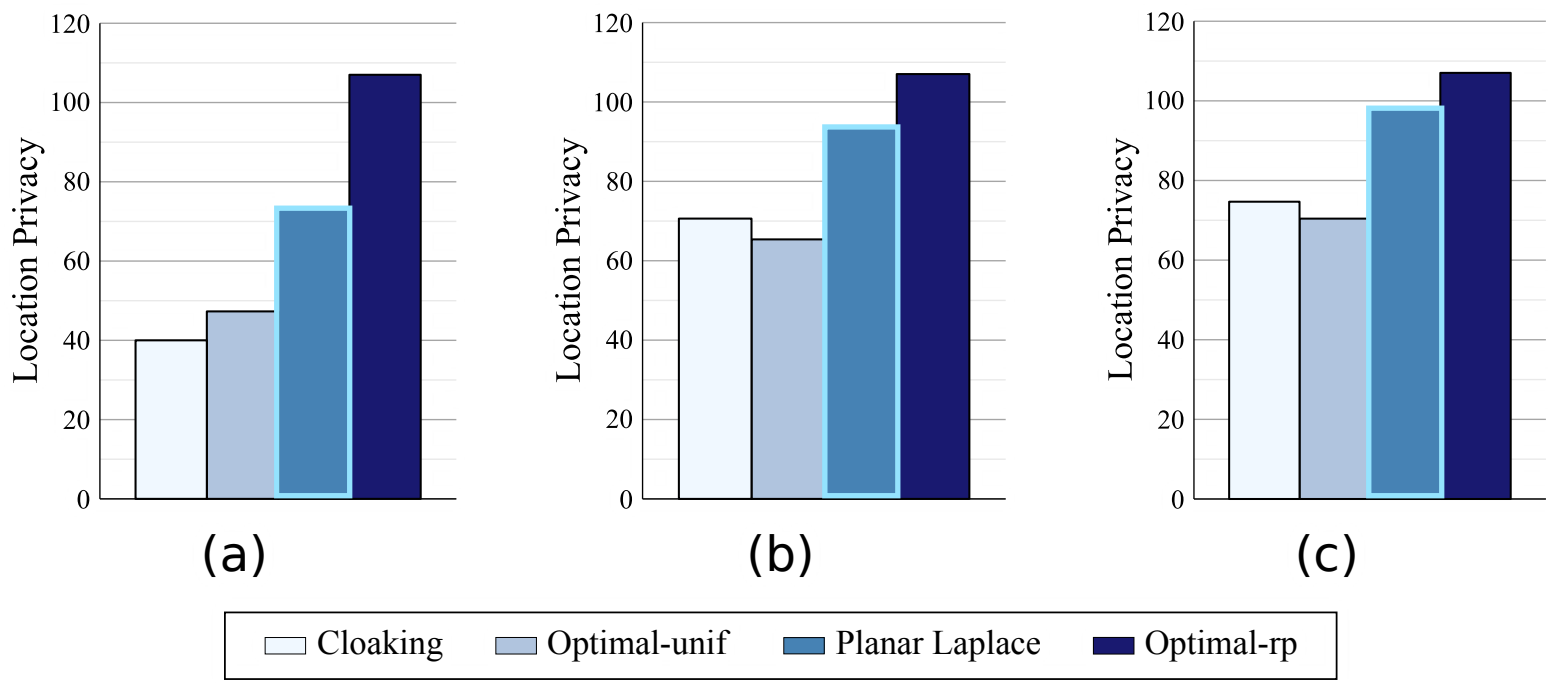
(c)

Privacy versus QoS: evaluation

The four mechanisms:

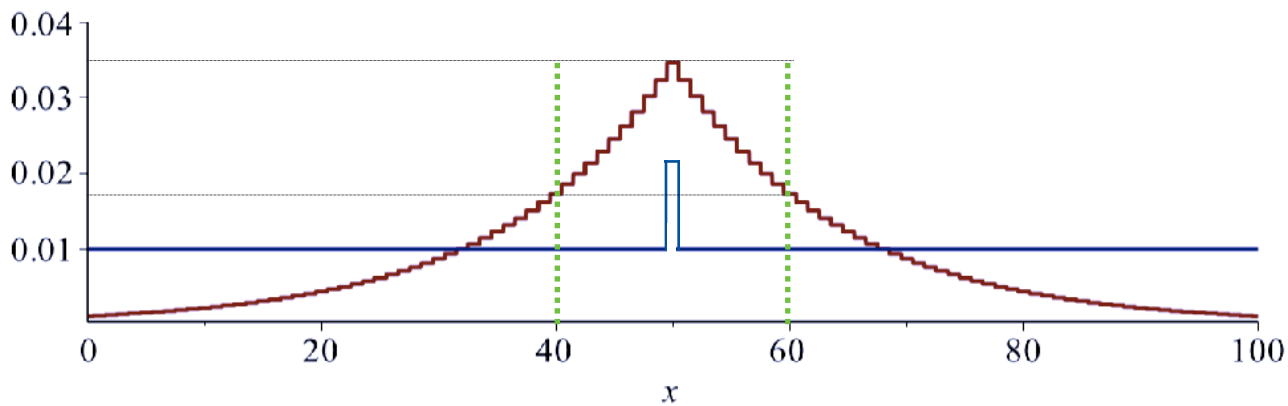
- Cloaking,
- Optimal by [Shroki et al. CCS 2012] generated assuming uniform prior
- Ours (Planar Laplacian)
- Optimal by [Shroki et al. CCS 2012] generated assuming the given prior

Based on linear optimization: high complexity



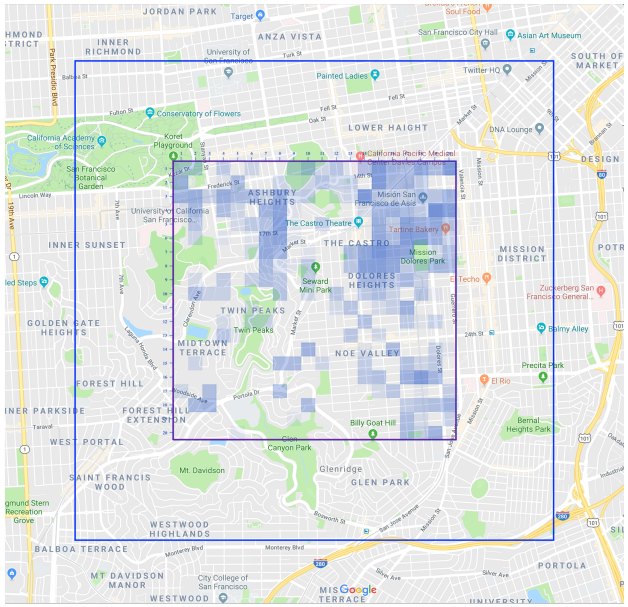
Trade-off between privacy and statistical utility
Comparison with k-RR

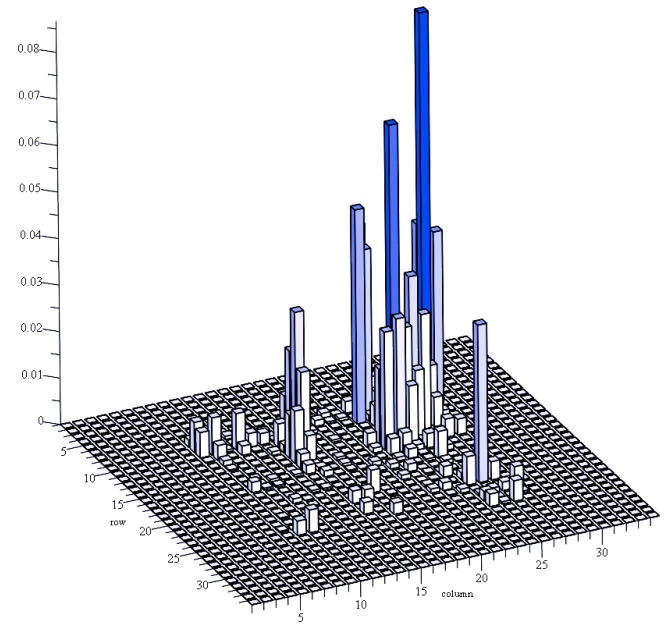
Both K-RR and the geometric / laplacian mechanisms are parametrized by ε, but it has a different meaning. We need to calibrate ε, in such a way that the requested ratio is satisfied in the "area of interest" (area in which we want to be indistinguishable)



Experiments on the Gowalla dataset

- Gowalla is a dataset of geographical checkins in several cities in the world
- We have used it to compare the statistical utility of kRR and Planar Laplacian with the respective e calibrated so to satisfy the same privacy constraint:
  same level of privacy within about 1 Km^2

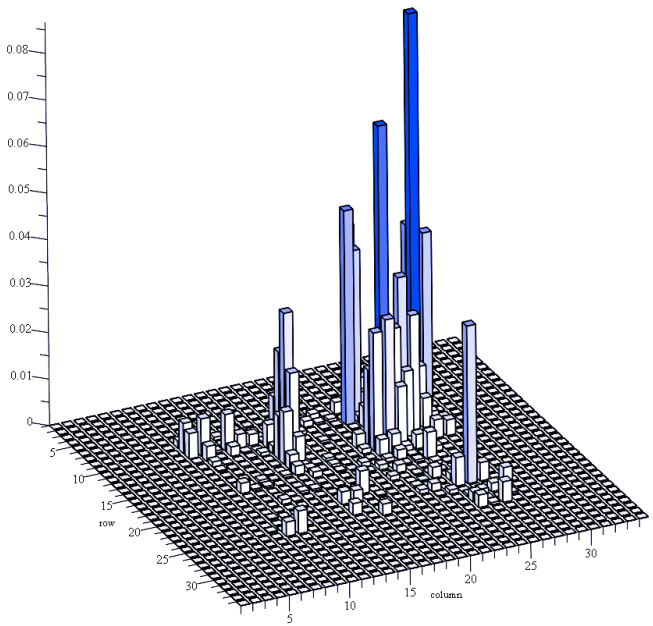




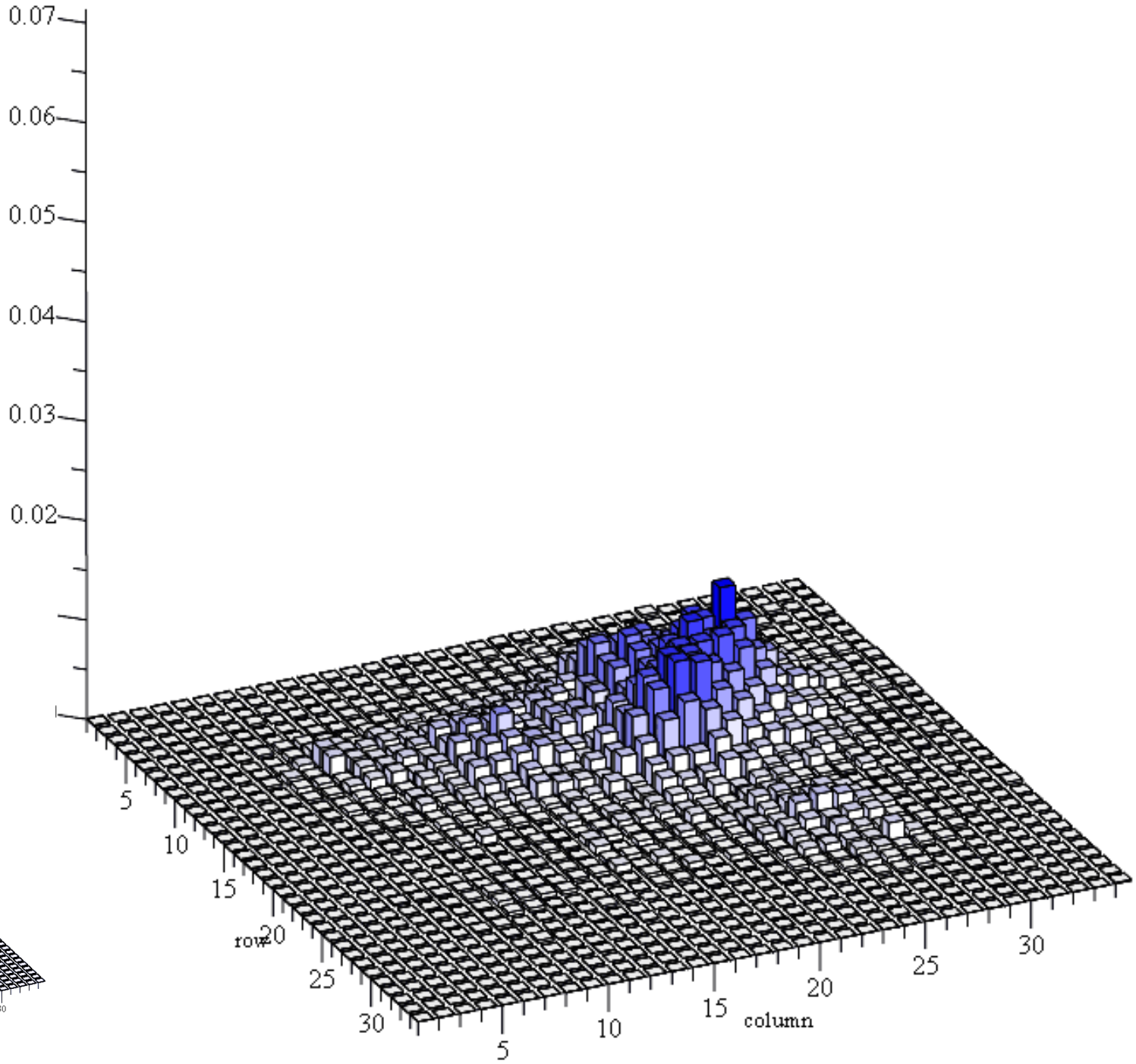
Gowalla checkins in an area of 3x3 km^2 in San Francisco downtown (about 10K checkins)

The Planar Laplace mechanism

$\varepsilon = \ln(2)$



The real distribution

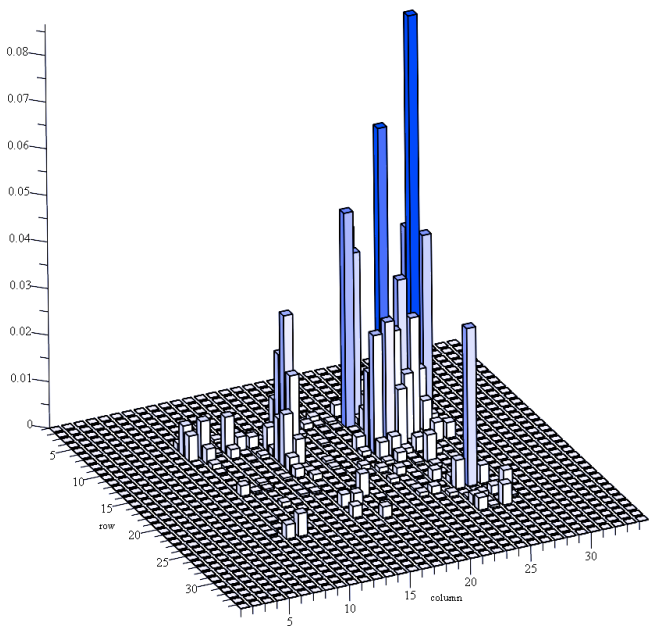


The noisy distribution and the result of the IBU (300 iterations)

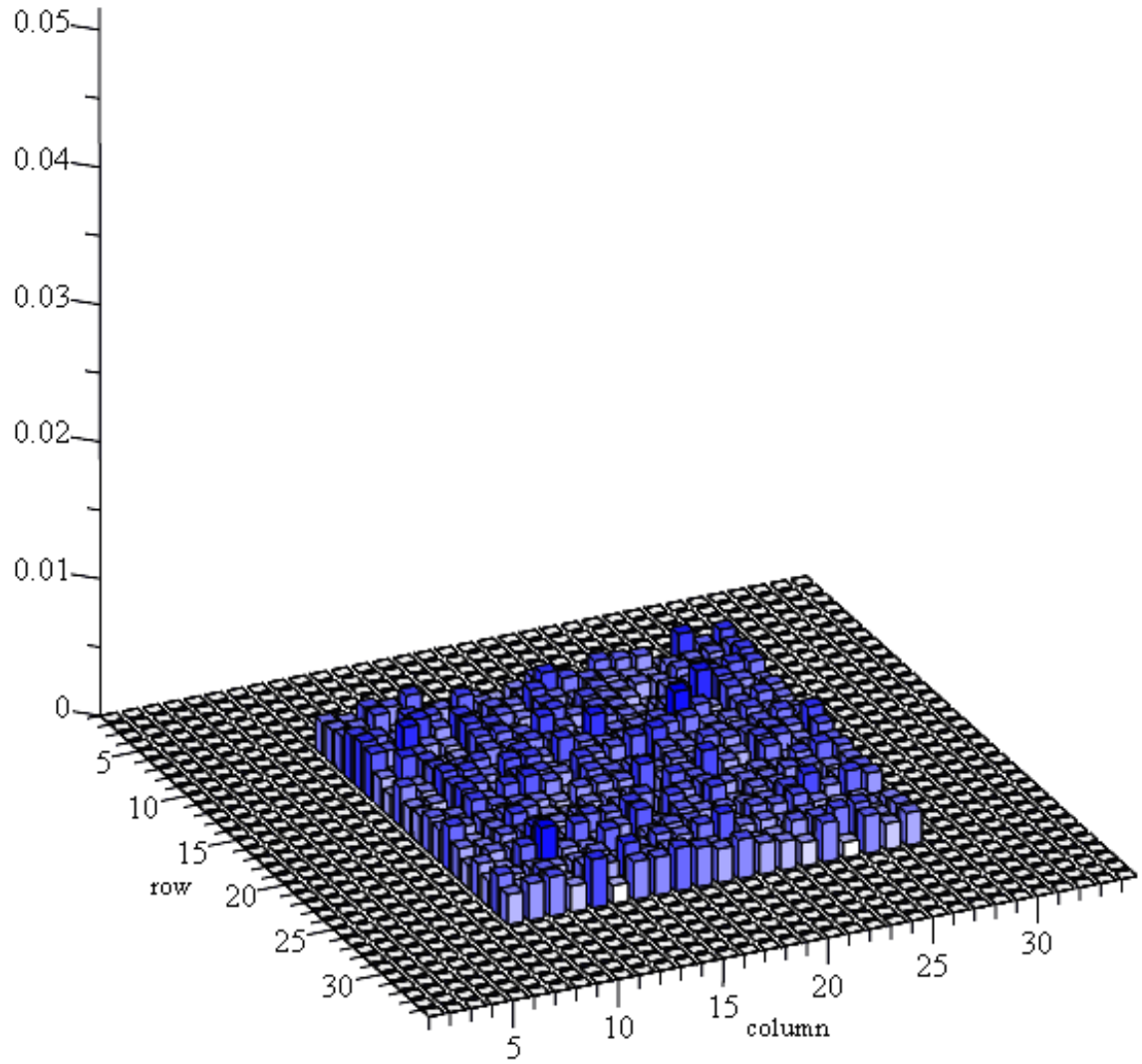
$n = 0.$

The kRR mechanism

$\varepsilon = \ln(8)$

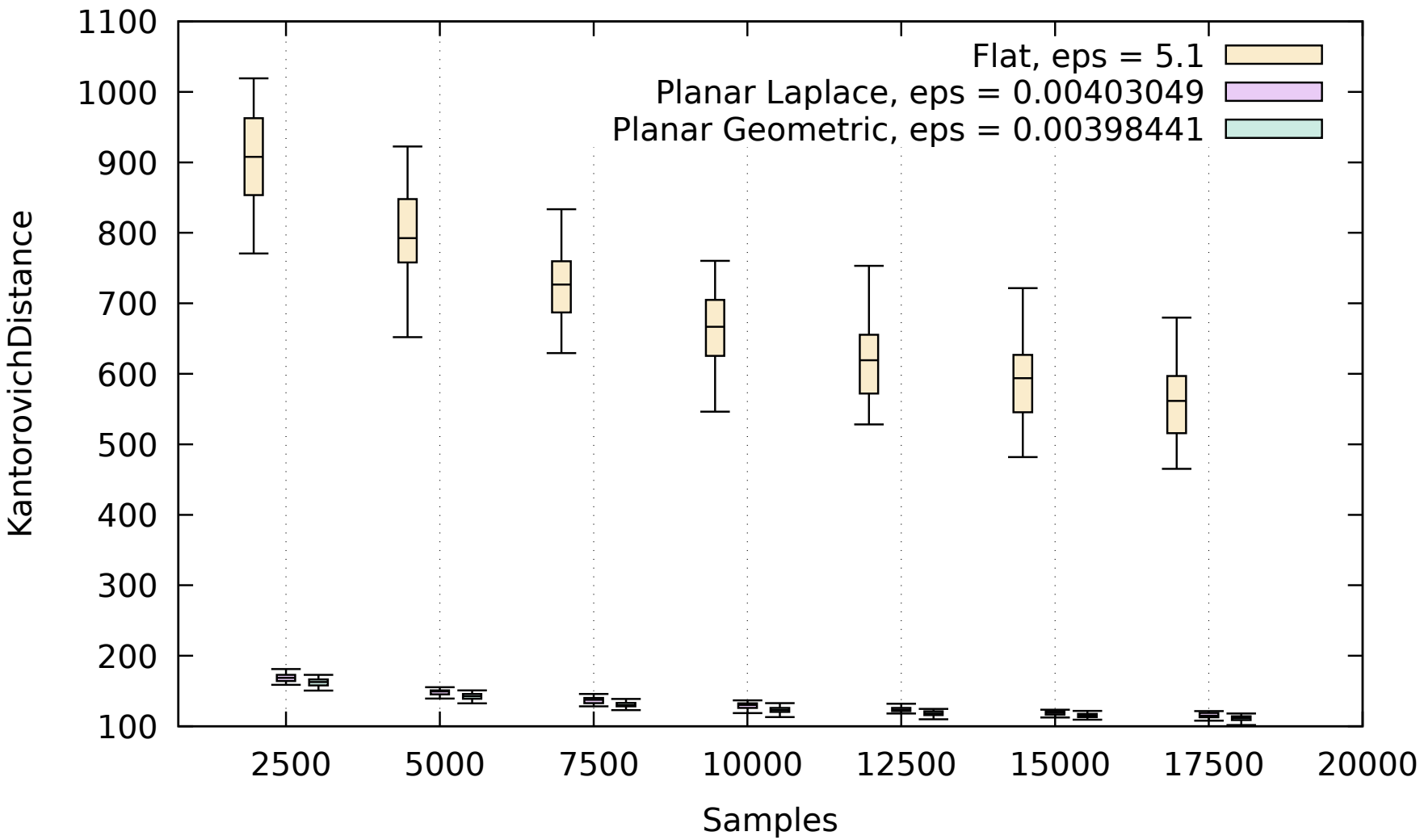


The real distribution

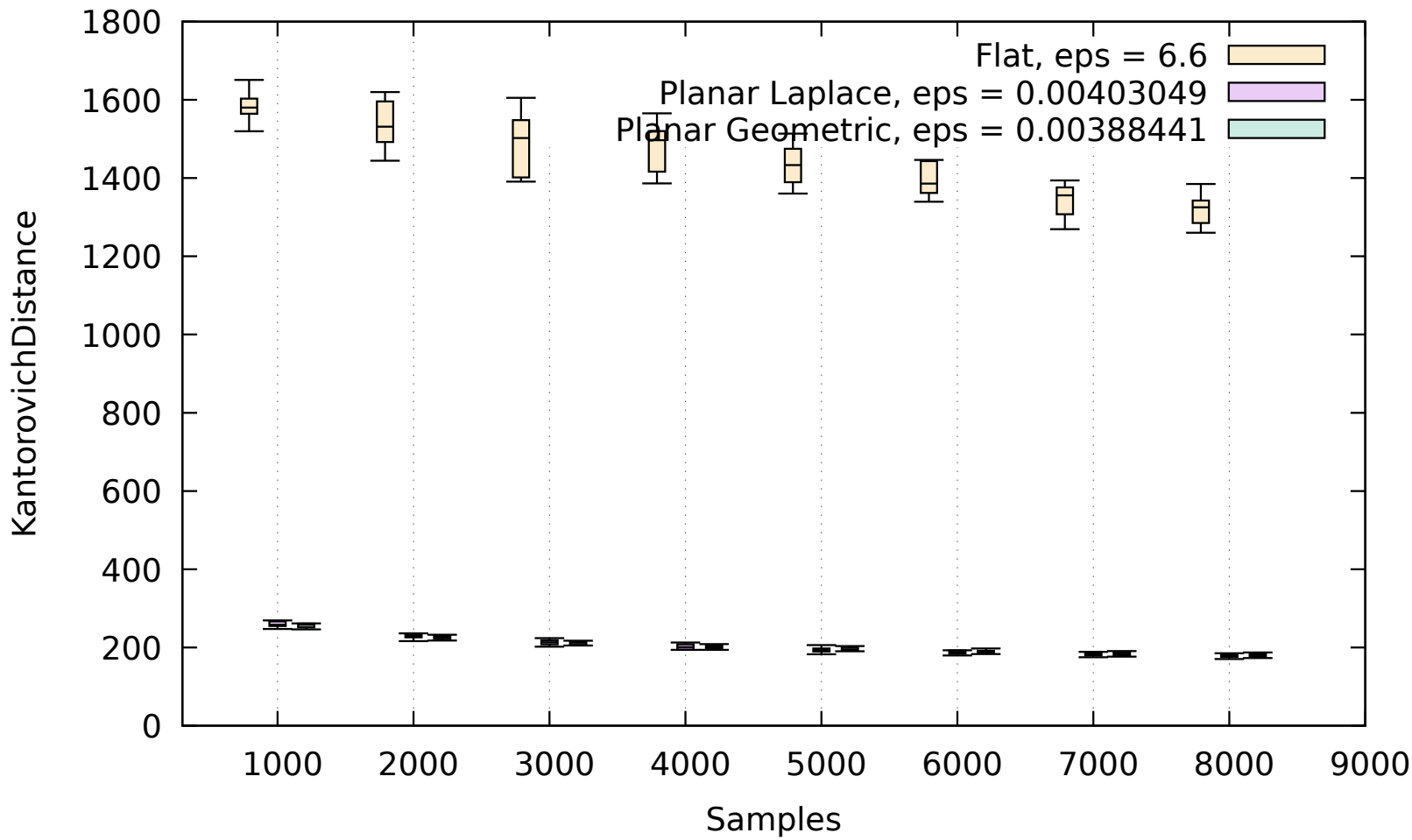


The noisy distribution and the result of the IBU (500 iterations)

Evaluation: San Francisco



Evaluation: Paris



Thanks!

Questions ?